Optimal Real-Time QBI using Regularized Kalman Filtering with Incremental Orientation Sets

Rachid Deriche, Jeff Calder, Maxime Descoteaux

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Optimal Real-Time Q-Ball Imaging using Regularized Kalman Filtering with Incremental Orientation Sets

Rachid Deriche — Jeff Calder — Maxime Descoteaux

N° 6793

January 2009

Thème BIO

apport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6793--FR+ENG

Optimal Real-Time Q-Ball Imaging using Regularized Kalman Filtering with Incremental Orientation Sets

Rachid Deriche* , Jeff Calder† , Maxime Descoteaux‡

Thème BIO — Systèmes biologiques
Projet Odyssée

Rapport de recherche n° 6793 — January 2009 — 44 pages

Abstract: Diffusion MRI has become an established research tool for the investigation of tissue structure and orientation from which has stemmed a number of variations, such as Diffusion Tensor Imaging (DTI) Diffusion Spectrum Imaging (DSI) and Q-Ball Imaging (QBI). The acquisition and analysis of such data is very challenging due to its complexity. Recently, an exciting new Kalman filtering framework has been proposed for DTI and QBI reconstructions in real time during the repetition time (TR) of the acquisition sequence [27, 29]. In this article, we first revisite and thoroughly analyze this approach and show it is actually *sub-optimal* and *not* recursively minimizing the intended criterion due to the Laplace-Beltrami regularization term. Then, we propose a new approach that implements the QBI reconstruction algorithm in real-time using a fast and robust Laplace-Beltrami regularization without sacrificing the optimality of the Kalman filter. We demonstrate that our method solves the correct minimization problem at each iteration and recursively provides the optimal QBI solution. We validate with real QBI data that our proposed real-time method is equivalent in terms of QBI estimation accuracy to the standard off-line processing techniques and outperforms the existing solution. Last, we propose a fast algorithm to recursively compute gradient orientation sets whose partial subsets are almost uniform and show that it can also be applied to the problem of efficiently ordering an existing point-set of any size. Our work allows to start an acquisition just with the minimum number of gradient directions and an initial estimate of the q-ball and then all the rest, including the next gradient directions and the q-ball estimates, are recursively and optimally determined, allowing the acquisition to be stopped as soon as desired or at any iteration with the optimal q-ball estimate. This opens new and interesting opportunities for real-time feedback for clinicians during an acquisition and also for researchers investigating into optimal diffusion orientation sets and, real-time fiber tracking and connectivity mapping.

* Rachid.Deriche@sophia.inria.fr
† Jeff.Calder@sophia.inria.fr
‡ Maxime.Descoteaux@cea.fr. NeuroSpin, IFR 49, I2BM, CEA Saclay, Gif-sur-Yvette, France

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Key-words: High Angular Resolution Diffusion Imaging, Q-ball imaging, Orientation Distribution Function, real-time diffusion MRI, Kalman filtering, Spherical Harmonics, Laplace-Beltrami Regularization

Imagerie par Q-Ball Optimale et en Temps-Réel base de Filtre de Kalman Regularisé et d'Orientations de Diffusion Définis Incrémentalement

Résumé : L'IRM de diffusion, avec ses techniques récentes comme l'imagerie par tenseur de diffusion (DTI) et l'imagerie par q-ball (QBI), est maintenant reconnue comme un outil important pour l'étude de l'architecture neuronale du système nerveux central. Cependant le temps d'acquisition et l'analyse de ce type d'imagerie de diffusion restent encore aujourd'hui complexes et très lourds. Récemment, une nouvelle méthode d'acquisition DTI et QBI temps-réel a été proposée en utilisant le filtre de Kalman [27, 29]. Dans ce rapport, nous revisitons d'abord cette très intéressante approche à base de filtre de Kalman et montrons qu'elle est en fait, *sous-optimale*, et qu'elle ne minimise pas le critère désiré à cause du terme de régularisation de Laplace-Beltrami. Nous proposons donc une nouvelle approche optimale de QBI temps-réel qui est rapide et robuste, en incluant le terme de regularisation Laplace-Beltrami directement dans le filtrage de Kalman. Nous démontrons que notre solution minimise effectivement le bon critère à toute les itérations de l'estimation et que la solution temps-réel est équivalente et aussi précise qu'une estimation QBI hors-ligne. Enfin, nous développons une méthode récursive pour calculer un jeu d'orientations de gradients de diffusion, avec sous-ensembles partiels d'orientations presque uniforme sur la sphère. Ceci permet de démarrer l'acquisition de diffusion avec un minimum de directions de diffusion et de rajouter incrémentalement de nouvelles directions jusqu'à ce que l'estimation désirée soit obtenue. L'acquisition peut donc être arrêtée à tout moment, tout en ayant une estimation QBI optimale. Cette nouveauté ouvre des perspectives très intéressantes pour l'interaction et l'intervention en temps-réel des cliniciens pendant l'acquisition, afin de mieux optimiser les directions de gradients utilisées et de faire ensuite du suivie de fibres (tractographie) en temps-réel.

Mots-clés : Imagerie par résonance magnétique (IRM), IRM de diffusion à haute résolution angulaire, imagerie par q-ball (QBI), fonction d'orientation de distribution des fibres (ODF), IRM de diffusion temps-réel, filtre de Kalman, harmoniques sphériques, regularisation Laplace-Beltrami.

Contents

1 Introduction **5**

2 Background **7**
2.1 Linear Observation Model 7
2.2 Kalman Filtering 7
2.3 Q-Ball Linear Model 8

3 Kalman Filtering with Regularization **9**
3.1 Existing Method 10
3.1.1 Analysis 10
3.1.2 Some Naive Approaches 12
3.2 Proposed Method 12

4 Diffusion Gradient Orientation Sets **15**
4.1 Existing Methods 16
4.2 An incremental algorithm to generate orientation sets 17

5 Experimental Validation **25**
5.1 Implementation Details 25
5.2 Results and Discussion 25

6 Conclusion **34**

A Pseudo-inverse of $\mathbf{B}^+$ **35**

B Analysis of Kalman Filtering from [27,29] **36**

C Matrix Inversion Formula **37**

D Formula for Kalman Filter Covariance Matrix **37**

E Formula for Kalman Filter Gain Matrix **38**

F Another Kalman Filtering Solution **39**

1 Introduction

Diffusion MRI (dMRI) is a recent Magnetic Resonance Imaging technique introduced in the middle of the 80's by [20, 21, 32]. Since the first acquisitions of diffusion-weighted images (DWI) in vivo by [22, 23] and the development of the rigorous formalim of the diffusion tensor (DT) model by [2, 3, 4], dMRI has become an established research tool for the investigation of tissue structure and orientation and has opened up a landscape of extremely exciting discoveries for medicine and neuroscience. dMRI utilizes the measurement of Brownian motion of water molecules to gain information about tissue structure and orientation inside the brain and other organs. Using dMRI to infer the three dimensional diffusion probability displacement function (PDF) requires the acquisition of many diffusion images sensitized to different orientations in the sampling space. The number of diffusion weighted images (DWI) required depends on how the diffusion is modeled. The well known DT model assumes the PDF is Gaussian and requires at least 6 DWIs plus an additional unweighted image. However, the Gaussian assumption is over-simplifying the diffusion of water molecules and thus has some limitations. While the Gaussian assumption is adequate for voxels in which there is only a single fiber orientation (or none), it breaks down for voxels in which there is more complicated internal structure. This is an important limitation, since resolution of DTI acquisition is between 1 mm^3 and 27 mm^3 while the physical diameter of fibers can be between 1 μm and 30 μm [6, 26]. Research groups currently agree that there is complex fiber architecture in most fiber regions of the brain [25]. In fact, it is currently thought that between one third to two thirds of imaging voxels in the human brain white matter contain multiple fiber bundle crossings [7].

Therefore, it is of utmost importance to develop techniques that go beyond the limitations of diffusion tensor imaging (DTI). To do so, high angular resolution diffusion imaging (HARDI) has been proposed to measure diffusion images along several directions. Some HARDI reconstruction techniques are model dependent, some model-free, some have linear solutions whereas others require non-linear optimization schemes. A good review of these methods can be found in [12]. In this paper, we will be using a recent HARDI technique known as the Q-Ball Imaging (QBI). Originally proposed by [36], QBI allows to reconstruct the angular part of the diffusion displacement probability density function (PDF) of water molecules, also called the diffusion orientation distribution function (ODF). This is a spherical function very useful to drive the tractography since its maxima are aligned with the underlying fiber directions at every voxel. QBI and the diffusion ODF play a central role in this work focused on the development of a real-time regularized ODF solution that outperforms the state-of-the-art ODF estimation. However, note that the regularized Kalman filtering framework could also be adapted easily to the fiber orientation density (FOD) [34, 35] reconstruction as well.

Generally, HARDI requires much more diffusion weighted measurements than traditional DTI acquisitions, but they can resolve some of the fiber crossing problems. This comes at the price of a longer acquisition time, which can be problematic for clinical studies involving children and people inflicted with certain diseases. Excessive motion of the patient during the acquisition process can force a acquisition to be aborted or make the diffusion weighted images useless. Thus, one would like to make only as many acquisitions as is necessary. According to the literature, this number is

likely to be somewhere between 50 and 200 diffusion weighted measurements but this is still an open question.

Recently [27, 29] nicely addressed this issue and proposed an original algorithm for real-time estimation of the diffusion tensor and the orientation distribution function (ODF) using the Kalman filtering framework. [3] were the first to mention to the potential use of the Kalman Filtering in the process of DTI reconstruction. However, to our knowledge, until the recent paper by [27, 29], no one had developed this idea, implemented it and put it in practice. The DT model, without any positivity constraints, is linear and easily fits into the Kalman filtering framework. However, the fast and robust analytical ODF reconstruction algorithm proposed in [13] includes a regularization term to deal with poor signal to noise (SNR) ratios, which adds a term in the minimization problem. In an attempt to include this regularization term in their Kalman filtering algorithm, [27, 29] make some adjustments to the reconstruction model. These adjustments make the Kalman filtering algorithm *sub-optimal* in terms of the Laplace-Beltrami criterion with the largest errors occurring at the beginning of the acquisition sequence. This has significant implications for the intended applications of the real-time dMRI processing. As we would like to stop the acquisition as soon as the estimation has converged, a good estimation of the ODFs is highly desirable at the beginning of the acquisition and thus, the development of an optimal and incremental solution is important.

We propose a Kalman filtering solution that will correctly incorporate the regularization term from [13] into the filter's parameters without changing the ODF reconstruction model. The basic idea is to go back to the derivation of the Kalman filtering equations and include this regularization term. The surprising result is that *only* the initial covariance matrix needs to be modified to correctly implement the regularization term. We will show that our proposed algorithm yields *optimal* ODF estimations at each iteration (ie: continuously from the beginning to the end of the acquisition) and hence clearly provides an important added value over the elegant and original approach pionnered in [27, 29].

Therefore, we revisite and analyze the Kalman filtering framework to derive the correct equations that allow to recursively estimate the optimal ODF at each iteration. In order for this framework to be fully incremental and take its full power, we also tackle the problem of the optimal choice of the diffusion weighted gradient orientation set. Typically, each measurement is acquired along a given orientation extracted from an optimised set of orientations estimated and ordered off-line, before the acquistion is started. Hence, another important contribution of this work will be to propose a fast algorithm to incrementally compute gradient orientation sets whose partial subsets are almost uniform. This will allow to start an acquisition just with the minimum number of gradient directions and an initial estimate of the ODF so that all other processing steps, including next gradient direction generation and ODF reconstruction, are recursively, incrementally and optimally determined. This will truly allow the acquisition to be stopped at any time with the optimal ODF estimate.

As background for the reader, we describe the Kalman filter in section 2.2. In section 3, we first thoroughly analyze the method from [27, 29] and then describe our proposed recursive solution and prove that it is optimal with respect to the regularization criterion used in [13]. Next, in section 4 we propose a fast algorithm to recursively compute gradient orientation sets whose partial subsets are roughly uniform and show that it can also be applied to the problem of efficiently ordering an

existing point-set of any size. Finally, in section 5 we present various set of experimental results verifying our claims before concluding the paper.

2 Background

Before presenting the body of this paper, we will provide some background on Kalman filtering and ODF reconstruction techniques as well as clarify some notation.

2.1 Linear Observation Model

Throughout this paper we will use the following notation for the linear observation model.

$$y_i = \mathbf{C}_i\mathbf{x} + \epsilon_i \tag{2.1}$$

where y_i is a scalar observation at the discrete time step i, $\mathbf{x} \in \mathbb{R}^n$ is the state vector we wish to estimate, $\mathbf{C}_i$ is the $1 \times n$ observation matrix and ϵ_i is a Gaussian random variable with zero mean and variance R_i. We will use the notation $E[\cdot]$ for the the expected value of a random variable or vector. Furthermore, we will assume that $E[\epsilon_i\epsilon_j] = 0$ for $i \neq j$. We will frequently combine all the measurements up to time k into a linear system in order to compute the least squares estimation of the state $\mathbf{x}$. In this case, we will use the notation

$$\mathbf{y}_k = \mathbf{B}_k\mathbf{x} + \eta_k \tag{2.2}$$

where $\mathbf{y}_k = [y_1 \ y_2 \ \dots \ y_k]^T$ is a vector of observations, $\eta_k = [\epsilon_1 \ \epsilon_2 \ \dots \ \epsilon_k]^T$ is the vector of Gaussian random variables, and $\mathbf{B}_k = [\mathbf{C}_1^T \ \mathbf{C}_2^T \ \dots \ \mathbf{C}_k^T]^T$ is the $k \times n$ matrix containing all the observation matrices up to time k. Here, we use the T operator to denote matrix or vector transposition. Let us denote the covariance matrix of η_k by $\mathbf{W}_k$. Since the noise sequence is zero mean, this is written as $\mathbf{W}_k = E\left[\eta_k\eta_k^T\right]$. Since we are assuming that the components of the Gaussian noise vector are uncorrelated, $\mathbf{W}_k$ is diagonal with diagonal elements given by $\text{Var}(\epsilon_j) = R_j$ for $j = 1 \dots k$.

We will denote by $\hat{\mathbf{x}}_k$ the minimum variance estimation of the state $\mathbf{x}$ given all observations from time $i = 1 \dots k$. Thus, we can express $\hat{\mathbf{x}}_k$ as

$$\hat{\mathbf{x}}_k = \arg\min_{\mathbf{x}\in\mathbb{R}^n} (\mathbf{y}_k - \mathbf{B}_k\mathbf{x})^T\mathbf{W}_k^{-1}(\mathbf{y}_k - \mathbf{B}_k\mathbf{x}). \tag{2.3}$$

The solution to equation (2.3) is easily obtained by differentiating with respect to $\mathbf{x}$. We obtain

$$\hat{\mathbf{x}}_k = (\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{y}_k \tag{2.4}$$

2.2 Kalman Filtering

The Kalman Filter is a linear optimal recursive estimator that first appeared in [19]. It was originally designed for linear time-varying dynamic systems but has since been successfully used for non-linear systems and problems involving state constraints. For an excellent textbook reference please

also see [8] . We recall the observation model equation (2.1)

$$y_i = \mathbf{C}_i\mathbf{x} + \epsilon_i.$$

As we record each new measurement, we gain an incremental amount of information about the state $\mathbf{x}$ and by combining all the recorded measurements into a linear system, we can compute the minimum variance estimation after k observations $\hat{\mathbf{x}}_k$ using equation (2.4).

$$\hat{\mathbf{x}}_k = (\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{y}_k$$

However, as we only gain an incremental amount of information about the state between observations y_{k-1} and y_k, it is reasonable to expect that the corresponding change in the state estimate between $\hat{\mathbf{x}}_{k-1}$ and $\hat{\mathbf{x}}_k$ is also incremental. The Kalman Filter allows us to quantify this incremental estimation correction and express $\hat{\mathbf{x}}_k$ in terms of $\hat{\mathbf{x}}_{k-1}$ and the new observation. The Kalman filtering equations are given below.

$$\left\{\begin{array}{rcl} \hat{\mathbf{x}}_0 = & E[\mathbf{x}_0] & \\ \mathbf{P}_0 = & E\left[(\mathbf{x}_0-\hat{\mathbf{x}}_0)(\mathbf{x}_0-\hat{\mathbf{x}}_0)^T\right] & \\ \mathbf{G}_k = & \mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T+R_k\right)^{-1} & \\ \mathbf{P}_k = & (I-\mathbf{G}_k\mathbf{C}_k)\,\mathbf{P}_{k-1} & \\ \hat{\mathbf{x}}_k = & \hat{\mathbf{x}}_{k-1}+\mathbf{G}_k\,(y_k-\mathbf{C}_k\hat{\mathbf{x}}_{k-1}) & \end{array}\right. \tag{2.5}$$

$\mathbf{z}_k = y_k - \mathbf{C}_k\hat{\mathbf{x}}_{k-1}$ is generally called the innovations sequence and $\mathbf{G}_k$ is called the Kalman gain vector. It can be shown that

$$\mathbf{P}_k = E\left[(\mathbf{x}-\hat{\mathbf{x}}_k)(\mathbf{x}-\hat{\mathbf{x}}_k)^T\right] \tag{2.6}$$

so $\mathbf{P}_k$ is the covariance matrix of the estimation error at time k.

2.3 Q-Ball Linear Model

QBI has the advantage over diffusion tensor imaging of being model independent [36]. He showed that it was possible to reconstruct a smoothed version of the diffusion orientation distribution function (ODF), $\psi(\mathbf{g}) = \int_0^\infty P(\mathbf{g}r)\mathrm{d}r$, directly from a single shell HARDI acquisition. The method involves viewing the HARDI acquisition signals as samples of a function on the sphere and taking the Funk-Radon transform (FRT) to arrive at the diffusion ODF. Many methods have been proposed to compute the FRT of the HARDI signal and most recently, [13] proposed a fast and robust analytical method to estimate the ODF by decomposing the HARDI signal onto a spherical harmonic basis. As in [27, 29], we will use this method because it is robust with respect to noise and well-suited for real-time applications.

Hence, letting Y_ℓ^m denote the SH of order ℓ and degree m ($m = -\ell, ..., \ell$) in the standard basis and Y_j ($j(\ell, m) = (\ell^2+\ell+2)/2+m$) be the SH in the modified real and symmetric basis [11, 13], the HARDI signal can now be expressed with the linear observation model

$$S(\theta_i,\phi_i) = \sum_{j=1}^{n}\tilde{x}_j Y_j(\theta_i,\phi_i) = \tilde{\mathbf{C}}_i\tilde{\mathbf{x}} \tag{2.7}$$

where $n = (L+1)(L+2)/2$ is the number of terms in the modified spherical harmonic basis of order L, $S(\theta_i, \phi_i)$ is the measured HARDI signal in the direction given by the polar coordinates (θ_i, ϕ_i) and $\tilde{\mathbf{C}}_i = [Y_1(\theta_i\phi_i) \;\; Y_2(\theta_i, \phi_i) \dots Y_n(\theta_i, \phi_i)]$. Since we have a linear observation model, given any number of measurements, we can decompose the HARDI signal onto the spherical harmonic basis by computing the least squares solution from equation (2.4). However, as diffusion weighted images are typically very noisy, [11] have proposed a robust regularized estimation algorithm using the Laplace-Beltrami operator, which leads to the following regularized minimization equation,

$$M(\tilde{\mathbf{x}}) = \left(\tilde{\mathbf{y}}_k - \tilde{\mathbf{B}}_k\tilde{\mathbf{x}}\right)^T \left(\tilde{\mathbf{y}}_k - \tilde{\mathbf{B}}_k\tilde{\mathbf{x}}\right) + \lambda\tilde{\mathbf{x}}^T\tilde{\mathbf{L}}\tilde{\mathbf{x}}, \tag{2.8}$$

where $\tilde{\mathbf{B}}_k = [\tilde{\mathbf{C}}_1^T \;\; \tilde{\mathbf{C}}_2^T \;\; \dots \;\; \tilde{\mathbf{C}}_k^T]^T$ and $\tilde{\mathbf{y}}_k = [S(\theta_1, \phi_1) \;\; S(\theta_2, \phi_2) \dots S(\theta_k, \phi_k)]^T$ contains the HARDI measurements.

Now we are really interested in relating the ODF to the HARDI signal and as the spherical harmonic functions are eigenfunctions of the FRT, there is a simple relationship between the HARDI signal and the ODF in the spherical harmonic basis. In [13], the authors show that the spherical harmonic coefficients of the ODF, $\mathbf{x}$, are given by $\mathbf{x} = \frac{1}{S_0}\mathbf{P}\tilde{\mathbf{x}}$ where $\mathbf{P}$ is the $n \times n$ FRT matrix with diagonal elements p_{kk} given by $p_{kk} = 2\pi P_{\ell(j)}(0)$ and $P_{\ell(j)}(0)$[1] is the Legendre polynomial of degree ℓ evaluated at 0 Since we want to estimate the ODF directly from HARDI data, we can rewrite the minimization criterion (2.8) in terms of the spherical harmonic representation, $\mathbf{x}$, of the ODF. We have

$$M(\mathbf{x}) = (\mathbf{y}_k - \mathbf{B}_k\mathbf{x})^T (\mathbf{y}_k - \mathbf{B}_k\mathbf{x}) + \lambda\mathbf{x}^T\mathbf{L}\mathbf{x} \tag{2.9}$$

where $\mathbf{B}_k = \tilde{\mathbf{B}}_k\mathbf{P}^{-1}$ and $\mathbf{y}_k = \tilde{\mathbf{y}}_k/S_0$ and $\mathbf{L} = \mathbf{P}^{-1}\tilde{\mathbf{L}}\mathbf{P}^{-1}$. So by minimizing equation (2.9) we can robustly estimate the ODF directly from the HARDI signal. Now, although we have the same linear observation model as used in the Kalman filter, the minimization criterion is slightly different than equation (2.3) due to the additional regularization term. In the next section, we will address this problem in the Kalman filtering framework.

3 Kalman Filtering with Regularization

The Kalman filtering algorithm presented in section 2.2 recursively solves equation (2.3) at each time step k. However, as in the ODF estimation proposed by [13], we would like to consider the following minimization problem

$$\hat{\mathbf{x}}_k = \arg\min_{\mathbf{x}\in\mathbb{R}^n} (\mathbf{y}_k - \mathbf{B}_k\mathbf{x})^T\mathbf{W}_k^{-1}(\mathbf{y}_k - \mathbf{B}_k\mathbf{x}) + \lambda\mathbf{x}^T\mathbf{L}\mathbf{x} \tag{3.1}$$

which is slightly modified from equation (2.3) by the addition of $\lambda\mathbf{x}^T\mathbf{L}\mathbf{x}$ which is a quadratic regularization term on the state variable $\mathbf{x}$. The closed form solution to equation (3.1) is easily obtained by setting the partial derivatives equal to zero. This yields

$$\hat{\mathbf{x}}_k = (\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{y}_k \tag{3.2}$$

[1] $\ell(j)$ is the order associated with the j^{th} element of the spherical harmonic basis, ie: for $j = 1, 2, 3, 4, 5, 6, 7, \dots$ $\ell(j) = 0, 2, 2, 2, 2, 2, 4, \dots$

We would like to modify the Kalman filtering equations (2.5) so that we can recursively solve equation (3.2) with the addition of each new observation. Currently, we are only aware of the work by [27, 29] addressing this problem in practice for DT and ODF estimation with the Kalman filtering framework, although the idea was first mentioned in [3] for DTI reconstruction. We show in the next sections that the method in [27, 29] is actually *sub-optimal* for every time step except the last one. Then, in section 3.2, we present a new Kalman filtering method that deals with this regularization term with a very simple and elegant modification of the Kalman filtering equations from section 2.2.

3.1 Existing Method

In [27, 29], the authors suggest to start with the solution (3.2) and invert it to obtain a linear system, whose least squares solution coincides with equation (3.2). Let N be the total number of measurements in the MR acquisition. In [27, 29], they use $N = 200$ in their experiments and they take the noise covariance matrix as the identity (ie: $\mathbf{W}_N = I_{N\times N}$). The linear system they use is $\mathbf{y}_N = \mathbf{B}^+\mathbf{x} + \eta_N$ where $\mathbf{B}^+$ is defined as

$$\mathbf{B}^+ = \left((\mathbf{B}_N^T\mathbf{B}_N + \lambda\mathbf{L})^{-1}\mathbf{B}_N^T\right)^\dagger$$

. The symbol $(\cdot)^\dagger$ is used to denote the Moore-Penrose pseudo-inverse operator. Now, denote the k^{th} row of $\mathbf{B}^+$ by $\mathbf{C}_k^+$. By using the linear system $\mathbf{y}_N = \mathbf{B}^+\mathbf{x} + \eta_N$, [27, 29] are using the following observation equation

$$y_k = \mathbf{C}_k^+\mathbf{x} + \epsilon_k, \tag{3.3}$$

as input to the Kalman filter at each time step k. In theory, there is no reason why this is the correct incremental solver. Their results do show the Kalman filter estimation converging to the correct estimate after N iterations, but this is evident (with some thought) from the definition of $\mathbf{B}^+$. It is not clear, however, if any of the intermediate ODF estimations (ie: for $1 \leq k < N$) are actually minimizing the correct criterion (3.1) which is reproduced here.

$$M(\mathbf{x}) = (\mathbf{y}_k - \mathbf{B}_k\mathbf{x})^T\mathbf{W}_k^{-1}(\mathbf{y}_k - \mathbf{B}_k\mathbf{x}) + \lambda\mathbf{x}^T\mathbf{L}\mathbf{x} \tag{3.4}$$

In the next section, we analyze this approach and derive an explicit expression for the estimation at each iteration. In doing so, we show that this approach in fact *does not* minimize equation (3.1) at any iterations *except* for the last.

3.1.1 Analysis

In keeping with the exact method proposed in [27, 29], we assume the noise covariance matrix $\mathbf{W}_k$ is the identity matrix for all time steps k. Before doing any detailed analysis, we can make some notes just from the definition of $\mathbf{B}^+$ and the observation equation

$$y_k = \mathbf{C}_k^+\mathbf{x} + \epsilon_k. \tag{3.5}$$

As $\mathbf{C}_k^+$ is a row of $\mathbf{B}^+$, it is clear from definition of $\mathbf{B}^+$ that every observation matrix $\mathbf{C}_k^+$ depends on $\mathbf{B}_N$ and therefore on *all* N gradient acquisition directions. Hence, the acquisition of a given

diffusion gradient orientation will be modeled differently depending on how many acquisitions are required in the acquisition. Furthermore, this means that when applying the Kalman filter to this model, every estimation depends on the entire diffusion orientation set being used. Thus, changing some diffusion gradients at the end of the acquisition could influence the estimations from the beginning. This is definitely undesirable behavior for an incremental optimal estimator. Based on the experimental results in [27, 29] this method converges after all N acquisitions, but they do not provide validation to show how close the intermediate estimations are to the optimal intermediate estimations.

The best way to analyze the method from [27, 29] is to derive an explicit expression for the Kalman filter output at each iteration under this method. First, we derive in appendix A a closed form expression for $\mathbf{B}^+$. The result is

$$\begin{aligned} \mathbf{B}^+ &= \left((\mathbf{B}_N^T\mathbf{B}_N + \lambda\mathbf{L})^{-1}\mathbf{B}_N^T\right)^\dagger \\ &= \mathbf{B}_N\left(I_{n\times n} + \lambda\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\right). \end{aligned} \tag{3.6}$$

Now let $\mathbf{D}_k = [\mathbf{C}_1^{+T}\ \mathbf{C}_2^{+T}\ \ldots\ \mathbf{C}_k^{+T}]^T$ be the $k \times n$ matrix containing the first k observation matrices (or rows of $\mathbf{B}^+$). Using the result in equation (3.6), we can write $\mathbf{D}_k$ as

$$\mathbf{D}_k = \mathbf{B}_k\left(I_{n\times n} + \lambda\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\right) \tag{3.7}$$

This formula comes from the fact that taking the first k rows of the product of two matrices (ie: $\mathbf{D}_N = \mathbf{B}_{N\times n}\mathbf{A}_{n\times n}$) is equivalent to taking the first k rows of the first matrix, $\mathbf{B}_{k\times n}$ and multiplying them by the second matrix $\mathbf{A}_{n\times n}$ (ie: $\mathbf{D}_k = \mathbf{B}_{k\times n}\mathbf{A}_{n\times n}$). Now consider the linear system

$$\mathbf{y}_k = \mathbf{D}_k\mathbf{x} + \eta_k. \tag{3.8}$$

This linear system relates all the observations up to time k to the state $\mathbf{x}$ using the observation equation (3.3) used in [27, 29]. So, after k observations, the "optimal" state estimate will be given by the least squares solution to this linear system. That is, the output of the Kalman filter, $\hat{\mathbf{x}}_k$ at iteration k is given by

$$\hat{\mathbf{x}}_k = \left(\mathbf{D}_k^T\mathbf{D}_k\right)^{-1}\mathbf{D}_k^T\mathbf{y}_k$$

This can be simplified (see appendix B) to

$$\hat{\mathbf{x}}_k = \left(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{B}_k^T\mathbf{B}_k\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\right)^{-1}\mathbf{B}_k^T\mathbf{y}_k \tag{3.9}$$

A quick comparison with equation (3.2) shows that this is *not* the optimal estimate unless

$$\mathbf{B}_k^T\mathbf{B}_k = \mathbf{B}_N^T\mathbf{B}_N$$

or $\lambda = 0$. The former is only satisfied when $k = N$. Thus, we have shown that this method is sub-optimal at every estimation step except for the last. We should note that we do not mean to imply that the authors of [27, 29] are mistaken about their results. Since the state estimate will converge

to the optimal estimate by the last diffusion gradient acquisition, there will be iterations, especially near the end, where the estimate is very close to optimal. However, there will most definitely be many intermediate iterations where the estimation is far from optimal and this was not investigated in [27, 29]. The implications of this sub-optimality should be made perfectly clear. If the expected number of acquisitions is set to $N = 200$ and the acquisition was stopped before completion (say at $k = 100$) the ODF estimates obtained from the Kalman filtering method of [27, 29] will *not* be minimizing the correct criterion from equation (3.1). They will in fact be minimizing

$$M(\mathbf{x}) = (\mathbf{y}_k - \mathbf{D}_k\mathbf{x})^T (\mathbf{y}_k - \mathbf{D}_k\mathbf{x}) \tag{3.10}$$

which yields a *different minimizing argument* (3.9) compared to the correct criterion (3.1) for $k < 200$.

3.1.2 Some Naive Approaches

There are some naive approaches one might take to correct the method from [27, 29]. The first comes from the observation that this method could be "corrected" by replacing N by k in the definition of $\mathbf{D}_k$ (3.7). Then the least squares solution to $\mathbf{y}_k = \mathbf{D}_k\mathbf{x}$ would be equal to the correct solution from equation (3.2). This in fact would be true, but it is impossible to apply Kalman filtering in this situation. Recall that the Kalman filter is derived from the linear observation model (2.1); it depends heavily on the fact that $\mathbf{B}_{k-1}$ and $\mathbf{B}_k$ differ by *only* the addition of one row $\mathbf{C}_k$. If we were to make this modification to $\mathbf{D}_k$, then $\mathbf{D}_k$ would differ from $\mathbf{D}_{k-1}$ by *much more* than just the addition of a single row. Therefore, there would be *no system of observation models* of the form $y_k = \mathbf{C}_k\mathbf{x} + \epsilon_k$ such that $\mathbf{D}_k = [\mathbf{C}_1^T\ \mathbf{C}_2^T\ \dots\ \mathbf{C}_k^T]^T$ for all k. This makes it impossible to derive a recursive relationship between $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_{k-1}$ and hence impossible to apply the Kalman filter.

The second approach is similar to the first; since we cannot apply the Kalman filter recursively with the modification described above, we could apply the Kalman filter to *all* of the data when we get a new observation. But by doing this, the solution is no longer obtained in a recursive way and real-time processing is impossible. For example, to compute the estimation at iteration k the Kalman filter would have to be iterated over all the previously recorded observations which is k *times more expensive* than the truly recursive Kalman filter and thus not suitable for real-time applications. In fact, there is little difference between this method and computing the offline estimation at each iteration.

The purpose of this discussion is to clarify that there are no simple modifications that could correct the method from [27, 29]. In the next section, we will present a modification of the Kalman filter that allows the regularization term to be incorporated while still producing the optimal estimation at *every* time step.

3.2 Proposed Method

We propose a new Kalman filtering method that will incorporate the regularization term in the initial condition of $\mathbf{P}_0$. Recall that we wish to solve the following equation (3.1)

$$\hat{\mathbf{x}}_k = \arg\min_{\mathbf{x}\in\mathbb{R}^n} (\mathbf{y}_k - \mathbf{B}_k\mathbf{x})^T\mathbf{W}_k^{-1}(\mathbf{y}_k - \mathbf{B}_k\mathbf{x}) + \lambda\mathbf{x}^T\mathbf{L}\mathbf{x}$$

in a recursive manner. The Kalman filtering equations for our proposed approach are

$$\begin{cases} \hat{\mathbf{x}}_0 = & E[\mathbf{x}_0] \\ \tilde{\mathbf{P}}_0 = & E\left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T(\mathbf{x}_0 - \hat{\mathbf{x}}_0)\right] \\ \mathbf{P}_0 = & \left(\tilde{\mathbf{P}}_0^{-1} + \lambda\mathbf{L}\right)^{-1} \\ \mathbf{G}_k = & \mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T + R_k\right)^{-1} \\ \mathbf{P}_k = & \left(I - \mathbf{G}_k\mathbf{C}_k\right)\mathbf{P}_{k-1} \\ \hat{\mathbf{x}}_k = & \hat{\mathbf{x}}_{k-1} + \mathbf{G}_k\left(y_k - \mathbf{C}_k\hat{\mathbf{x}}_{k-1}\right) \end{cases} \tag{3.11}$$

The state prediction correction and covariance recursive update equations are unchanged and the regularization term only appears in the initial condition. To derive these equations, we will start with the solution to equation (3.1) which we reproduce here for convenience

$$\hat{\mathbf{x}}_k = (\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{y}_k$$

We now apply Kalman filtering in this context by expressing $\hat{\mathbf{x}}_k$ in terms of $\hat{\mathbf{x}}_{k-1}$. We go through the detailed derivation. First note that

$$\begin{aligned} \mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k + \lambda\mathbf{L} &= \begin{bmatrix}\mathbf{B}_{k-1}^T & \mathbf{C}_k^T\end{bmatrix}\begin{bmatrix}\mathbf{W}_{k-1}^{-1} & 0 \\ 0 & R_k^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{B}_{k-1} \\ \mathbf{C}_k\end{bmatrix} + \lambda\mathbf{L} \\ &= \mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{B}_{k-1} + \mathbf{C}_k^T R_k^{-1}\mathbf{C}_k + \lambda\mathbf{L} \end{aligned}$$

and

$$\begin{aligned} \mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{y}_k &= \begin{bmatrix}\mathbf{B}_{k-1}^T & \mathbf{C}_k^T\end{bmatrix}\begin{bmatrix}\mathbf{W}_{k-1}^{-1} & 0 \\ 0 & R_k^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{y}_{k-1} \\ y_k\end{bmatrix} \\ &= \mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{y}_{k-1} + \mathbf{C}_k^T R_k^{-1} y_k \end{aligned}$$

Using the above 2 equations with equation (3.2) we get

$$\begin{aligned} & \left(\mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{B}_{k-1} + \mathbf{C}_k^T R_k^{-1}\mathbf{C}_k + \lambda\mathbf{L}\right)\hat{\mathbf{x}}_k \\ = \quad & \mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{y}_{k-1} + \mathbf{C}_k^T R_k^{-1} y_k \end{aligned} \tag{3.12}$$

and by direct computation, we have

$$\begin{aligned} & \left(\mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{B}_{k-1} + \mathbf{C}_k^T R_k^{-1}\mathbf{C}_k + \lambda\mathbf{L}\right)\hat{\mathbf{x}}_{k-1} \\ = \quad & \left(\mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{B}_{k-1} + \lambda\mathbf{L}\right)\hat{\mathbf{x}}_{k-1} + \mathbf{C}_k^T R_k^{-1}\mathbf{C}_k\hat{\mathbf{x}}_{k-1} \\ = \quad & \mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{y}_{k-1} + \mathbf{C}_k^T R_k^{-1}\mathbf{C}_k\hat{\mathbf{x}}_{k-1} \end{aligned} \tag{3.13}$$

Subtracting equations (3.12) and (3.13) we have

$$\left(\mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{B}_{k-1} + \mathbf{C}_k^T R_k^{-1}\mathbf{C}_k + \lambda\mathbf{L}\right)(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1}) = \mathbf{C}_k^T R_k^{-1}\left(y_k - \mathbf{C}_k\hat{\mathbf{x}}_{k-1}\right)$$

Now, let $\mathbf{G}_k = \left(\mathbf{B}_{k-1}^T\mathbf{W}_{k-1}^{-1}\mathbf{B}_{k-1} + \mathbf{C}_k^T R_k^{-1}\mathbf{C}_k + \lambda\mathbf{L}\right)^{-1}\mathbf{C}_k^T R_k^{-1}$ and we have the recursive equation

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{G}_k\left(y_k - \mathbf{C}_k\hat{\mathbf{x}}_{k-1}\right) \tag{3.14}$$

which is exactly the same as the linear Kalman filter update equation and the regularization constraint is incorporated in the Kalman gain matrix. To complete the derivations, we need to derive a recursive equation for $\mathbf{G}_k$. Let $\mathbf{P}_k = \left(\mathbf{B}_{k-1}^T \mathbf{W}_{k-1}^{-1} \mathbf{B}_{k-1} + \mathbf{C}_k^T R_k^{-1} \mathbf{C}_k + \lambda \mathbf{L}\right)^{-1}$ so that $\mathbf{G}_k = \mathbf{P}_k \mathbf{C}_k^T R_k^{-1}$ and note that

$$\begin{aligned} \mathbf{P}_k &= \left(\mathbf{B}_{k-1}^T \mathbf{W}_{k-1}^{-1} \mathbf{B}_{k-1} + \mathbf{C}_k^T R_k^{-1} \mathbf{C}_k + \lambda \mathbf{L}\right)^{-1} \\ &= \left(\mathbf{B}_k^T \mathbf{W}_k^{-1} \mathbf{B}_k + \lambda \mathbf{L}\right)^{-1} \end{aligned} \tag{3.15}$$

which leads to the recursive equation

$$\mathbf{P}_k^{-1} = \mathbf{P}_{k-1}^{-1} + \mathbf{C}_k^T R_k^{-1} \mathbf{C}_k \tag{3.16}$$

Using the result in appendix C, we can invert the above equation and we have

$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1} \mathbf{C}_k^T \left(\mathbf{C}_k \mathbf{P}_{k-1} \mathbf{C}_k^T + R_k\right)^{-1} \mathbf{C}_k \mathbf{P}_{k-1}$$

Now, it can be shown (see appendix E) that

$$\mathbf{G}_k = \mathbf{P}_k \mathbf{C}_k^T R_k^{-1} = \mathbf{P}_{k-1} \mathbf{C}_k^T \left(\mathbf{C}_k \mathbf{P}_{k-1} \mathbf{C}_k^T + R_k\right)^{-1}$$

which gives us the following recursive equations for $\mathbf{G}_k$ and $\mathbf{P}_k$

$$\begin{aligned} \mathbf{G}_k &= \mathbf{P}_{k-1} \mathbf{C}_k^T \left(\mathbf{C}_k \mathbf{P}_{k-1} \mathbf{C}_k^T + R_k\right)^{-1} \\ \mathbf{P}_k &= \left(I - \mathbf{G}_k \mathbf{C}_k\right) \mathbf{P}_{k-1} \end{aligned}$$

We can see that these are the same as for the linear Kalman filter without regularization but it is important to note that although the recursive update equations for $\mathbf{P}_k$ and $\mathbf{G}_k$ do not change with the added regularization term, the matrices themselves are different and this is taken into account by the initial condition $\mathbf{P}_0$ which is no longer the covariance matrix of $\mathbf{x}_0$. To discuss the initial condition, let $\hat{\mathbf{x}}_k^0$ be the optimal estimate *without* regularization (ie: $\lambda = 0$) and let $\tilde{\mathbf{P}}_k$ be the corresponding sequence of covariance matrices for the linear Kalman filter. In appendix D it is shown that

$$\tilde{\mathbf{P}}_k = E\left[(\mathbf{x} - \hat{\mathbf{x}}_k^0)(\mathbf{x} - \hat{\mathbf{x}}_k^0)^T\right] = \left(\mathbf{B}_k^T \mathbf{W}_k^{-1} \mathbf{B}_k\right)^{-1}$$

Recall also that we have defined $\mathbf{P}_k$ as

$$\mathbf{P}_k = \left(\mathbf{B}_k^T \mathbf{W}_k^{-1} \mathbf{B}_k + \lambda \mathbf{L}\right)^{-1}$$

Using these two equations, we have the relation

$$\mathbf{P}_k^{-1} = \tilde{\mathbf{P}}_k^{-1} + \lambda \mathbf{L}$$

So, even though $\mathbf{P}_k$ does not have the usual physical interpretation as the error covariance when $\lambda \neq 0$, we have found a relation between $\mathbf{P}_k$ and $\tilde{\mathbf{P}}_k$. A special case of the above formula is

$$\mathbf{P}_0^{-1} = \tilde{\mathbf{P}}_0^{-1} + \lambda \mathbf{L}$$

which gives us

$$\mathbf{P}_0 = \left(\tilde{\mathbf{P}}_0^{-1} + \lambda \mathbf{L}\right)^{-1} \tag{3.17}$$

So the initial matrix $\mathbf{P}_0$ can be expressed in terms of the initial matrix $\tilde{\mathbf{P}}_0$. Combining all these results, we have the Kalman filtering equations with a quadratic regularization term

$$\begin{cases} \hat{\mathbf{x}}_0 = & E[\mathbf{x}_0] \\ \tilde{\mathbf{P}}_0 = & E\left[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T(\mathbf{x}_0 - \hat{\mathbf{x}}_0)\right] \\ \mathbf{P}_0 = & \left(\tilde{\mathbf{P}}_0^{-1} + \lambda \mathbf{L}\right)^{-1} \\ \mathbf{G}_k = & \mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T + R_k\right)^{-1} \\ \mathbf{P}_k = & \left(I - \mathbf{G}_k\mathbf{C}_k\right)\mathbf{P}_{k-1} \\ \hat{\mathbf{x}}_k = & \hat{\mathbf{x}}_{k-1} + \mathbf{G}_k\left(y_k - \mathbf{C}_k\hat{\mathbf{x}}_{k-1}\right) \end{cases}$$

By implementing these Kalman filtering equations we can recursively solve equation (3.1) at every time step k. As we started the derivations with the correct minimization criterion we are assured that this will produce the optimal estimation at every iteration. In the experimental Section , we show our optimal solution on real HARDI data. We illustrate that our proposed real-time method is equivalent in terms of QBI estimation accuracy to the standard off-line processing techniques and outperforms the existing solution of [27, 29], especially at an early stage of the acquisition.

At this stage, we have revisited and analyzed the proposed Kalman filtering algorithm for ODF estimation and we have shown it is actually *sub-optimal* and *not* recursively minimizing the intended criterion due to the Laplace-Beltrami regularization term. More importantly, we have also derived the correct equations that allow to recursively estimate the optimal ODF at each iteration i.e we have shown how to update the optimal ODF each time we have a new measurement. Hence, to take full advantage of this new framework, it is of interest to see if successive gradient encoding directions can also be computed in real-time to roughly have a uniform distribution on the sphere. In the next section, we will tackle this problem of finding the optimal choice of the diffusion gradient orientation sets and propose a fast algorithm to incrementally compute gradient orientation sets whose partial subsets are almost uniform.

4 Diffusion Gradient Orientation Sets

The choice of diffusion orientations sets for diffusion MRI has been extensively studied in the literature. For single shell HARDI acquisitions, a standard approach is to acquire N measurements that are uniformly distributed on the unit sphere so that the errors in measures derived from the diffusion weighted images are independent of tissue orientation. The problem of distributing N points uniformly on a sphere is not a new problem in diffusion MRI; it has been studied by mathematicians and physicists for at least 100 years. In fact, British Physicist J. J. Thomson proposed the problem of determining the minimum energy configuration of N classical electrons distributed on the surface of a sphere in his 1904 paper [33]. It has since been termed *The Thomson Problem* and has been thoroughly studied since [15]. There has been a significant amount of research on this within the context

of diffusion MRI. We first describe the state of the art methods and then propose a fast algorithm that achieves similar results.

4.1 Existing Methods

In the context of diffusion MRI the problem of uniformly distributing points on a sphere needs to be slightly modified due to the fact that the diffusion of water is symmetric meaning that a measurement in the direction $\mathbf{g} = [x, y, z]$ is equivalent to a measurement in the direction $-\mathbf{g}$. To account for this, [18] proposed a slight variation on *The Thomson Problem* where each gradient direction is modeled by an antipodal pair of charged particles and the desired configuration is the one obtained by minimizing the sum of the forces on the particles. [24] proposed a similar approach by maximizing the minimum distance between any 2 charged particles. We define the electrostatic energy between 2 orientations (or 2 pairs of charges) as

$$E(\mathbf{g}_i, \mathbf{g}_j) = \frac{1}{||\mathbf{g}_i + \mathbf{g}_j||} + \frac{1}{||\mathbf{g}_i - \mathbf{g}_j||} \tag{4.1}$$

and therefore, the total electrostatic energy between all pairs of orientations is

$$E_N(\mathbf{g}_1, \ldots, \mathbf{g}_N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} E(\mathbf{g}_i, \mathbf{g}_j) \tag{4.2}$$

In this work, we define an electrostatic point-set of size N as the collection of directions that minimizes equation (4.2). [10] have computed the electrostatic point-sets for $3 \leq N \leq 150$ and publicly provide them as part of the Camino Diffusion MRI Toolkit ([9]). They computed these point-sets using Levenberg-Marquardt optimization starting from 500 different initial conditions and taking the lowest minimum. In this work, we will use these electrostatic point-sets as a reference for comparing different diffusion orientation sets.

The electrostatic point-sets approach can produce optimized orientation sets when one considers all the diffusion acquisitions, however, if the acquisition is aborted before completion due to motion of the patient, the subset of acquired orientations will be directionally biased and will be unusable for DT or ODF estimation. [10] and [14] have proposed algorithms for generating uniform point-sets whose ordered subsets are also approximately uniform. Thus, if a acquisition is aborted before completion, the acquired orientation set will be fairly uniform and the acquisition can be partially saved.

[14] propose to order the orientations by acquisition time and make a modification of equation (4.2) to ensure that orientations that are close together in the acquisition sequence are more separated spatially. To do this, they add an interaction weight α_{ij} to the electrostatic energy

$$E_N(\mathbf{g}_1, \ldots, \mathbf{g}_N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \alpha_{ij} E(\mathbf{g}_i, \mathbf{g}_j). \tag{4.3}$$

When the interaction weight is constant ($\alpha_{ij} = 1$) the minimizing point-set will be equal to the electrostatic point-set modulo a permutation of the acquisition order. For an acquisition where corruption is a possibility due to motion of the patient, they propose to set $\alpha_{ij} = 1$ for orientations close

in time and $\alpha_{ij} < 1$ for orientations that are distant in time. The idea is that by placing a higher electrostatic penalty on gradients that are positioned close together in the acquisition sequence, we can encourage ordered subsets of gradient directions to be more uniformly distributed. This method can produce very nice results, but one drawback is that a point-set that minimizes equation (4.3) does not necessarily minimize (4.2). Also, [10] empirically determined that the incorporation of interaction weights increases the number of local minima making the global minimum more difficult to find.

[10] have proposed to take the optimal electrostatic point-sets and choose an optimal ordering based on the sum of the energies of the partial subsets

$$f = \sum_{P=6}^{N} E_P(\mathbf{g}_1, \ldots, \mathbf{g}_P) P^{-2}. \tag{4.4}$$

The electrostatic energy of P isotropically distributed pairs is approximately proportional to P^2 so they have chosen the normalization factor P^{-2} to ensure that each subset of orientations contributes similarly to the objective function. As the points themselves are fixed, the minimization is done only over the ordering of the points. This can be done by exhaustive search for small N but by $N = 18$ the search space becomes absurdly large, so [10] propose to perform the minimization by simulated annealing and they provide their software for public use through the Camino Diffusion MRI Toolkit ([9]). [10] conclude that ordering the acquisition scheme has a significant impact on the quality of partial acquisitions. They also compare their method to [14] and conclude that both methods yield similar results.

4.2 An incremental algorithm to generate orientation sets

We propose a fast recursive algorithm to incrementally generate orientation sets whose ordered subsets are approximately uniform. What we propose is not an improvement or a replacement for the existing algorithms as they are already very effective at what they do. What we propose is a different approach altogether. The algorithms proposed in [14] and [10] perform their respective minimizations over all the points in the orientation set. Thus, the algorithms are very slow and the global minima are difficult to find as there are many local minima present. Our idea comes from the observation that this problem naturally leads to a recursive solution; instead of minimizing over all the orientations at once, we propose to minimize incrementally over each ordered subset. Clearly we will not be able to minimize the same criterion from [14] or [10], but we will show that we can generate comparably uniform point-sets with an algorithm that whose complexity is linear with respect to the number of points in the orientation set. Furthermore, inspired by [10] we will show that our approach can be slightly modified and used to order an existing electrostatic point-set so that the ordered subsets are roughly uniformly distributed. Our proposed methods will be further described and compared to existing methods in the next few sections.

Fast Recursive Computation of Orientation Sets

We propose a new and fast algorithm for incrementally computing uniformly distributed orientation sets whose partial subsets are also uniform. This method is defined recursively: given a set of k

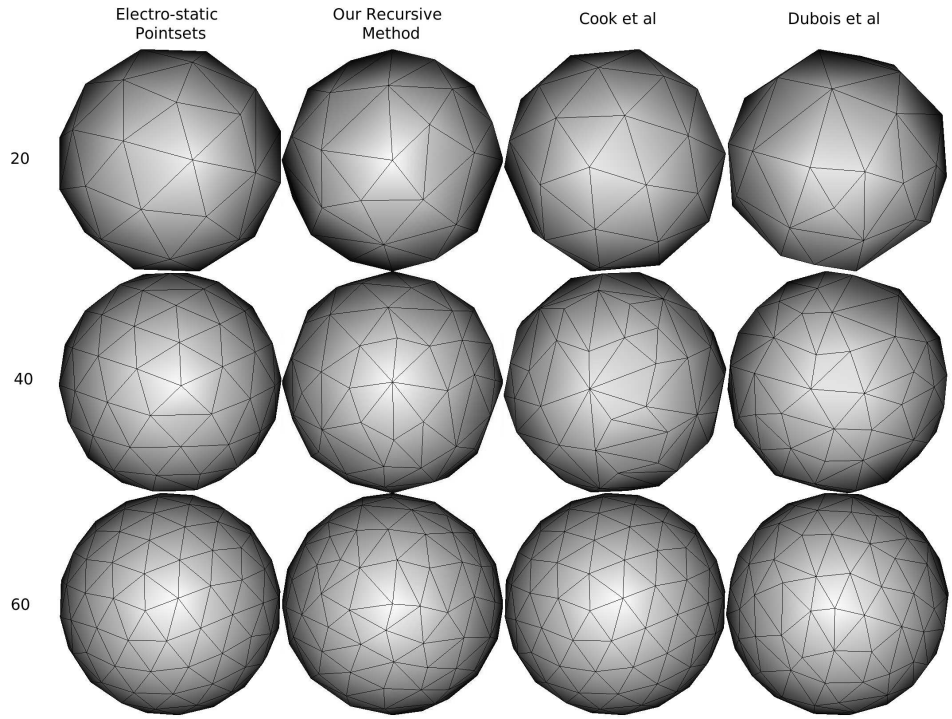


Figure 1: Comparison of point-sets: (left to right) Electrostatic point-sets, our recursive method, method of [10], method of [14]. Point-sets are shown for (top to bottom) 20, 40 and 60 points. The point-sets from [10] and [14] were specifically optimized for 60 orientations. Our method was recursively generated starting with 1 gradient direction.

orientations, choose direction $k+1$ so as to minimize the incremental electrostatic energy. We should note that this method will not minimize the criterion from [10] or [14] over all $k+1$ orientations. We are, however, minimizing the total electrostatic energy of the $k+1$ orientations given that the first k *must remain fixed*.

To further describe our method, let $\Sigma_k = \{\mathbf{g}_1, \ldots, \mathbf{g}_k\}$ be a set of k orientations, let $\mathbf{g}(\theta, \phi)$ be the unit vector in the direction given by the spherical coordinates (θ, ϕ) and let $\Psi_k(\theta, \phi)$ be defined as

$$\Psi_k(\theta, \phi) = \sum_{i=1}^{k} E\left(\mathbf{g}_i, \mathbf{g}(\theta, \phi)\right) \tag{4.5}$$

Thus, $\Psi_k(\theta, \phi)$ gives the incremental electrostatic energy associated with choosing $\mathbf{g}(\theta, \phi)$ as the next encoding direction. Now, given a set of k orientations, Σ_k, we compute the next orientation by

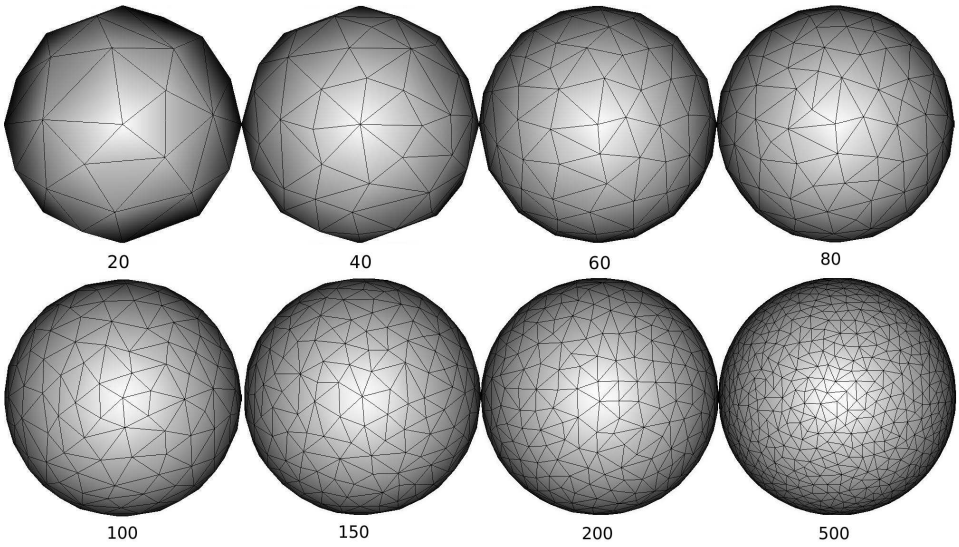


Figure 2: Our point-set generation method was used recursively to generate a set of 500 orientations starting with a single gradient. Shown in the figure are the ordered subsets corresponding to 20, 40, 60, 80, 100, 150, 200, and 500 orientations.

the formula $\mathbf{g}_{k+1} = \mathbf{g}(\hat{\theta}_k, \hat{\phi}_k)$ where

$$(\hat{\theta}_k, \hat{\phi}_k) = \underset{(\theta,\phi)\in[0,\pi)\times[0,\pi)}{\arg\min} \Psi_k(\theta, \phi) \tag{4.6}$$

Thus $\Sigma_{k+1} = \{\Sigma_k, \mathbf{g}_{k+1}\}$. Now there are many algorithms that could be used to perform this minimization, but since the search space is relatively small, we will use an exhaustive search. First, however, we should note that if we were to implement an exhaustive search on equation (4.5), the computational complexity of the function Ψ_k would grow linearly with k. To increase the performance of our algorithm, we will exploit a nice recursive relationship between Ψ_k and Ψ_{k-1}:

$$\Psi_k(\theta, \phi) = \Psi_{k-1}(\theta, \phi) + E\left(\mathbf{g}_k, \mathbf{g}(\theta, \phi)\right) \tag{4.7}$$

Using the above recursive relationship, we can maintain a sampled representation of Ψ_k which is updated after each new gradient direction is computed. Through experimentation, we have determined that sampling with a precision of 0.01rad (0.57°) yields the correct minimum and only requires a fraction of a second to update Ψ_k and perform the minimization. With regards to the memory requirements of our algorithm, since we are sampling with a precision of 0.01, we need to store $(\pi/0.01)^2 \approx 98700$ samples. If we store the samples with 32-bit floating point precision, the memory requirement is approximately 400 kB which is very modest by today's standards.

We have compared our methods with the state of the art point-sets using sizes of $N = 60, 150$. For the comparison with [10], we used the ordered electrostatic point-set for $N = 60$ that is pro-

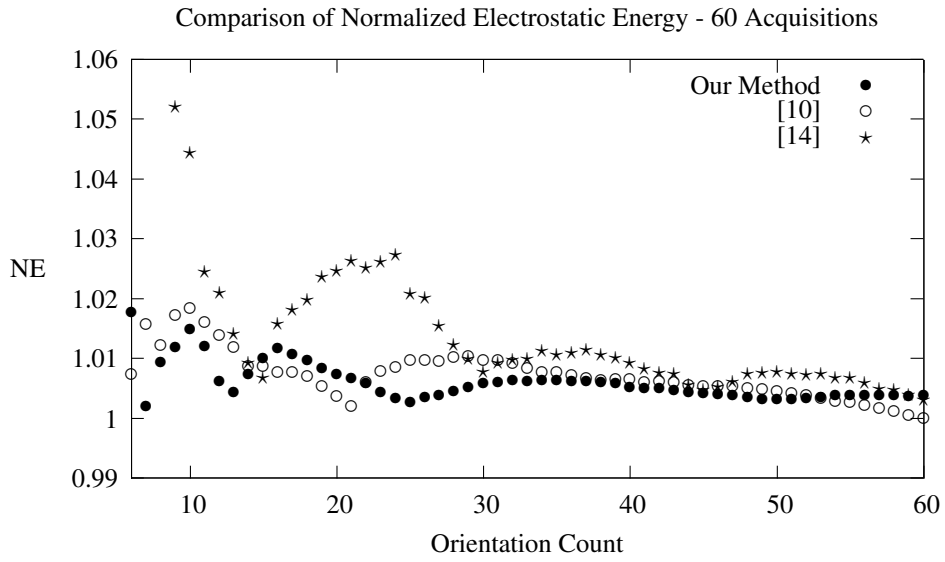


Figure 3: A comparison of the normalized energy of our recursive method, [14] and [10] over 60 acquisitions.

vided with the Camino Diffusion MRI Toolkit ([9]). For $N = 150$, we computed the ordering using the code they provided with Camino. We used the default initial conditions and allowed the simulated annealing to run for 137 hours until the temperature was at $2.67e^{-6}$ before we terminated the program. For the comparison with [14], for $N = 60$, we have used the orientation set that is given on page 140 of their paper(entitled A_{60}). Figure 1 shows a visual comparison of the 3 algorithms alongside the optimal electrostatic point-sets for $N = 60$. Our algorithm was initialized with $\mathbf{g}_1 = [1 \;\; 0 \;\; 0]^T$ and all directions $\mathbf{g}_k,\ k > 1$ were generated recursively. In figure 1, we terminated our algorithm at $N = 60$ to do a proper comparison. Figure 2 shows the results if we continue our algorithm to $N = 500$.

We have also done a numerical comparison of the normalized electrostatic energy (NE)between the three algorithms which is shown in figures 3 and 4. The normalization was done by dividing the electrostatic energy of each algorithm's ordered subsets by the optimal energy of the electrostatic point-set of equal size. This provides a good measure of how uniform the ordered subsets are; in comparing two point-sets, a value closer to one indicates a more uniform distribution of gradient acquisitions. From figures 3 and 4 we can see that for both $N = 60$ and $N = 150$ our algorithm performs comparably to the existing methods. Our algorithm yields more uniform point-sets at certain points and less uniform sets at other points.

The one glaring difference is that our algorithm is not optimized for a certain number of orientations, so it will never be exactly optimal whereas the method from [10] produces the optimal electrostatic point-set, by design, at the end of the acquisition. Intuitively, we know from the Kalman

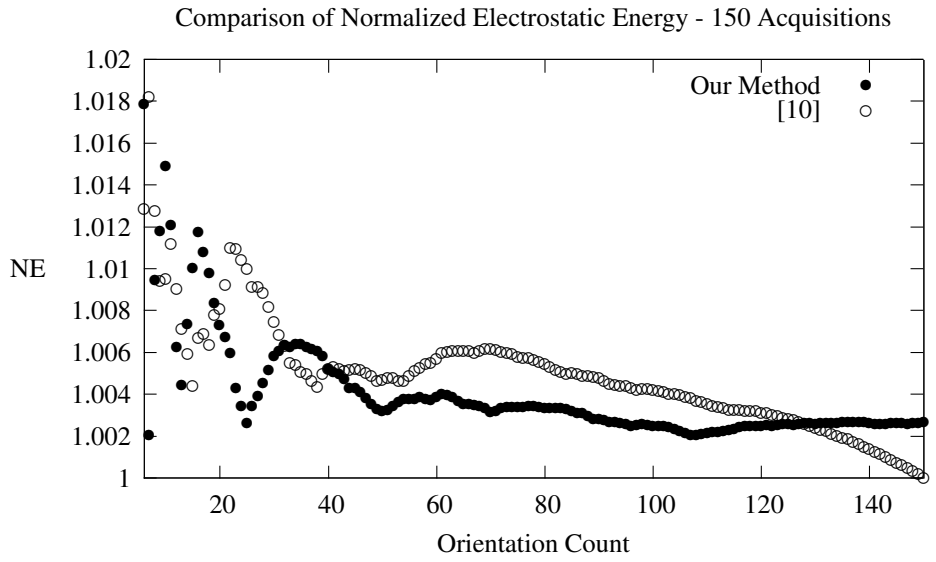


Figure 4: A comparison of the normalized energy of our recursive method and [10] over 150 acquisitions.

filtering results that, after a certain point, each additional acquisition will affect the diffusion tensor or ODF estimation less than the previous acquisition. This implies that as the number of orientations grows, the uniformity (or lack thereof) of the orientation set will have less of an influence on the DT or ODF estimation. It is very possible that the small sub-optimality that our algorithm shows at the end of the acquisition would have little influence over the results of the acquisition. However, this is an open issue that can really only be proven experimentally and it would depend on many different variables.

The advantage our algorithm has over the current techniques is that its linear complexity enables the algorithm to be run in real-time during an MR acquisition. The recursive ODF estimation approach we proposed in the previous section could also take a great benefit from this algorithm. We could start a acquisition acquisition just with the minimum number of gradients directions and an initial estimate of the ODF and then all the rest, including the next gradient directions and the ODF estimates, are recursively and optimally determined, allowing the acquisition to be stopped as soon as desired or at any iteration with the optimal ODF estimate. Also, the linear complexity of our algorithm makes it possible to generate large orientation sets (ie: 1000+ points) very quickly whereas the existing methods would quickly become too slow to be useful. Therefore, we think that this work opens new and interesting opportunities for clinicians and researchers.

Fast Ordering of Orientation Sets

We have proposed in the previous subsection a fast algorithm to compute diffusion gradient orientation sets whose partial subsets are roughly uniform. Now, inspired by [10], we can also apply our method to the problem of ordering an existing electrostatic point-set of size N. Thus we can guarantee all the ordered subsets of diffusion orientations are approximately uniform and that after N acquisitions, the point-set is exactly equal to the electrostatic point-set of size N. Let Σ_N be an electrostatic point-set of size N. Recall we defined an electrostatic point-set as the collection of orientations that minimizes equation (4.2). Now, the goal is to order the set Σ_N so that every ordered subset of orientations is roughly uniform. We select the first orientation, $\mathbf{g}_1$ at random and define the next $N-1$ recursively:

$$\mathbf{g}_{k+1} = \underset{\mathbf{g}\in\Sigma_N}{\arg\min} \sum_{i=1}^{k} E(\mathbf{g}_i, \mathbf{g}) \tag{4.8}$$

This algorithm will likely not produce the optimal ordering that [10] have computed, but it can produce a much superior ordering to the conventional orderings, with the added benefit that the complexity of the algorithm is linear with respect to the number of points in the orientation set.

We have compared this method for $N = 60, 150$ to the point-sets from [10] that were described in the previous section. We have also shown a random ordering of the points to demonstrate the added value of ordering. The results are shown in figure 5. We can see that our method performs very well compared to the random ordering. Furthermore, it is very similar to [10] yet usually produces a slightly less uniform point-set at each iteration. This should be expected; our method is a classic greedy algorithm simply minimizing the incremental energy at each iteration. The advantage of our algorithm is the linear complexity, thus our algorithm may be suitable for very large orientation sets where the simulated annealing proposed in [10] takes far too long to converge.

Further Possibilities

So far, we have presented a fast algorithm for computing a roughly uniform set of orientations with roughly uniform subsets. Our algorithm was defined recursively and we have always chosen only the first gradient and the rest were determined recursively. However, we have the flexibility to select any initial condition of as many orientations as we would like. This gives us some freedom to design orientation sets that could suit certain needs. In fact we can even mix our algorithm with the algorithms from [10] and [14] by selecting one of their optimized point-sets as an initial condition for our algorithm. For example, consider a scenario where we might desire 150 to 200 measurements, but we require that the first 60 acquisitions be exactly uniformly distributed. One solution is to use the method from [10] to order the electrostatic point-set for $N = 60$ and then recursively apply our algorithm for all $N \geq 61$. We have done this and the normalized energy is plotted in figure 6. We can see that the hybrid method does indeed achieve the optimal electrostatic set at $N = 60$, however, we can see that there is a trade-off here; the point-set may be optimal at $N = 60$, but because we forced this optimality, it becomes sub-optimal between iterations $N = 65$ and $N = 140$. There are doubtlessly countless ways that our algorithm could be combined with others, but we should

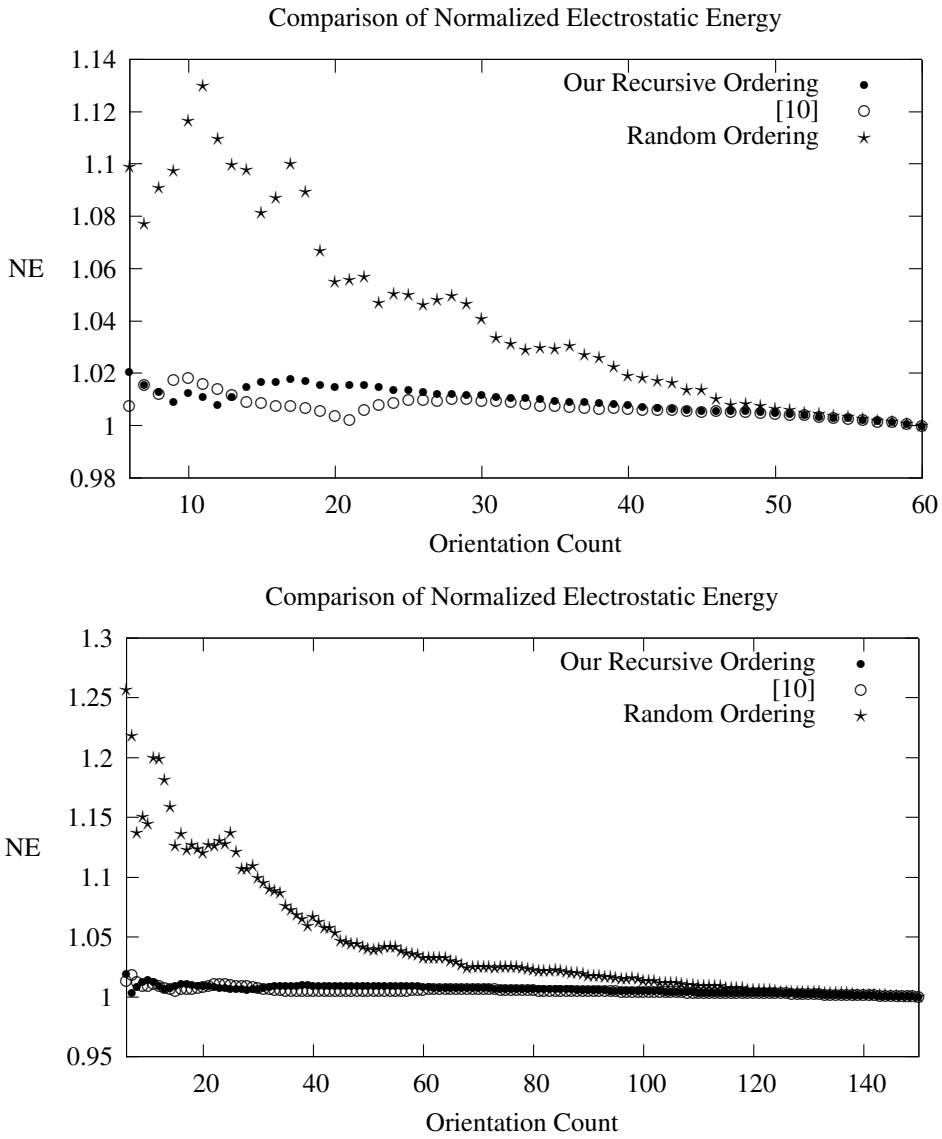


Figure 5: A comparison of the normalized energy of our recursive ordering algorithm, [10] and a randomized ordering.

always keep in mind that at some point, optimizations in certain parts of the acquisition sequence will directly decrease the optimality in other sections.

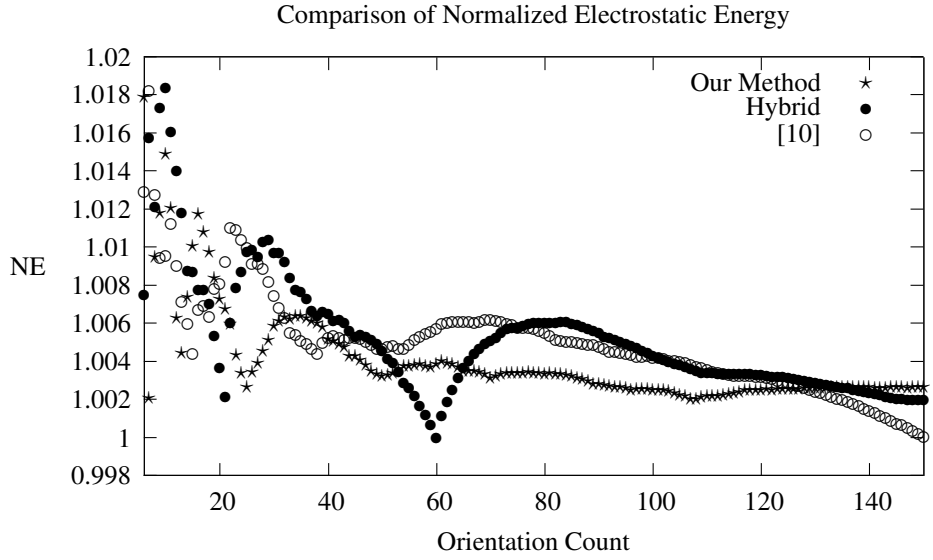


Figure 6: A comparison of the normalized energy of our recursive method, [10], and a hybrid between the two. The hybrid method uses the ordered point-set from [10] for $N = 1 \dots 60$ and uses our recursive method for $N \geq 61$.

5 Experimental Validation

In order to validate our method, we have tested it on real HARDI data and compared with the offline methods. Before we present the details of our results, we should clarify the exact algorithm that we have implemented as the Kalman filtering equations were derived in some generality.

5.1 Implementation Details

In any implementation of the Kalman filter, one must be careful in selecting proper initial conditions, most importantly so is the initial error covariance matrix $\mathbf{P}_0$ (written $\tilde{\mathbf{P}}_0$ in the previous section) . One usually selects $\mathbf{P}_0 = \sigma^2 I_{n\times n}$, and chooses an appropriate value for σ. In the case of ODF estimation, before the first measurement, we have no *a priori* information about the diffusion properties at any voxel. To convey our ignorance of the system's state to the filter, we must set σ to a sufficiently large value so that the Kalman filter does not place any weight on the initial state $\mathbf{x}_0$ which we set to the zero vector. In our experiments we typically use $\sigma = 1000$. We should note that this is in direct contradiction with [27, 29] who suggest to set $\sigma = 1$. However, they provide no justification for this selection and we have not been able to reproduce their results without using a large value for σ.

The only other parameter to be set is the noise covariance matrix $\mathbf{W}_k$. To ensure that we are in fact minimizing equation (2.9), we use $\mathbf{W}_k = I_{k\times k}$ which means that $R_k = 1$ in all the Kalman filtering equations. The Kalman filtering equations used in our implementation are

$$\begin{cases} \hat{\mathbf{x}}_0 = 0 \\ \mathbf{P}_0 = \left(1/\sigma^2 I_{n\times n} + \lambda \mathbf{L}\right)^{-1} \\ \mathbf{G}_k = \mathbf{P}_{k-1}\mathbf{C}_k^T \left(\mathbf{C}_k \mathbf{P}_{k-1} \mathbf{C}_k^T + 1\right)^{-1} \\ \mathbf{P}_k = \left(I - \mathbf{G}_k \mathbf{C}_k\right) \mathbf{P}_{k-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \mathbf{G}_k \left(y_k - \mathbf{C}_k \hat{\mathbf{x}}_{k-1}\right) \end{cases}$$

where $\sigma = 1000$. We should note here that in the case where $\mathbf{L}$ is invertible, we can take the limit as $\sigma \to \infty$ and initialize $\mathbf{P}_0 = (\lambda \mathbf{L})^{-1}$. However, for the case of ODF reconstruction, the Laplace-Beltrami operator has no dependence on the zeroth order spherical harmonic coefficient which makes $\mathbf{L}$ singular.

5.2 Results and Discussion

Two sets of MRI data have been used in these experiments. In all experiments, a rank 4 spherical harmonic basis was used so there are 15 coefficient to estimate. The comparison metric used is the mean squared error (MSE) between the spherical harmonic coefficients of the ODFs over all relevant portions of the brain. The relevant regions of the brain were deduced from the T2 image. The first database is from the CEA Neurospin Lab in Paris, France. It is the public HARDI database of [28]. The second is from the Max Planck Institute(MPI) in Leipzig, Germany [1]. We will refer to these by the acronyms CEA and MPI from now on. The CEA data was acquired on a 1.5T acquisitionner

with 200 encoding directions, $b = 3000s/mm^2$, 60 slices with 2mm thickness, 25 $b = 0s/mm^2$ images, 128x128 image matrix, TE=93.2 ms, TR = 19 s. The MPI data was acquired with a 3T acquisitionner using 60 encoding directions and the average of three measurements per direction, seven $b = 0$ images, a b-value of $1000s/mm^2$, 72 slices with 1.7 mm thickness, 128x128 image matrix, TE = 100 ms, TR = 12s.

First, the proposed algorithm has been tested and compared at each iteration with the optimal offline estimation. Figure 7 shows the evolution of the MSE for the MPI and the CEA datasets. We see that in both graphs the MSE monotonically decreases and converges to zero after the last diffusion gradient acquisition. From these graphs we can conclude approximately how many measurements are actually necessary. For the MPI dataset, there is little change between iteration 20 and 60, thus 20 measurements might be appropriate here. For the CEA dataset, the equivalent number is probably somewhere between 80 and 120. Keep in mind however, that these graphs are highly dependent on the fact that we have used a rank 4 spherical harmonic basis. Using a higher rank would result in more coefficients to be estimated and thus require more acquisitions. Furthermore, we cannot conclude with certainty how many measurements are required without investigating the effects on fiber tracking, fiber clustering and any other algorithms that depend on the ODF estimation.

The results in figure 7 are very convincing, but they do not actually prove that our method yields the smallest MSE at each iteration. We do, however, have a closed form for the optimal estimation after each acquisition in equation (3.2), so this is what we should be comparing our solution to. We have computed offline the optimal estimation after each diffusion gradient orientation as per equation (3.2) and compared our Kalman filtering method to this at each iteration. The results are shown in figure 8. This in fact validates our method, as the difference between our method and the optimal estimation is seen to be negligible at each iteration.

Next, we need to support our claim that the method from [27, 29] is sub-optimal. We have implemented their method and tested it with the same datasets. Figure 9 compares the MSE between our proposed method and the method of [27, 29] on both the MPI and CEA datasets. We can see that at the beginning of the acquisition, the MSE associated with the method from [27, 29] is an order of magnitude higher than our proposed method. Our method yields a lower MSE for *every* iteration in the acquisition until both methods converge to the optimal offline estimation at the last iteration.

To make a visual comparison between the two algorithms, we have generated ODF visualizations for a small region within a single axial slice of each dataset. The locations of the regions are shown in figures 10. Axial slice number 36 was chosen in both datasets and the ROIs were chosen in fiber crossing regions from the genu of the corpus callosum with peripheral fibers from the lateral cortex. In figures 11 and 12, we show the evolution of the ODFs under both algorithms next to the optimal estimation at that iteration for the MPI and CEA datasets respectively. We should note that the MPI HARDI data has an very high SNR, estimated to be roughly 37 (since 3 averaging) in the white matter, so the visual difference between the optimal estimation using 15 versus 60 acquisitions is very small. Thus, this is an excellent example showing visually that the Kalman filtering method from [27, 29] is sub-optimal. After 15 acquisitions, the optimal estimation is very good and close to the true ODFs, but the method of [27, 29] is clearly quite far from optimal and in fact has very little in common visually with the true ODF field. After 20 acquisitions, the two methods are visually much more similar, but numerically still much different, and we see that by 60 acquisitions, they both

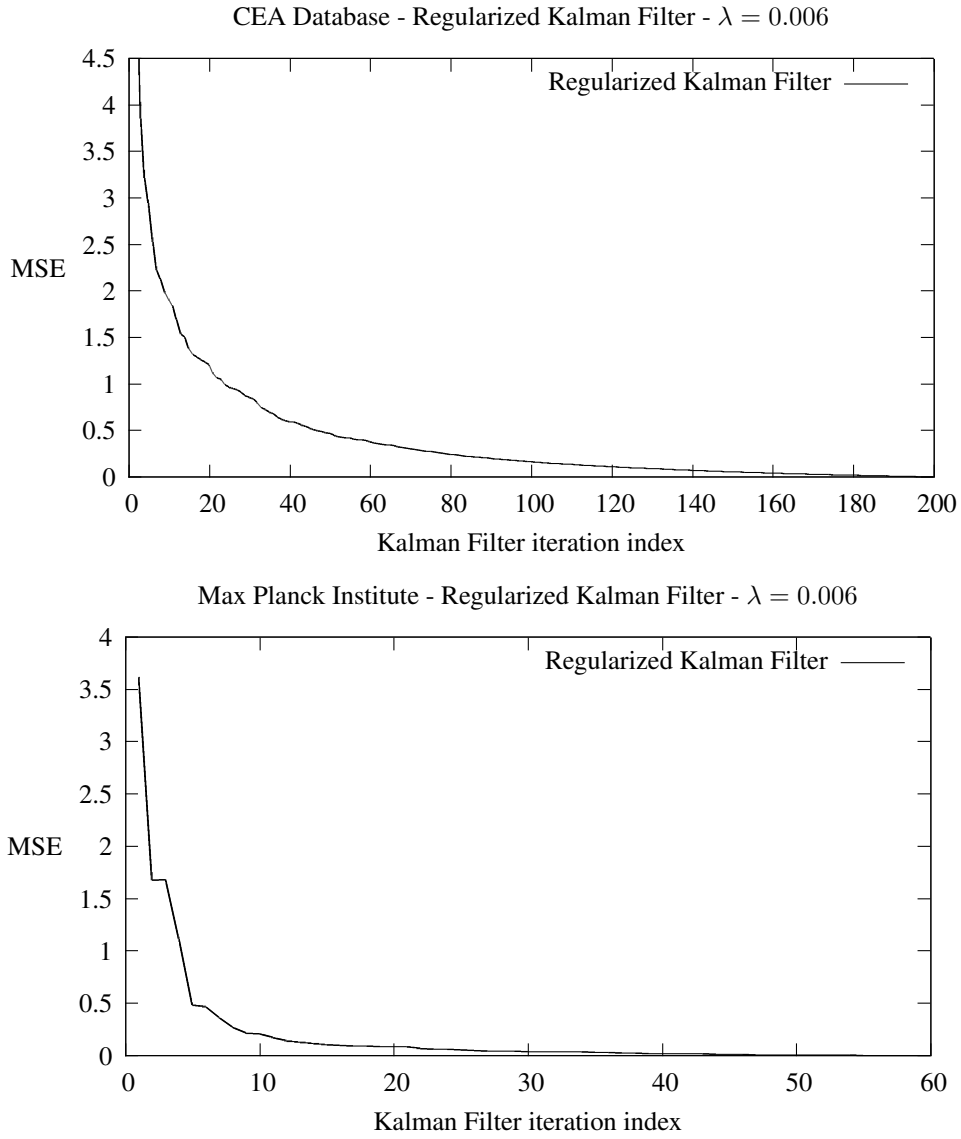


Figure 7: Mean squared error between the SH coefficients of the final offline estimation and each iteration's Kalman filter output using our proposed Regularized Kalman filtering method. The MSE is monotonically decreasing and converges to zero by the last acquisition

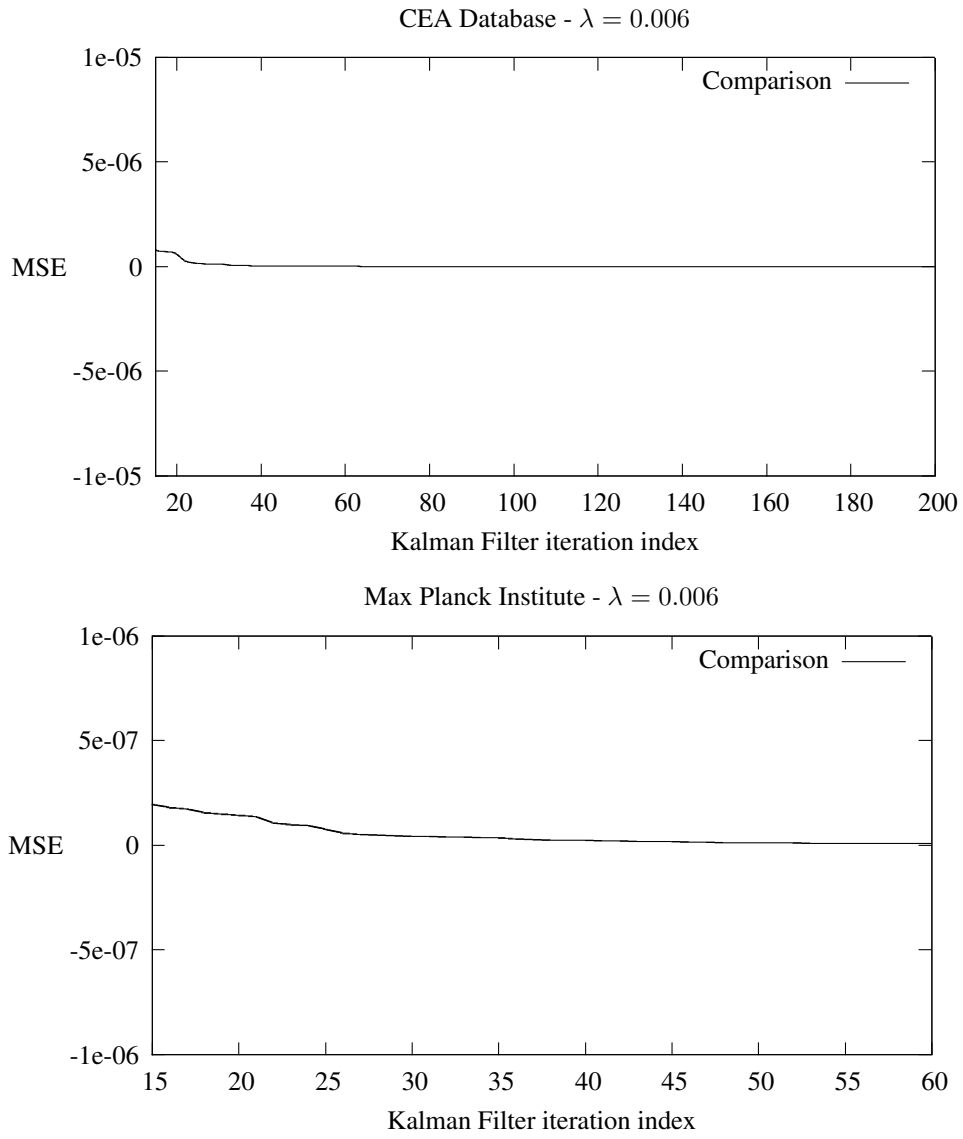


Figure 8: Mean squared error between the SH coefficients of the offline estimation and each iteration's Kalman filter output using our Regularized Kalman filtering algorithm. The offline estimation is computed for the subset of gradient acquisitions corresponding to the current Kalman filter iteration, thus it is the "best" we can do at each iteration. The MSE is approximately zero for the entire acquisition. This validates our algorithm by showing that it is equivalent to the offline estimation methods at each iteration.

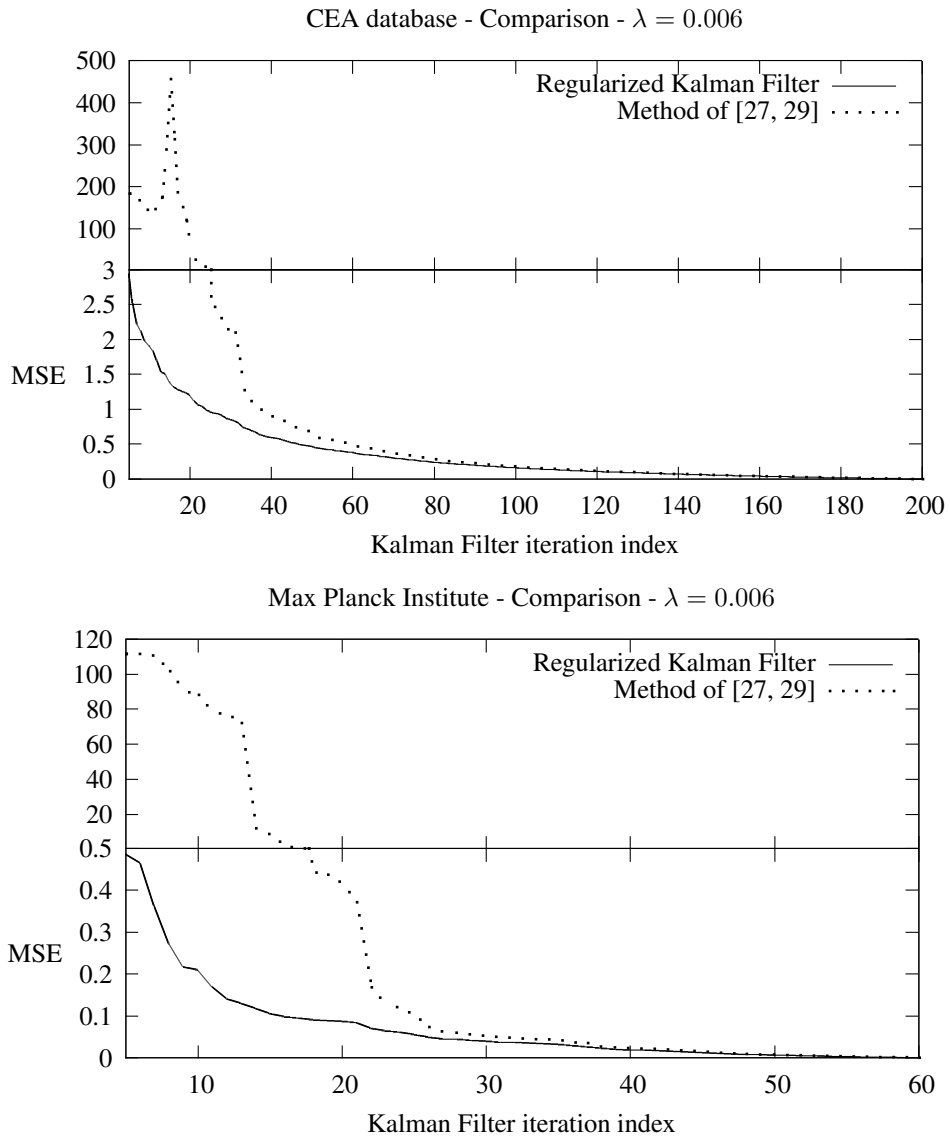
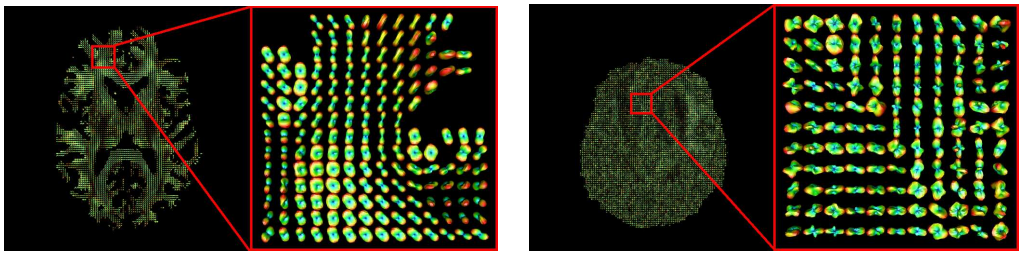


Figure 9: Mean squared error between the SH coefficients of the final offline estimation and each iteration's Kalman filter output. This graph compares our proposed Regularized Kalman filtering method to the method proposed in [27, 29]. We can see that our proposed algorithm is superior at every iteration with the largest differences coming in the first half of the acquisition. In order to show the important features, the tops and bottoms of the plots are on different scales



MPI data CEA data

Figure 10: The Kalman filtering methods are visually compared in figures 11 and 12 on small regions of the MPI and CEA data respectively, illustrating fiber crossing regions from the genu of the corpus callosum with peripheral fibers from the lateral cortex.

converge to the optimal solution. For the CEA data, the ODFs become visually indistinguishable after about 60 to 80 acquisitions.

To validate that the algorithm is indeed real-time, we measured the execution time for the CEA dataset. The total processing time to update the Kalman filter across the entire volume (ie: all slices) after a new acquisition was approximately 6.23 seconds for both our algorithm and the method from [27, 29]. As this is less than the repetition time of 12.5 seconds stated in [27, 29], both algorithms are truly real-time. The important difference between the algorithms is not the execution time, but that our proposed method solves the correct minimization problem at each iteration and hence provides a better estimation of the ODFs. For the test, both algorithms were implemented in C++ and executed on a 64-bit Linux machine with a dual-core 3.4GHz processor and 3.0 GBytes of RAM. However, the code was not written to take advantage of a multi-core processor and it was verified during the test that only one processor was being used. Theoretically, it would be possible to improve the performance by parallelizing the code, but this has not been investigated. We should note that we have defined real-time as the ability to produce the spherical harmonic coefficients of the ODF in real-time. In many applications, especially in a clinical setting, visualization is very important and one's definition of real-time depends on the applications at hand. Visualization is a very computationally intensive process and on a single workstation cannot be done in real-time. Further work could be aimed at implementing this algorithm on a cluster of workstations so visualization could also be performed in real-time.

Another topic for future work is the issue of the true Rician nature of the noise in the diffusion weighted images [5, 16, 31]. In this work we have modeled the noise as Gaussian because it is required by the linear Kalman filter. At high SNR, the Gaussian distribution is a good approximation to Rician noise, but the approximation is poor at low SNR which is typical of diffusion MRI. We can see these differences in the quality of the estimations between the MPI and CEA datasets. The MPI data is acquired by averaging three measurements per orientation. Firstly, this lowers the noise variance, and secondly, from the central limit theorem in probability theory we know that the average of identically distributed random variables will tend to a Gaussian distribution. Thus, the noise in the MPI dataset is well modeled by a Gaussian distribution whereas the noise in the CEA data is not.

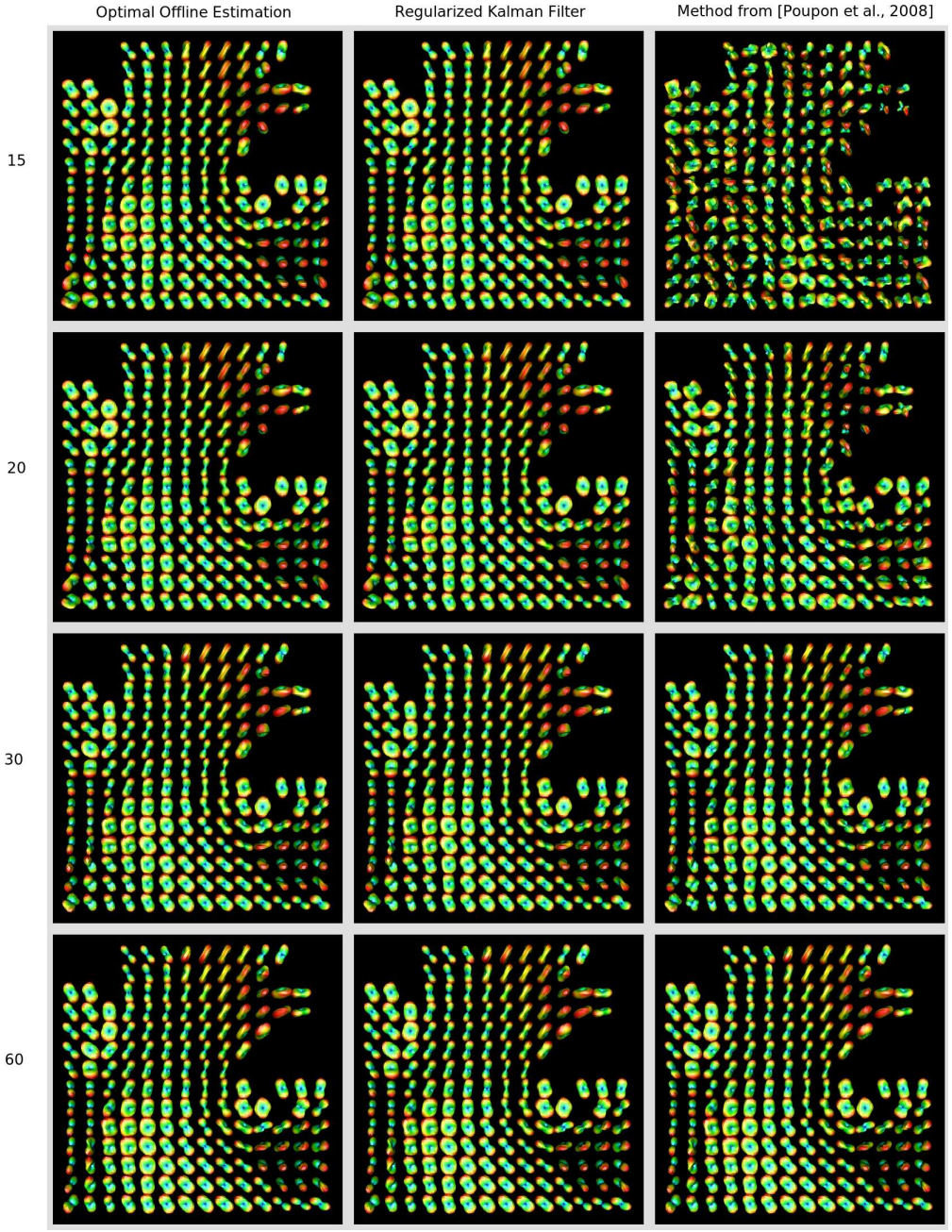


Figure 11: Visualization of the ODFs computed from the MPI data by (left) Optimal least squares solution (middle) our proposed Regularized Kalman filtering method, and (right) the method proposed by [27, 29]. The ODFs are shown after 15, 20, 30, and 60 diffusion gradient acquisitions (top to bottom).

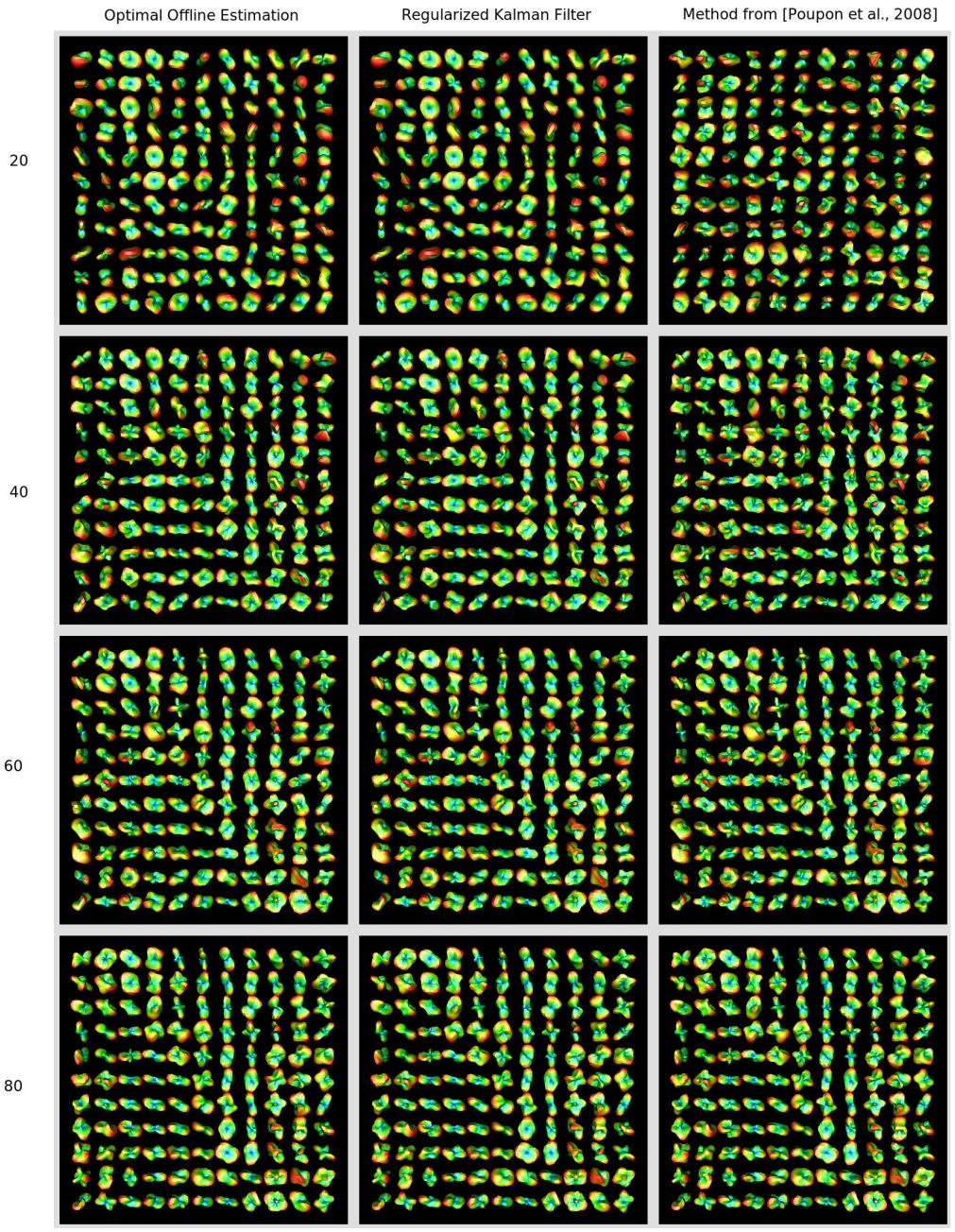


Figure 12: Visualization of the ODFs computed from the CEA data by (left) Optimal least squares solution (middle) our proposed Regularized Kalman filtering method, and (right) the method proposed by [27, 29]. The ODFs are shown after 20, 40, 60, and 80 diffusion gradient acquisitions (top to bottom).

The incorporation of the Rician noise model into a recursive filtering algorithm would likely yield better results for data with low SNR than the linear Kalman filter used in this work. However, this is a topic for future research.

6 Conclusion

We have developed a truly incremental Regularized Kalman filtering algorithm for real-time processing of diffusion Magnetic Resonance Q-Ball Imaging. This method allows for real-time estimation of the orientation distribution function (ODF) during an ongoing MRI scan. We have validated that our proposed method is equivalent to the standard offline processing techniques and is therefore fit to be used as a real-time processing algorithm in MRI acquisitions. We have also analyzed the only other proposed method known to the authors [27, 29] and have shown that it is *sub-optimal* and not truly an incremental solution. We have also proposed a fast algorithm for incrementally generating orientation sets that have uniformly distributed ordered subsets. Our results are encouraging and open some new exciting challenges and perspectives. For instance, the problem of motion and distortion correction is a challenge in practice and could potentially be tackled in real-time. These motion and distortion artefacts can be induced by the patient's motion during acquisition and from different DWIs depending on the applied gradient [30]. Other applications of real-time Kalman filtering for Q-ball imaging are numerous. It could be used to provide clinicians with a quality measure of the acquisition after each diffusion measurement and provide a stopping condition for the acquisition. This could shorten many HARDI acquisitions and reduce the probability of having to abort the acquisition due to motion of the patient. Aside from its obvious clinical applications, this real-time Kalman filtering framework will likely prove to be a useful tool for investigations into optimal diffusion gradient orientation sets, real-time fiber-tracking and connectivity mapping, real-time fiber orientation distribution function (FOD) estimation through spherical deconvolution [17, 35], and many other areas. We are currently working towards achieving these objectives.

Acknowledgments

This work was partly supported by the INRIA ARC Diffusion MRI Program, the INRIA Internship Program and the EADS Foundation. We express our sincere thanks to Cyril Poupon, Fabrice Poupon and colleagues for their help in providing us with the CEA Dataset. We are also grateful to Alfred Anwander and his colleagues for their help in providing us the MPI HARDI data.

Appendices

A Pseudo-inverse of $\mathbf{B}^+$

First we remind that pseudo-inverse exists and is unique for every matrix. For an invertible matrix, the pseudo-inverse coincides with the actual inverse. Furthermore, the pseudo-inverse is a reversible process and is its own inverse (ie: $\left(\mathbf{A}^\dagger\right)^\dagger = \mathbf{A}$).

We remind also the following results concerning the computation of pseudo-inverses.

Lemma 1. *Let A be an m-by-n matrix. If the columns of A are linearly independent then the pseudo-inverse of A is*

$$\mathbf{A}^\dagger = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$$

Proof. Since the columns of A are linearly independent, Rank$(\mathbf{A}) = n$. Now, $\mathbf{A}^T\mathbf{A}$ is an n-by-n matrix with Rank$(\mathbf{A}^T\mathbf{A}) =$ Rank$(\mathbf{A}) = n$ thus $\mathbf{A}^T\mathbf{A}$ is a square matrix with full rank and is therefore invertible. It is now easy to verify that the above formula satisfies the definition of the pseudo-inverse. □

Lemma 2. *Let A be an m-by-n matrix. If the rows of A are linearly independent then the pseudo-inverse of A is*

$$\mathbf{A}^\dagger = \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}$$

Proof. The proof is exactly the same as lemma 1. □

Now, we will derive a closed form formula for $\mathbf{A}_k = \left((\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T\right)^\dagger$. We first show that $(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T$ has full rank. For this, we will rely a bit on the physical process we are modelling. First, we need to take k large enough such that the system $\mathbf{y}_k = \mathbf{B}_k\mathbf{x}$ is an overdetermined linear system (ie: $k \geq n$). In the case of ODF estimation using a spherical harmonic basis of order 4 this means $k \geq 15$. In this case, it is easy to see that $\mathbf{B}_k$ has full rank for if it did not, then a column of $\mathbf{B}_k$ could be eliminated which would imply that one of the spherical harmonic basis functions is redundant, violating their orthogonality. It follows that $\mathbf{B}_k^T\mathbf{B}_k$ is non-singular and therefore positive definite. Now we will make some assumptions on $\mathbf{L}$. Given that it is the weighting matrix for a penalty of the form $\mathbf{x}^T\mathbf{L}\mathbf{x}$ on the state variable, we can without loss of generality assume that L is symmetric and positive semi-definite. Therefore, the sum $\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L}$ is positive definite and hence non-singular (invertible). And since $\mathbf{B}_k$ has full rank, the image of $\mathbf{B}_k^T$ has dimension n so the image of $(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T$ has dimension n and therefore has full rank.

Now, since $(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T$ has full rank, its rows are linearly independent, so we can use lemma 2. We have

$$\begin{aligned}
\mathbf{A}_k &= \left((\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T\right)^T \Bigg[\left((\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T\right) \\
&\qquad \left((\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1}\mathbf{B}_k^T\right)^T \Bigg]^{-1} \\
&= \mathbf{B}_N(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1} \Bigg[(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1} \left(\mathbf{B}_k^T\mathbf{B}_k\right) \\
&\qquad (\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1} \Bigg]^{-1} \\
&= \mathbf{B}_k(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L})^{-1}(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L}) \left(\mathbf{B}_k^T\mathbf{B}_k\right)^{-1} (\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L}) \\
&= \mathbf{B}_k \left(I_{n\times n} + \lambda \left(\mathbf{B}_k^T\mathbf{B}_k\right)^{-1}\mathbf{L}\right) \qquad \text{(A.1)}
\end{aligned}$$

B Analysis of Kalman Filtering from [27,29]

We will derive a formula here for $\left(\mathbf{D}_k^T\mathbf{D}_k\right)^{-1}\mathbf{D}_k^T$. First, we have

$$\begin{aligned}
\left(\mathbf{D}_k^T\mathbf{D}_k\right)^{-1} &= \left[\left(\mathbf{B}_k + \lambda\mathbf{B}_k\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\right)^T \left(\mathbf{B}_k + \lambda\mathbf{B}_k\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\right)\right]^{-1} \\
&= \Bigg[\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{L}\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{B}_k^T\mathbf{B}_k\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L} \\
&\quad + \lambda^2\mathbf{L}\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{B}_k^T\mathbf{B}_k\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\Bigg]^{-1} \\
&= \Bigg[\left(\mathbf{B}_N^T\mathbf{B}_N + \lambda\mathbf{L}\right)\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{B}_k^T\mathbf{B}_k \\
&\quad + \left(\mathbf{B}_N^T\mathbf{B}_N + \lambda\mathbf{L}\right)\lambda\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{B}_k^T\mathbf{B}_k\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\Bigg]^{-1} \\
&= \left[\left(\mathbf{B}_N^T\mathbf{B}_N + \lambda\mathbf{L}\right)\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\left(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{B}_k^T\mathbf{B}_k\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\right)\right]^{-1} \\
&= \left(\mathbf{B}_k^T\mathbf{B}_k + \lambda\mathbf{B}_k^T\mathbf{B}_k\left(\mathbf{B}_N^T\mathbf{B}_N\right)^{-1}\mathbf{L}\right)^{-1}\left(\mathbf{B}_N^T\mathbf{B}_N\right)\left(\mathbf{B}_N^T\mathbf{B}_N + \lambda\mathbf{L}\right)^{-1}
\end{aligned}$$

and

$$\begin{aligned}\mathbf{D}_k^T &= \left(\mathbf{B}_k + \lambda \mathbf{B}_k \left(\mathbf{B}_N^T \mathbf{B}_N\right)^{-1} \mathbf{L}\right)^T \\ &= \mathbf{B}_k^T + \lambda \mathbf{L} \left(\mathbf{B}_N^T \mathbf{B}_N\right)^{-1} \mathbf{B}_k^T \\ &= \left(\mathbf{B}_N^T \mathbf{B}_N + \lambda \mathbf{L}\right) \left(\mathbf{B}_N^T \mathbf{B}_N\right)^{-1} \mathbf{B}_k^T\end{aligned}$$

Combining the above two equations we have

$$\left(\mathbf{D}_k^T \mathbf{D}_k\right)^{-1} \mathbf{D}_k^T = \left(\mathbf{B}_k^T \mathbf{B}_k + \lambda \mathbf{B}_k^T \mathbf{B}_k \left(\mathbf{B}_N^T \mathbf{B}_N\right)^{-1} \mathbf{L}\right)^{-1} \mathbf{B}_k^T \tag{B.1}$$

C Matrix Inversion Formula

In the following lemma, we will prove a useful result used to derive a recursive formula for the Kalman gain matrix.

Lemma 3. *Let A, B, C be $n \times n$, $m \times m$, and $n \times m$ matrices respectively such that A, B, $\left(A^{-1} + CB^{-1}C^T\right)$, and $\left(B + C^T AC\right)$ are nonsingular. Then*

$$\left(A^{-1} + CB^{-1}C^T\right)^{-1} = A - AC\left(B + C^T AC\right)^{-1} C^T A$$

Proof. By direct computation, we have

$$\begin{aligned}& \left(A^{-1} + CB^{-1}C^T\right)\left(A - AC\left(B + C^T AC\right)^{-1} C^T A\right) \\ = \;& A^{-1}A - A^{-1}AC\left(B + C^T AC\right)^{-1} C^T A \\ & + CB^{-1}C^T A - CB^{-1}C^T AC\left(B + C^T AC\right)^{-1} C^T A \\ = \;& I_{n\times n} - CB^{-1}\left(B\left(B + C^T AC\right)^{-1} + C^T AC\left(B + C^T AC\right)^{-1}\right) C^T A \\ & + CB^{-1}C^T A \\ = \;& I_{n\times n} - CB^{-1}\left(B + C^T AC\right)\left(B + C^T AC\right)^{-1} C^T A + CB^{-1}C^T A \\ = \;& I_{n\times n} - CB^{-1}C^T A + CB^{-1}C^T A = I_{n\times n}\end{aligned}$$

□

D Formula for Kalman Filter Covariance Matrix

For completeness, we will reproduce here a result from the exercises in [8]. In particular, we will show that

$$E\left[(\mathbf{x} - \hat{\mathbf{x}}_k^0)(\mathbf{x} - \hat{\mathbf{x}}_k^0)^T\right] = \left(\mathbf{B}_k^T \mathbf{W}_k^{-1} \mathbf{B}_k\right)^{-1}$$

Recall that $\hat{\mathbf{x}}_k^0$ is the optimal state estimate for $\lambda = 0$ and that under the Gaussian noise assumption of the Kalman filter, we are modeling the observations as

$$\mathbf{y}_k = \mathbf{B}_k\mathbf{x} + \eta_k$$

where η_k is a Gaussian random vector with covariance matrix

$$E\left[\eta_k\eta_k^T\right] = \mathbf{W}_k$$

So, we have

$$\begin{aligned}
&E\left[(\mathbf{x}-\hat{\mathbf{x}}_k^0)(\mathbf{x}-\hat{\mathbf{x}}_k^0)^T\right]\\
=&E\Bigg[\left(\mathbf{x}-\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{y}_k\right)\\
&\qquad\left(\mathbf{x}-\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{y}_k\right)^T\Bigg]\\
=&E\Bigg[\left(\mathbf{x}-\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\left(\mathbf{B}_k\mathbf{x}+\eta_k\right)\right)\\
&\qquad\left(\mathbf{x}-\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\left(\mathbf{B}_k\mathbf{x}+\eta_k\right)\right)^T\Bigg]\\
=&E\left[\left(\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\eta_k\right)\left(\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\eta_k\right)^T\right]\\
=&\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}E\left[\eta_k\eta_k^T\right]\mathbf{W}_k^{-1}\mathbf{B}_k\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\\
=&\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{W}_k\mathbf{W}_k^{-1}\mathbf{B}_k\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\\
=&\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}\\
=&\left(\mathbf{B}_k^T\mathbf{W}_k^{-1}\mathbf{B}_k\right)^{-1}
\end{aligned}$$

E Formula for Kalman Filter Gain Matrix

For completeness, we include another result concerning the Kalman gain matrix. We will show that

$$\mathbf{G}_k = \mathbf{P}_k\mathbf{C}_k^T R_k^{-1} = \mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T + R_k\right)^{-1}$$

Using the result from appendix C we have

$$\begin{aligned}
&\mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T+R_k\right)^{-1}\\
=&\mathbf{P}_{k-1}\mathbf{C}_k^T\left(R_k^{-1}-R_k^{-1}\mathbf{C}_k\left(\mathbf{P}_{k-1}^{-1}+\mathbf{C}_kR_k^{-1}\mathbf{C}_k^T\right)^{-1}\mathbf{C}_k^TR_k^{-1}\right)\\
=&\left(\mathbf{P}_{k-1}-\mathbf{P}_{k-1}\mathbf{C}_k^TR_k^{-1}\mathbf{C}_k\left(\mathbf{P}_{k-1}^{-1}+\mathbf{C}_kR_k^{-1}\mathbf{C}_k^T\right)^{-1}\right)\mathbf{C}_k^TR_k^{-1}\\
=&\Bigg(\mathbf{P}_{k-1}-\mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T+R_k\right)^{-1}\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T+R_k\right)\\
&\quad R_k^{-1}\mathbf{C}_k\left(\mathbf{P}_{k-1}^{-1}+\mathbf{C}_kR_k^{-1}\mathbf{C}_k^T\right)^{-1}\Bigg)\mathbf{C}_k^TR_k^{-1}\\
=&\Bigg(\mathbf{P}_{k-1}-\mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T+R_k\right)^{-1}\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^TR_k^{-1}\mathbf{C}_k+\mathbf{C}_k\right)\\
&\quad\left(\mathbf{P}_{k-1}^{-1}+\mathbf{C}_kR_k^{-1}\mathbf{C}_k^T\right)^{-1}\Bigg)\mathbf{C}_k^TR_k^{-1}\\
=&\Bigg(\mathbf{P}_{k-1}-\mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T+R_k\right)^{-1}\mathbf{C}_k\mathbf{P}_{k-1}\left(\mathbf{C}_k^TR_k^{-1}\mathbf{C}_k+\mathbf{P}_{k-1}^{-1}\right)\\
&\quad\left(\mathbf{P}_{k-1}^{-1}+\mathbf{C}_kR_k^{-1}\mathbf{C}_k^T\right)^{-1}\Bigg)\mathbf{C}_k^TR_k^{-1}\\
=&\left(\mathbf{P}_{k-1}-\mathbf{P}_{k-1}\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T+R_k\right)^{-1}\mathbf{C}_k\mathbf{P}_{k-1}\right)\mathbf{C}_k^TR_k^{-1}\\
=&\mathbf{P}_k\mathbf{C}_k^TR_k^{-1}\\
=&\mathbf{G}_k
\end{aligned}$$

F Another Kalman Filtering Solution

The Kalman filtering solution we have presented in this work aims to modify the Kalman filter for a different minimization criterion. There is, however, another way of approaching this problem that involves some subtle manipulation of the minimization criterion, but provides some insight into the somewhat surprising result that only the initial condition is affected by the Laplace-Beltrami regularization. Recall the minimization criterion from equation (2.9)

$$M(\mathbf{x})=\left(\mathbf{y}_k-\mathbf{B}_k\mathbf{x}\right)^T\left(\mathbf{y}_k-\mathbf{B}_k\mathbf{x}\right)+\lambda\mathbf{x}^T\mathbf{L}\mathbf{x} \tag{F.1}$$

We can assume, as we have said before in this paper, that because $\lambda\mathbf{x}^T\mathbf{L}\mathbf{x}$ is a quadratic penalty on the state, that $\mathbf{L}$ is both positive semi-definite and symmetric. Thus, we can take the square root of

$\lambda\mathbf{L}$. Now, we can write the minimization above as

$$\begin{aligned} M(\mathbf{x}) &= (\mathbf{y}_k - \mathbf{B}_k\mathbf{x})^T (\mathbf{y}_k - \mathbf{B}_k\mathbf{x}) + \lambda\mathbf{x}^T\mathbf{L}\mathbf{x} \\ &= (\mathbf{y}_k - \mathbf{B}_k\mathbf{x})^T (\mathbf{y}_k - \mathbf{B}_k\mathbf{x}) + \left((\lambda\mathbf{L})^{1/2}\mathbf{x}\right)^T \left((\lambda\mathbf{L})^{1/2}\mathbf{x}\right) \\ &= \begin{bmatrix} \mathbf{0} - (\lambda\mathbf{L})^{1/2}\mathbf{x} \\ \mathbf{y}_k - \mathbf{B}_k\mathbf{x} \end{bmatrix}^T \begin{bmatrix} \mathbf{0} - (\lambda\mathbf{L})^{1/2}\mathbf{x} \\ \mathbf{y}_k - \mathbf{B}_k\mathbf{x} \end{bmatrix} \end{aligned}$$

Now, this is the minimization criterion for the least squares solution to the following linear system

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} (\lambda\mathbf{L})^{1/2} \\ \mathbf{B}_k \end{bmatrix} \mathbf{x} \tag{F.2}$$

By applying the Kalman filter to this linear system, we can recursively minimize equation (2.9). Essentially what this shows is that the Laplace-Beltrami regularization can be incorporated by running the Kalman filter on n "fake" measurements with a value of zero *before* the scan begins using observation matrices given by the rows of $(\lambda\mathbf{L})^{1/2}$. Once the Kalman filter has been iterated these n times, the state estimate will still be zero and, as we will now show, the covariance matrix $\mathbf{P}_n$ will be equal to the initial covariance matrix we derived for the regularized Kalman filter (3.11). This gives a more intuitive explanation of why our proposed method only requires the modification of the initial covariance matrix.

Now, we would like to show that running the Kalman filter on these n "fake" measurements before the true measurements will initialize the Kalman identically to our proposed regularized Kalman filtering algorithm. So, given that $\mathbf{P}_0 = \sigma^2 I_{n\times n}$ we would like to derive a formula for $\mathbf{P}_n$ after the first n "fake" measurements. From the Kalman filtering equations (2.5) we have the following recursive equation for $\mathbf{P}_k$

$$\mathbf{P}_k = \mathbf{P}_{k-1} - \mathbf{P}_{k-1}\mathbf{C}_k^T \left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T + 1\right)^{-1} \mathbf{C}_k\mathbf{P}_{k-1}$$

where $\mathbf{C}_k$ are now the rows of $(\lambda\mathbf{L})^{1/2}$ for $1 \le k \le n$ and for simplicity, we will assume the noise variances R_k are all unity. Furthermore, to simplify the calculations, we will take $\mathbf{L}$ to be diagonal as in the case of the Laplace-Beltrami regularization. Thus

$$(\lambda\mathbf{L})^{1/2} = \begin{bmatrix} (\lambda\ell_1)^{1/2} & & \\ & \ddots & \\ & & (\lambda\ell_n)^{1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1 \\ \vdots \\ \mathbf{C}_n \end{bmatrix}$$

Since $(\lambda\mathbf{L})^{1/2}$ is diagonal, $\mathbf{P}_k$ will remain diagonal for $1 \le k \le n$. Furthermore, $\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T + 1\right)^{-1}$ is a scalar, so the recursive equation for $\mathbf{P}_k$ can be re-written for $1 \le k \le n$ as

$$\mathbf{P}_k \quad = \quad \mathbf{P}_{k-1} - \mathbf{P}_{k-1}\mathbf{C}_k^T\mathbf{C}_k\mathbf{P}_{k-1} \left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T + 1\right)^{-1}$$

Now, since $\mathbf{P}_k$ is diagonal and $\mathbf{C}_k$ has one non-zero entry, is is not hard to see that $\mathbf{P}_{k-1}\mathbf{C}_k^T\mathbf{C}_k\mathbf{P}_{k-1}$ will be an $n \times n$ matrix with all zero entries except for the k^{th} diagonal entry, which will be given by $\lambda\ell_k\sigma^4$. Furthermore, we have $\left(\mathbf{C}_k\mathbf{P}_{k-1}\mathbf{C}_k^T + 1\right)^{-1} = \left(\lambda\ell_k\sigma^2 + 1\right)^{-1}$. Putting all this together, we see that each iteration will only affect *one* diagonal entry of $\mathbf{P}_k$ and that $\mathbf{P}_n$ will be a diagonal matrix where the k^{th} diagonal entry is $[\mathbf{P}_n]_k$

$$\begin{aligned}
[\mathbf{P}_n]_k &= \sigma^2 - \lambda\ell_k\sigma^4/(\lambda\ell_k\sigma^2 + 1) \\
&= (\lambda\ell_k\sigma^4 + \sigma^2 - \lambda\ell_k\sigma^4)/(\lambda\ell_k\sigma^2 + 1) \\
&= \sigma^2/(\lambda\ell_k\sigma^2 + 1) \\
&= (1/\sigma^2 + \lambda\ell_k)^{-1}
\end{aligned}$$

Thus we have that $\mathbf{P}_n = (1/\sigma^2 I_{n\times n} + \lambda\mathbf{L})^{-1}$ which is the exact same formula we have derived for $\mathbf{P}_0$ for the regularized Kalman filter in section 5.1. So the two methods provide different ways of looking at the same Kalman filtering problem. This method is useful in showing us why only the initial covariance matrix needs to be modified for our regularized Kalman filtering solution.

References

[1] A. Anwander, M. Tittgemeyer, D. Y. von Cramon, A. D. Friederici, and T. R. Knosche. Connectivity-based parcellation of broca's area. *Cerebral Cortex*, 17(4):816–825, 2007.

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

[3] P.J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66(1):259–267, 1994.

[4] P.J. Basser, J. Mattiello, R. Turner, and D. Le Bihan. Diffusion tensor echo-planar imaging of human brain. In *Proceedings of the SMRM*, page 584, 1993.

[5] Saurav Basu, Thomas Fletcher, and Ross Whitaker. Rician noise removal in diffusion tensor mri. In *Nineth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, Lecture Notes in Computer Science 4191, pages 117–125, Copenhagen, Denmark, October 2006.

[6] C. Beaulieu. The basis of anisotropic water diffusion in the nervous system - a technical review. *NMR in Biomedicine*, 15:435–455, 2002.

[7] T. E. J. Behrens, H. Johansen-Berg, S. Jbabdi, M. F. S. Rushworth, and M. W. Woolrich. Probabilistic diffusion tractography with multiple fibre orientations. what can we gain? *NeuroImage*, 34(1):144–155, 2007.

[8] C. K. Chui and G. Chen. *Kalman filtering with real-time applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[9] P. A. Cook, Y. Bai, S. Nedjati-Gilani, K. K. Seunarine, M. G. Hall, G. J. Parker, and D. C. Alexander. Open-source diffusion-mri reconstruction and processing. In *Proceedings of the 14th International Society of Magnetic Resonance inMedicine*, page 2759, Seattle, WA, USA, 2006. International Society for Magnetic Resonance in Medicine,url=http://www.cs.ucl.ac.uk/research/medic/camino/index.htm.

[10] Philip A Cook, Mark Symms, Philip A Boulby, and Daniel C Alexander. Optimal acquisition orders of diffusion-weighted mri measurements. *J Magn Reson Imaging*, 25(5):1051–1058, May 2007.

[11] M. Descoteaux, E. Angelino, S. Fitzgibbons, and R. Deriche. Apparent diffusion coefficients from high angular resolution diffusion imaging: Estimation and applications. *Magnetic Resonance in Medicine*, 56:395–410, 2006.

[12] Maxime Descoteaux. *High Angular Resolution Diffusion MRI: From Local Estimation to Segmentation and Tractography*. PhD thesis, Universite de Nice - Sophia Antipolis, 2008.

[13] Maxime Descoteaux, Elaine Angelino, Shaun Fitzgibbons, and Rachid Deriche. Regularized, fast and robust analytical q-ball imaging,. *Magnetic Resonance in Medicine*, 58:497–510, 2007.

[14] J. Dubois, C. Poupon, F. Lethimonnier, and D. Le Bihan. Optimized diffusion gradient orientation schemes for corrupted clinical dti data sets. *MAGMA*, 19(3):134–143, Aug 2006.

[15] T. Erber and G. M. Hockney. Complex systems: Equilibrium configurations of n equal charges on a sphere. *Advances in Chemical Physics*, 98:495–594, 1997.

[16] Pierre Fillard, Vincent Arsigny, Xavier Pennec, Kiralee M. Hayashi, Paul M. Thompson, and Nicholas Ayache. Measuring brain variability by extrapolating sparse tensor fields measured on sulcal lines. *NeuroImage*, 34:639–650, 2007.

[17] Bing Jian and Baba C. Vemuri. A unified computational framework for deconvolution to reconstruct multiple fibers from diffusion weighted mri. *IEEE Transactions on Medical Imaging*, 26(11):1464–1471, 2007.

[18] D. K. Jones, M. A. Horsfield, and A. Simmons. Optimal strategies for measuring diffusion in anisotropic systems by magnetic resonance imaging. *Magn Reson Med*, 42(3):515–525, Sep 1999.

[19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[20] D. LeBihan and E. Breton. Imagerie de diffusion in vivo par résonance magnétique nucléaire. *C. R. Acad. Sci. Paris*, 301 Série II:1109–1112, 1985.

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

[22] M.E. Moseley, Y. Cohen, J. Mintorovitch, J. Kucharczyk, J. Tsuruda, P. Weinstein, and D. Norman. Evidence of anisotropic self-diffusion. *Radiology*, 176:439–445, 1990.

[23] P.A. Osment, K.J. Packer, M.J. Taylor, J. J. Attard, T. A. Carpenter, L. D. Hall, S. J. Doran, and N. J. Herrod. Nmr imaging of fluids in porous solids. *Phil. Trans. Roy. Soc.*, 333:441–452, 1990.

[24] N. G. Papadakis, C. D. Murrills, L. D. Hall, C. L. Huang, and T. Adrian Carpenter. Minimal gradient encoding for robust estimation of diffusion anisotropy. *Magn Reson Imaging*, 18(6):671–679, Jul 2000.

[25] C. Pierpaoli, P. Jezzard, P. Basser, A. Barnett, and G. D. Chiro. Diffusion Tensor MR imaging of human brain. *Radiology*, 201:637–648, 1996.

[26] C. Poupon. *Détection des faisceaux de fibres de la substance blanche pour l'étude de la connectivité anatomique cérébrale.* PhD thesis, Ecole Nationale Supérieure des Télécommunications, December 1999.

[27] C. Poupon, F. Poupon, A. Roche, Y. Cointepas, J. Dubois, and J.-F. Mangin. Real-time mr diffusion tensor and q-ball imaging using kalman filtering. In *10th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'07)*, pages 27–35, Brisbane, Australia, November 2007.

[28] Cyril Poupon, Fabrice Poupon, Lionel Allirol, and Jean-Francois Mangin. A database dedicated to anatomo-functional study of human brain connectivity. In *Twelfth Annual Meeting of the Organization for Human Brain Mapping (HBM)*, 2006.

[29] Cyril Poupon, Alexis Roche, Jessica Dubois, Jean-Franois Mangin, and Fabrice Poupon. Real-time mr diffusion tensor and q-ball imaging using kalman filtering. *Medical Image Analysis, Volume 12, Issue 5, Pages 527-534*, Jun 2008.

[30] G.K. Rohde, A.S. Barnett, P.J. Basser, S. Marenco, and C. Pierpaoli. Comprehensive approach for correction of motion and distortion in diffusion-weighted mri. *Magn Reson Med.*, 51(1):103–114, 2004.

[31] J. Sijbers, A. J. den Dekker, J. Van Audekerke, M. Verhoye, and D. Van Dyck. Estimation of the noise in magnitude mr images. *Magnetic Resonance Imaging*, 16(1):87–90, 1998.

[32] D.G. Taylor and M.C. Bushell. The spatial mapping of translational diffusion coefficients by the nmr imaging technique. *Physics in Medicine and Biology*, 30(4):345–349, 1985.

[33] J. J. Thomson. On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. *Philosophical Magazine*, 7:237–265, 1904.

[34] J.-D. Tournier, F. Calamante, D.G. Gadian, and A. Connelly. Direct estimation of the fiber orientation density function from diffusion-weighted mri data using spherical deconvolution. *NeuroImage*, 23:1176–1185, 2004.

[35] J-Donald Tournier, Fernando Calamante, and Alan Connelly. Robust determination of the fibre orientation distribution in diffusion mri: Non-negativity constrained super-resolved spherical deconvolution. *NeuroImage*, 35(4):1459–1472, 2007.

[36] D. Tuch. Q-ball imaging. *Magnetic Resonance in Medicine*, 52(6):1358–1372, 2004.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399