Paper E

Shape Adaptive DCT for Denoising of Tensor Valued Images *

* Preprint.

Shape Adaptive DCT for Tensor Valued Imaging

Ørjan Bergmann, Oddvar Christiansen, Johan Lie and Arvid Lundervold

Version January 3, 2007

***Abstract*— During the last ten years or so, diffusion tensor imaging has been used in both research and clinical medical applications. In order to construct the diffusion tensor images, a large set of direction sensitive MRI acquisitions are required. These acquisitions in general have a lower signal-to-noise ratio than conventional MRI acquisitions. In this paper we discuss computationally effective algorithms for noise removal for DTI using the framework of shape-adaptive discrete cosine transform.**

I. Introduction

Diffusion Tensor Magnetic Resonance Imaging (DTI) is an important magnetic resonance imaging protocol used in both research and in clinical applications. The DTI modality has the advantage that highly structured tissue, for example the nerve fibers in the human brain can be studied non invasively [15], [25]. From a series (typically 6-50) of direction sensitive MR acquisitions a 3×3 diffusion tensor can be estimated for each voxel of the imaging domain. From these voxel-wise diffusion tensors a number of interesting clinical quantities can be estimated and used to investigate or differentiate between normal and abnormal tissue, e.g. in Multiple Sclerosis or Schizophrenia research [7], [8], [13].

However, it is a well-known fact that the MRI signal from the scanner contains measurement noise which degrades the images. The signal to noise ratio (SNR) in DTI is low compared to standard MRI. This makes it important to construct good models and methods for noise removal of diffusion tensor data. Due to the huge amount of data, the methods should be efficient with regards to computational time. In the following, we model the MRI signal as a composition of clean signal and additive normally distributed noise

$$S^{\text{noisy}} = S^{\text{clean}} + \eta(0, \sigma). \tag{1}$$

Although we cannot in general assume that the noise is normally distributed with zero mean, we may approximate the noise by such a distribution.

Several successful methods for denoising of diffusion tensor MRI have been proposed [2], [3], [20]–[22], [24]. The existing state-of-the art methods are based on partial differential equations. The nature of these methods typically makes them computationally heavy. In this paper we introduce an alternative and effective method for regularization of matrix valued images, based on *shape-adaptive discrete cosine transform* (SA-DCT) methods [1], [6], [10].

II. Background

A. Diffusion Tensor Imaging

Tensor valued data occurs in many branches of science, see e.g. [23]. In this paper the tensor valued data comes from diffusion tensor MRI of the human brain. From a set of K direction sensitive magnetic resonance images $\{S_k\}_{k=1}^K$ a symmetric positive definite tensor $D \in \mathbb{R}^{3\times 3}$ is constructed in each voxel of the image domain. This matrix yields structural information of the tissue in each voxel.

The relationship between each direction weighted measurement and the diffusion tensor D is given by the Stejskal-Tanner equation [19] [18]

$$S_k = S_0 e^{-b g_k^T D g_k}, k = 1, 2, \cdots, K \tag{2}$$

where b is a positive scalar, and $g_k \in \mathbb{R}^3$ is the direction in which measurement S_k is acquired. From K direction weighted measurements this gives us K equations that we use for estimating the six unknowns of the diffusion tensor D. This can be done for example by a linear least-squares method, or other more adaptive methods [16]. We note that since the transformation (2) is nonlinear, we do not know the distribution of the noise in each element of the tensor D. Hahn et.al has studied how noise propagates through the estimation process [9].

In structured tissue such as white matter, the self-diffusion of water is highly anisotropic. In gray matter and in serbrospinal fluid the self-diffusion of water is more or less isotropic. Therefore, based on knowledge of the diffusion tensor D, a model of the paths in the white matter can be constructed via fiber tracking [15] [14] [14] [25].

The quality of the estimated diffusion tensor depends on many parameters. One particular parameter is the number of direction sensitive acquisitions. A high number of direction sensitive acquisitions gives a tensor estimate of good quality. On the other hand, a small number of acquisitions gives a low scanner time for each patient. In this paper we investigate the possibility of post processing the data from a low number of acquisitions, and still construct tensor estimates of high quality. Thus, a partial goal is to decrease the scanner time for each patient.

B. Shape-Adaptive Discrete Cosine Transform

The 2D discrete cosine transform (DCT) is extensively used in image science. In its original formulation it transforms a quadratic region in the spatial domain to the frequency domain. Being a harmonic transform the DCT has a compactification property, i.e. fairly good approximations of the signal can be achieved by employing only a few of the coefficients in the frequency domain. However, when the domain of the transform contains sharp edges and only a few of the coefficients in the frequency domain are employed for the reconstruction to the spatial domain, various artifacts such as smearing of edges and Gibbs phenomena occur. To avoid these

artifacts, it has been shown by several authors that the region should be as homogeneous as possible. This is achieved by replacing the static regions from the standard DCT by regions which adapts to the information in the image. We choose these regions in such a way that the data can be well approximated by a smooth, slowly varying function. Such a function is well approximated by few coefficients from the frequency domain. The regions should ideally not contain any discontinuities.

In a series of papers, Katkovnik, Foi, Egiazarian, Astola and others describe shape adaptive DCT (SA-DCT) for denoising of 2D grayscale and color images [1], [5], [10], [11]. The algorithm can be divided into three different stages, namely *construction of an adaptive neighborhood* for each point in the domain, *transformation and thresholding* of each neighborhood, and finally *estimation* of the noise-free image. The adaptive neighborhoods are constructed by local polynomial approximations (LPA) in combination with the intersection of confidence intervals (ICI) rule []. The transformation of each neighborhood to the frequency domain is done by the Sikora DCT algorithm, and hard thresholding is applied on the coefficients in the frequency domain. The inverse Sikora DCT algorithm is then applied resulting in a denoised region. Since there is a region around every pixel, and in general these regions overlap, we have an over complete basis. This over complete basis is used to construct a final image by weighting the basis elements in a proper way.

The state-of-the-art results obtained by the SA-DCT methods in 2D as well as their efficiency makes them attractive for denoising of 3D scalar valued images as well as 3D matrix valued images. In this paper we extend the framework of SA-DCT to both 3D scalar valued and 3D matrix valued images.

C. Overview

In section III-A we will describe our 3D extension of the SA-DCT algorithm. In section III-B we describe how the 3D algorithm can be applied to our 3D matrix valued DTI data.

III. Methods

A. The Shape-adaptive discrete cosine transform

Let $I : \mathbb{R}^N \to \mathbb{R}$ denote the noisy dataset which is discretized on a uniform grid. In the rest of this paper we restrict the attention to 2-dimensional and 3-dimensional datasets, so $N \in \{2, 3\}$. We refer to the resulting denoised dataset as I^-. Presently, we treat the standard deviation σ of the noise of I as a parameter. However, we will later show how to obtain a good estimate of σ.

A main ingredient of the SA-DCT method is the adaptive neighborhood Ω_x surrounding each voxel $x \in \mathbb{R}^N$. The idea is that this neighborhood should contain voxels that in some way are "similar", i.e. homogeneous. A neighbouring point $y \in \mathbb{R}^N$ can either have an intensity $I(y)$ which is close to the intensity $I(x)$, or the intensities can differ substantially. In the case where $I(y) \approx I(x)$, we want to include the point y in the adaptive neighborhood of x. To decide which voxels that should belong to the adaptive neighborhood of a given point, we use local polynomial approximations (LPA) and the intersection of confidence intervals (ICI) rule [11].

$g^{(1)} = [1]$
$g^{(2)} = [0.6500000, 0.3500000]$
$g^{(3)} = [0.4083333, 0.3333333, 0.2583333]$
$g^{(5)} = [0.2400, 0.2200, 0.2000, 0.1800, 0.1600]$
$g^{(7)} = [0.15250, 0.14928, 0.14607, 0.14285, 0.13964, 0.13642, 0.13321]$
$g^{(9)} = [0.111, 0.111, 0.111, 0.111, 0.111, 0.111, 0.111, 0.111, 0.111]$

TABLE I
THE SIX LPA KERNELS $g^{(h)}, h \in \mathcal{H} = \{1, 2, 3, 5, 7, 9\}$ USED IN THE PAPER.

To construct the adaptive neighborhood we consider a set of directions $\theta_i \in \mathbb{R}^N$ such that each component of θ_i is either -1, 0 or 1, but never all equal to zero. It follows that there must be $3^N - 1$ such directions in an N-dimensional dataset. In 2D [10] there are eight such directions; the four cardinal and the four intermediate compass directions. In 3D there are 26 unique directions, following a similar pattern.

We span a star shaped skeleton Ω_x^* around each point x in the image domain by tracing the voxels along straight lines in the directions of θ_i. The length d_i corresponding to the straight line in the direction of θ_i in the skeleton is determined by the ICI algorithm. We close the skeleton such that it becomes a polygonal hull by joining neighboring endpoints of the vertices in the skeleton by line segments (in 2D) or triangles (in 3D). We denote the domain inside this closed polygonal hull by Ω_x. For each voxel in the image domain such an adaptive neighborhood is constructed. In the following section we explain how we can use the LPA-ICI method to compute the length d_i of each branch in the star.

1) LPA-ICI: To span the region Ω_x^* we calculate the support of each branch in the star, i.e. how many voxels that should be included along each direction vector θ_i, $i = 1, 2, \cdots, 3^N - 1$. The idea is that the voxels in Ω_x^* should have intensity values which are close to the intensity value of the voxel x. Variations in the included data should be due to the noise level and small local variations, and not due to edges in the image.

To achieve this we filter each direction with LPA kernels $\{g^{(h)}\}_{h \in \mathcal{H}}$ of varying length h. The specific choice of $\mathcal{H}$ is discussed in the section on numerical results, and shown in figure III-A.1. For each kernel $g^{(h)}$ containing weights $g_i^{(h)}$, where $i \in 1, 2, \cdots, h$, we have the property that the center voxel x has the highest weight $g_1^{(h)}$. In addition the weights sum to 1 and decrease with the length of the filter.

We can look at this filtering as a convolution of the data with a filter kernel of varying length. When the kernel $g^{(h)}$ is applied to the voxels in direction θ_i we get

$$\mu^{(h)} = \sum_{j=1}^{h} g_j^{(h)} I(x + (j-1)\theta_i). \tag{3}$$

Notice that for each direction θ_i we get $|\mathcal{H}|$ filtered values for the center pixel x.

The standard deviation of the noise in $\mu_i^{(h)}$ is given trough the relation

$$\sigma_{\mu^{(h)}} = \sigma \|g^{(h)}\| \tag{4}$$

Algorithm 1 sadct(I, σ, Γ)

Require: I is an N-dimensional image, σ is the standard deviation of the noise of I

for all $x \in I$ **do**
  Set $\Omega_x \leftarrow \text{lpaici}_x(I, \sigma, \Gamma)$
  Set $\hat{\Omega}_x \leftarrow \text{dct}(\Omega_x)$
  Define cutoff frequency f as in eq. (10)
  Apply f as specified in eq. (??) giving $\hat{\Omega}_x^-$
  Set $\Omega_x^- \leftarrow \text{dct}^{-1}(\hat{\Omega}_x^-)$
  Define weight $k \leftarrow (\|\hat{\Omega}_x^-\| + 1)\|\Omega_x\|$
  for all $y \in \Omega_x^-$ **do**
    Set $I^-(y) \leftarrow I^-(y) + k\Omega_x^-(y)$
    Set $\text{K}(y) \leftarrow \text{K}(y) + k$
  end for
end for
for all $x \in I^-$ **do**
  $I^-(x) \leftarrow I^-(x)$ / $\text{K}(x)$
end for

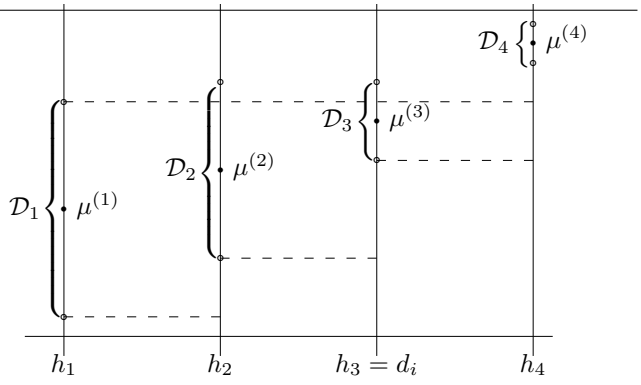

Fig. 1. An example of the LPA-ICI algorithm when $\mathcal{H} = \{h_j\}_{j=1}^4$. The area between the dashed lines shows the intersection of all previous confidence intervals. When the intersection is empty (here at h_4) the algorithm is terminated.

For each direction we then get the confidence intervals

$$\mathcal{D}_h = [\mu^{(h)} - \Gamma\sigma_{\mu^{(h)}}, \mu^{(h)} + \Gamma\sigma_{\mu^{(h)}}], \tag{5}$$

where $\Gamma > 0$ is a global parameter of the algorithm. A large Γ results in a large noise tolerance, and more voxels will be included in the regions, and vice versa.

The ICI rule states that along the direction vector θ_i we should choose the largest distance $d_i \in \mathcal{H}$ where we have intersection of all the confidence intervals, see figure III-A.1. More precisely

$$d_i = \max_{h \in \mathcal{H}} \{h : (\mathcal{D}_1 \cap \mathcal{D}_2 \cap \cdots \cap \mathcal{D}_h) \neq \emptyset\} \tag{6}$$

Having determined the length of each branch in the star shaped domain Ω_x^*, we define the neighborhood Ω_x as all voxels inside the polygonal hull closing Ω_x^* with branches $d_i\theta_i$, where $i \in 1, 2, \cdots, 3^N - 1$. By construction, the intensities in this region should not contain large changes due to edges in the image. The noise in this region can now easily be removed by thresholding small coefficients in the frequency domain. We use the discrete cosine transform for this purpose in section III-A.2.

Note that since we only perform a LPA-ICI estimation on the voxels that coincide with the skeletonized domain Ω_x^*, we do not have direct control over the intensity values in the set $\Omega_x \backslash \Omega_x^*$. It has been shown that for scalar images this approach is a good compromise between efficiency and accuracy. See for example [Foi].

2) The DCT algorithm of Sikora: The discrete cosine transform (DCT) is used extensively in image science. The one dimensional DCT is defined as

$$\hat{z_k} = c_k \sum_{n=0}^{N-1} z_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right], \tag{7}$$

where $c_0 = \sqrt{\frac{1}{N}}$ and $c_k = \sqrt{\frac{2}{N}} \quad k > 0$. The coefficients c_k ensures that the transform is orthogonal. Note that this transform can be expressed as a matrix-vector product

$$\hat{z} = Az \tag{8}$$

The matrix A is in general orthogonal, which implies that the inverse DCT can be expressed as

$$z = A^T\hat{z} \tag{9}$$

Two and three dimensional DCT are usually achieved by successively applying the one dimensional DCT along the coordinate axes. However, note that Ω_x will in general not be rectangular. Sikora has developed an algorithm for discrete cosine transform on non-rectangular domains [17]. In this paper we employ this algorithm. In the following we let Ω_x^0 denote the quadratic (in 2D) or cubic (in 3D) zero-extension of Ω_x. Note that when examining Ω_x along the coordinate axes, it may be non-contiguous. This causes problems when applying the traditional DCT algorithm as many components of the DCT domain will be needed to represent these high jumps in intensities introduced by the zero-padding. Sikoras approach avoids this problem by first shifting all values of Ω_x^0 corresponding to values in Ω_x along the first coordinate axis so that they become consecutive in Ω_x^0. A one dimensional DCT, of varying length, is then applied to all the shifted data. The same procedure is then applied to each dimension in turn, by first shifting the data and then applying the 1D DCT. When applying the inverse DCT we need to invert these shifts, so a record of the rearrangements must be maintained.

In Figure III-A.2 we display a 2D example of how the non-zero padded pixels are shifted to produce consecutive values, first in the x-direction and then in the y-direction.

3) Thresholding in the DCT domain: Let $\hat{\Omega}_x$ denote the domain transformed from Ω_x using the DCT algorithm described in the previous section and let $\hat{z}$ denote a given coefficient in $\hat{\Omega}_x$. In addition let $|\hat{\Omega}_x|$ denote the number of coefficients in the neighborhood $\hat{\Omega}_x$. The cutoff threshold f is given as

$$f = \sigma\sqrt{2\log(|\hat{\Omega}_x|) + 1}, \tag{10}$$

and the hard thresholded coefficients $\hat{z}^-$ is given as

$$\hat{z}^- = \begin{cases} \hat{z} & \text{if } \hat{z} \geq f \\ 0 & \text{if } \hat{z} < f \end{cases},$$

for all $\hat{z} \in \hat{\Omega}_x$.

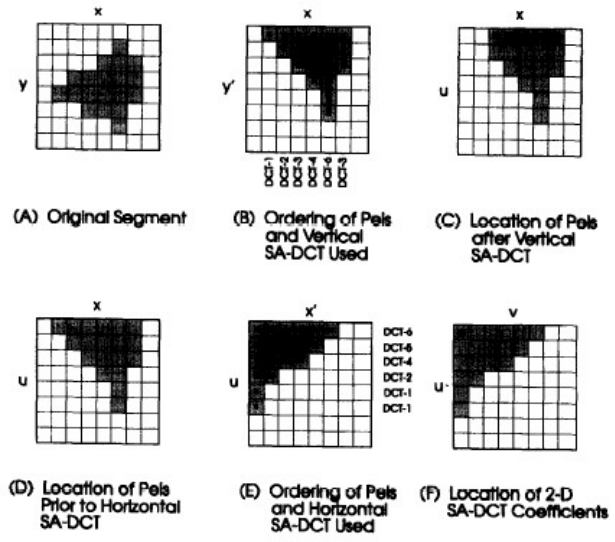

Fig. 2. An overview of how the nonzero pixels are collected in one corner by the algorithm of Sikora.

The thresholded region is then transformed and shifted back into the spatial domain by the inverse (Sikora) DCT (as outlined in section III-A.2) giving Ω_x^- in the spatial domain.

4) Estimation from over-complete basis: Notice that since we calculate a region Ω_x for every voxel in the image we have extensive region-overlap, i.e. we have an over complete basis. To reconstruct an image from this information, we have to weight the data in a proper way. We assign a weight to every region and use the information in overlapping regions to estimate the denoised image. It is a standard approach to use weights that are inversely proportional to the mean variance of the region. However, for adaptive regions this leads to over smoothing. To compensate for this we can divide the weights by the square of the size of the region.

The mean variance of the region Ω_x^- is given by

$$\sigma^2_{\Omega_x^-} = \sigma^2 \frac{1 + |\hat{\Omega}_x^-|}{|\hat{\Omega}_x|}, \tag{11}$$

where $|\hat{\Omega}_x^-|$ is the number of nonzero coefficients in $\hat{\Omega}_x^-$. This gives the following weights for the regions

$$w_x = \frac{1}{(1 + |\hat{\Omega}_x^-|)|\hat{\Omega}_x|}. \tag{12}$$

The regions can now be weighted together giving the final recovered estimate I^- using the relation

$$I^-(p) = \frac{\sum_x w_x \Omega_x^-(p)}{\sum_x w_x}, \tag{13}$$

for all $p \in I$ and where the sum is taken over all voxels x so that Ω_x^- contains p.

B. SA-DCT for denoising of DTI

It is not completely clear whether the raw data $\{S_k\}_{k=0}^K$ or the estimated tensor $\{D_{ij}\}_{i,j=1}^3$ should be processed in order to remove noise from diffusion tensor data. While it can be argued that the noise should be removed as close to its source as possible, it is possible to preserve implicit properties of the tensor if we work in the tensor domain. In the current version of this paper, we have chosen to process the data in the tensor domain. More research must be done in order to check which approach that is best for real DTI data. In another paper, two of the authors have investigated total variation regularization of tensor valued data [3]. In that paper, the estimated tensor is regularized. In order to ensure positive definiteness of the regularized tensor, it is represented implicitly as $D = LL^T$ where L is a lower triangular matrix. In this paper we adapt this approach when regularizing tensor valued data by SADCT methods.

IV. Preliminary Results

In this section we show preliminary qualitative numerical results achieved by the method proposed in this paper. A quantitative analysis of the results will be done in a maturized version of the paper, where we will also compare our approach with other approaches, like [3]. We process both synthetically produced images and real diffusion tensor images of a healthy human volunteer.

Example 1: *3D scalar valued data*

We have in this paper generalized the SADCT methods from 2D to 3D images. In the first example we want to show the difference between application of the 2D SADCT algorithm slice by slice (quazi-3D) and application of the full 3D algorithm. We use data from the "brainweb" database, a database of semi-realistic simulated MR images [4].

As expected, there is a difference between the two approaches. When we apply the quazi-3D algorithm along slices in the Z-dimension of the image, it is clear that the noise reduction is only be performed in each slice in the XY plane, and not across different slices. When we apply the full 3D algorithm, the noise reduction is performed in the same manner along all three dimensions. The difference between the two approaches can be seen in Figure 3. In Figure 3 we show a slice of the image which is orthogonal to the XY-plane along which the quazi-3D algorithm is applied. As we can observe from Figure 3 the result from the full 3D algorithm proposed in this paper is better than the result from the quazi-3D algorithm. In Figure 4 we show a zoom-in of a small portion of the result from Figure 3.

Example 2: *3D tensor valued synthetic data*

The main motivation behind this paper is regularization of tensor valued images, in particular diffusion tensor images. First we regularize a synthetic DTI dataset where the object that is simulated is a torus. The DTI torus has been generated by the software *teem*, written by Gordon Kindlmann [12]. We use the software DTI-studio developed by Mori and coworkers for the visualization of the color coded FA images [**?**]. We have

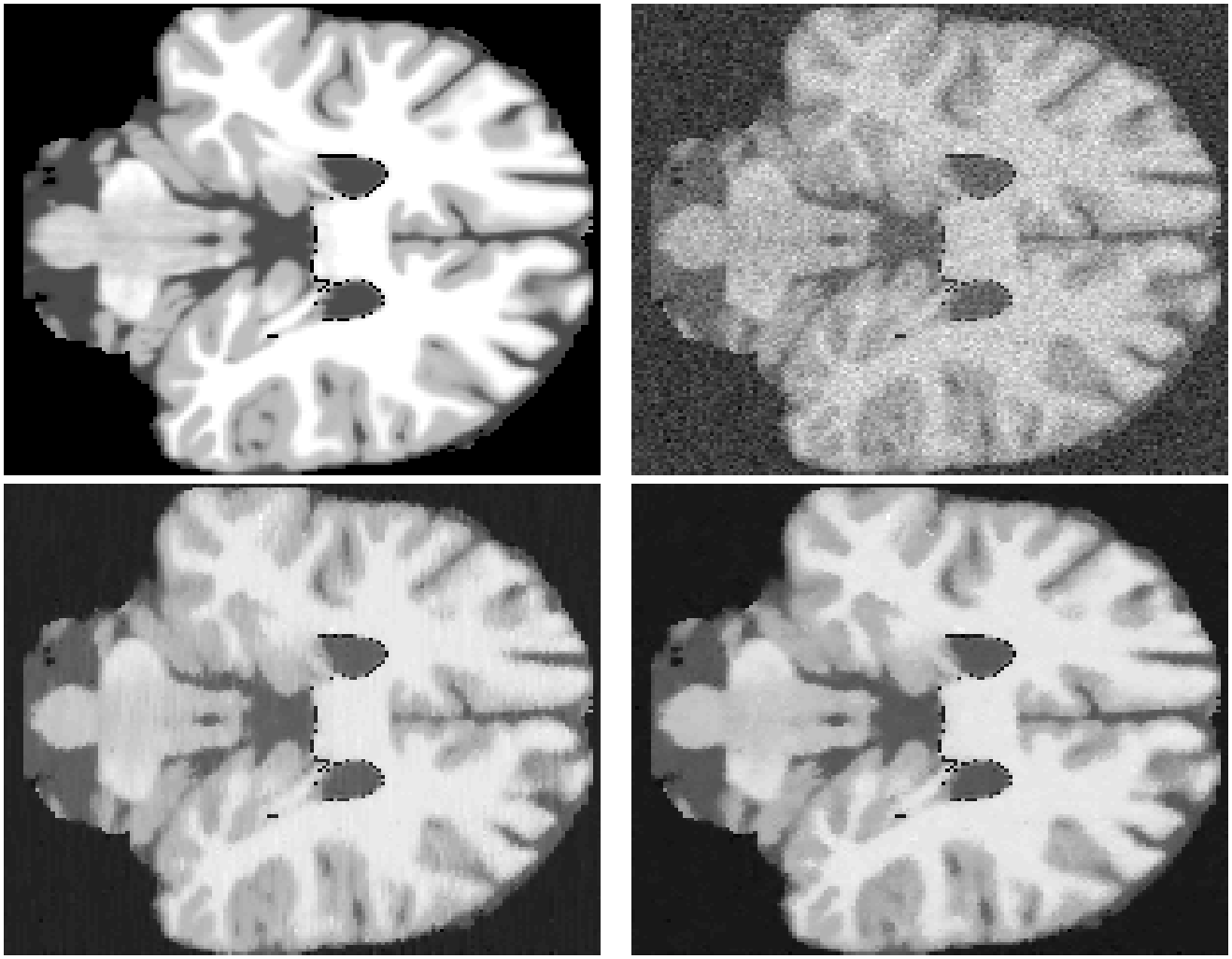

Fig. 3. In this figure we show the difference between application of the two-dimensional SADCT along slices in a 3D image, and application of the three-dimensional SADCT. In the upper left corner the true image is shown. In the upper right corner the noisy image is shown. In the lower left corner we see the result after application of the quazi-3D algorithm, and in the lower right corner we see the result after application of the full 3D algorithm proposed in this paper. See also Figure 4

used the gradient directions given by the rows of the matrix

$$\begin{pmatrix} 1.0000 & 0.0000 & 1.0000 \\ -1.0000 & 0.0000 & 1.0000 \\ 0.0000 & 1.0000 & 1.0000 \\ 0.0000 & 1.0000 & -1.0000 \\ 1.0000 & 1.0000 & 0.0000 \\ -1.0000 & 1.0000 & 0.0000 \end{pmatrix} \tag{14}$$

Example 3: *3D tensor valued real brain data*

We also show preliminary results on real diffusion tensor images of a healthy volunteer. The images have been acquired by a 3T MR scanner using the same gradient directions as in the previous example. In the example we use only one acquisition of each diffusion weighted image, thus the total number of acquisitions in this example is seven. We note that the denoised result is a bit over-regularized. In a later version of this paper we will however perform numerical tests with a number of different parameter-configurations, and compare the results with averages of multiple acquisitions.

V. Discussion

In this paper we have extended the SADCT methods from 2D scalar valued images to 3D tensor valued images. We have shown preliminary numerical experiments on both 3D scalar valued images and 3D tensor valued images. The numerical studies indicates that the SADCT framework can successfully be applied to tensor valued images. However, to make more solid conclusions we will perform more numerical experiments, in particular quantitative measurements of the quality of the denoised images.

The method is local in nature, thus it is natural and easy to paralellize the algorithm to speed up the calculations.

References

[1] *Shape-Adaptive DCT for Denoising and Image Reconstruction*, Proceedings of SPIE Electronic Imaging 2006, Image Processing: Algorithms

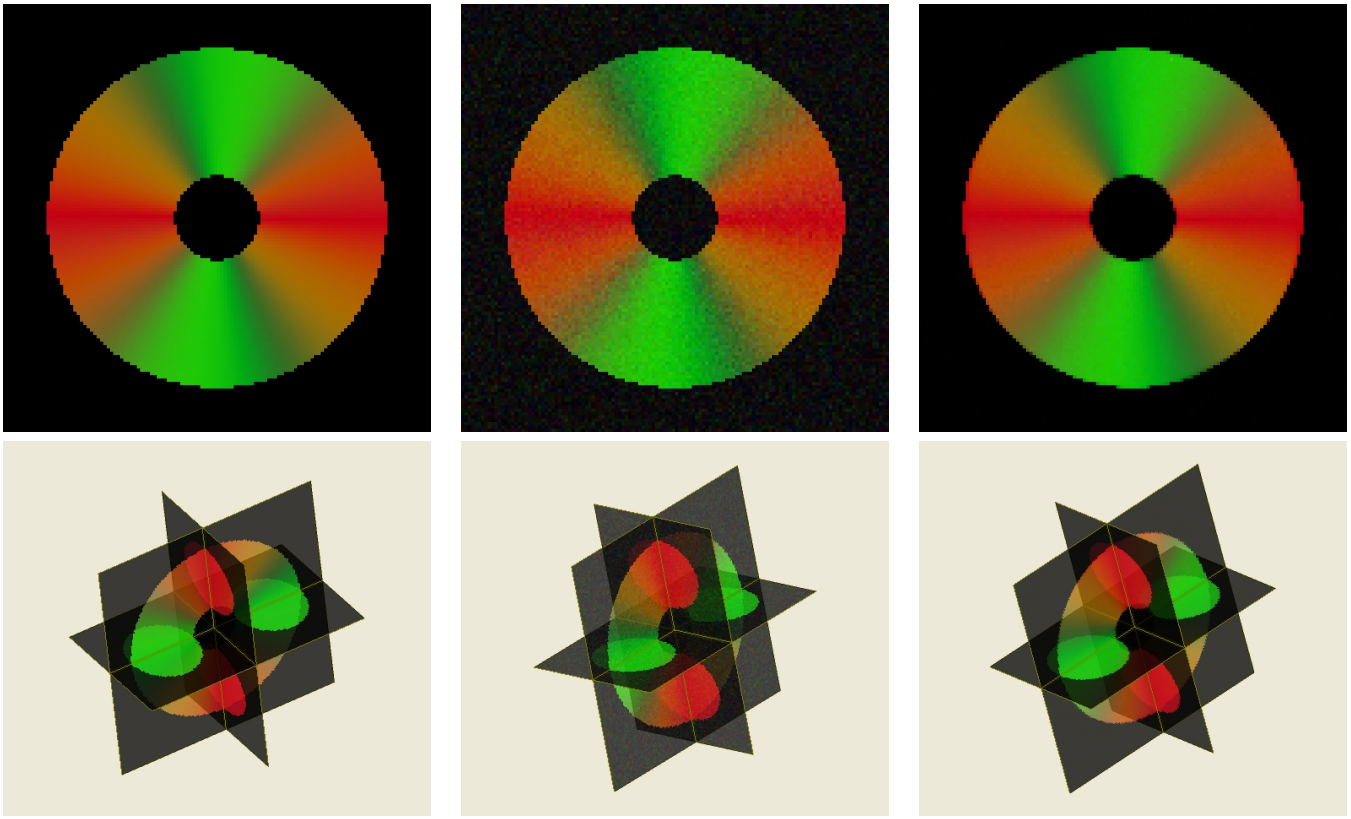

Fig. 5. In this figure we show the result from application of the proposed method to a synthetically produced diffusion tensor dataset where the object is a torus. The leftmost column shows the true data, the middle column shows the noisy data and the rightmost column shows the denoised data. All figures are color coded FA plots.

and Systems V, Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland, 2006.

[2] C. Chefd'hotel, D. Tchumperlé, R. Deriche, and O. Faugeras. Regularizing flows for constrained matrix-valued images. *Journal of Mathematical Imaging and Vision*, 20:147–162, 2004.

[3] Oddvar Christiansen, Tin-Man Lee, Johan Lie, Usha Sinha, and Tony F. Chan. Total variation regularization of matrix valued images. *Manuscript submitted to International Journal of Biomedical Imaging*, 2006.

[4] C. A. Cocosco, V. Kollokian, and A. C. Evans. Brainweb: Online interface to a 3d mri simulated brain database. *NeuroImage*, 5(4), 1997.

[5] A. Foi, K. Dabov, V. Katkovnik, and K. Egiazarian. Shape-adaptive dct for denoising and image reconstruction. In *Proc. SPIE Electronic Imaging 2006, Image Processing: Algorithms and Systems V, San Jose*, 2006.

[6] Alessandro Foi. *Anisotropic nonparametric image processing: theory, algorithms and applications*. PhD thesis, Dipartimento di Matematica, Politecnico di Milano, April 2005.

[7] J. Foong, M. Maier, C. A. Clark, G. J. Barker, D. H. Miller, and M. A. Ron. Neuropathological abnormalities of the corpus callosum in schizophrenia: a diffusion tensor imaging study. *J Neurol Neurosurg Psychiatry*, 68:242–244, 2000.

[8] D. Goldberg-Zimring, AUJ Mewes, M Maddah, and SK Warfield. Diffusion tensor magnetic resonance imaging in multiple sclerosis. *Journal of Neuroimaging*, 15(S):68–81, 2005.

[9] Klaus R. Hahn, Sergej Prigarin, Susanne Heim, and Khader Hasan. *Random Noise in Diffusion Tensor Imaging, its destructive impact and some corrections*, volume Visualization and Processing of Tensor Fields of *Mathematics + Visualization*, chapter 6. Springer, 2006.

[10] IEEE SP/CAS and Tampere International Center for Signal Processing. *POINTWISE SHAPE-ADAPTIVE DCT AS AN OVERCOMPLETE DENOISING TOOL*, number 5 in The International Workshop on Spectral Methods and Multirate Signal Processing, Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland, June 2005.

[11] Vladimir Katkovnik, Karen Egiazarian, and Jaakko Astola. Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule. *J. Math. Imaging Vision*, 16(3):223–235, 2002. Nonlinear image processing and pattern recognition, XII (San Jose, CA, 2001).

[12] Gordon Kindlmann. Teem: Tools to process and visualize scientific data and images. http://teem.sourceforge.net/.

[13] K. O. Lim, M. Hedehus, M. Moseley, A De Crespigny, E. V. Sullivan, and A. Pfefferbaum. Compromised white matter tract integrity in schizophrenia inferred from diffusion tensor imaging. *ARCHIVES OF GENERAL PSYCHIATRY*, 56(4):367–374, 1999.

[14] T. McGraw, B.C. Vemuri, Y. Chen, and T. Mareci M. Rao b. Dt-mri denoising and neuronal fiber tracking. *Medical Image Analysis*, 8:95–111, 2004.

[15] Susumi Mori and Peter C.M. van Zijl. Fiber tracking: principles and strategies - a techical review. *NMR Biomed*, 15:468–480, 2002.

[16] M. Niethammer, Raul San-Jose Estepar, Sylvain Bouix, Martha Shenton, and C.-F. Westin. On diffusion tensor estimation. In *28th IEEE EMBS*, pages 2622–2625, New York City, NY, USA, September 2006.

[17] Thomas Sikora. Low complexity shape-adaptive dct for coding of arbitrarily shaped image segments. *Signal Processing: Image Communication*, 7:381–395, 1995.

[18] E. O. Stejskal. Use of spin echoes in a pulsed magnetic-field gradient to study anisotropic, restricted diffusion and flow. *The Journal of Chemical Physics*, 43(10):3597–3603, 1965.

[19] E. O. Stejskal and J. E. Tanner. Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient. *The Journal of Chemical Physics*, 42(1):288–292, 1965.

[20] David Tschumperle and Rachid Deriche. Variational frameworks for DT-MRI estimation,regularization and visualization. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*, 2003.

[21] Z. Wang, B.C Vemuri, and Y Chen and T Mareci. A constrained variational principle for direct estimation and smoothing of the diffusion tensor field from complex dwi. *IEEE Transactions on Medical Imaging*, 23(8):930 – 939, 2004.

Fig. 4. In this figure we show a small portion of the results in Figure 3. The upper figure shows the result after application of the quazi-3D algorithm and the lower figure shows the result after application of the full 3D algorithm proposed in this paper.

[22] Joachim Weickert and Thomas Brox. Diffusion and regularization of vector- and matrix-valued images. Universitat des Saarlandes, Fachrichtung 6.1 Mathematik, Preprint No. 58, 2002.

[23] Joachim Weickert and Hans Hagen (Eds.). *Visualization and Processing of Tensor Fields*. Mathematics and Visualization. Springer, 2005.

[24] M. Welk, J. Weickert, F. Becker, C. Schnörr, C. Feddern, and B. Burgeth. Median and related local filters for tensor-valued images. *Signal Processing*, 87:291–308, 2007.

[25] Leonid Zhukov and Alan H. Barr. Oriented tensor reconstruction: Tracing neural pathways from diffusion tensor mri. IEEE Visualization 2002, 2002.

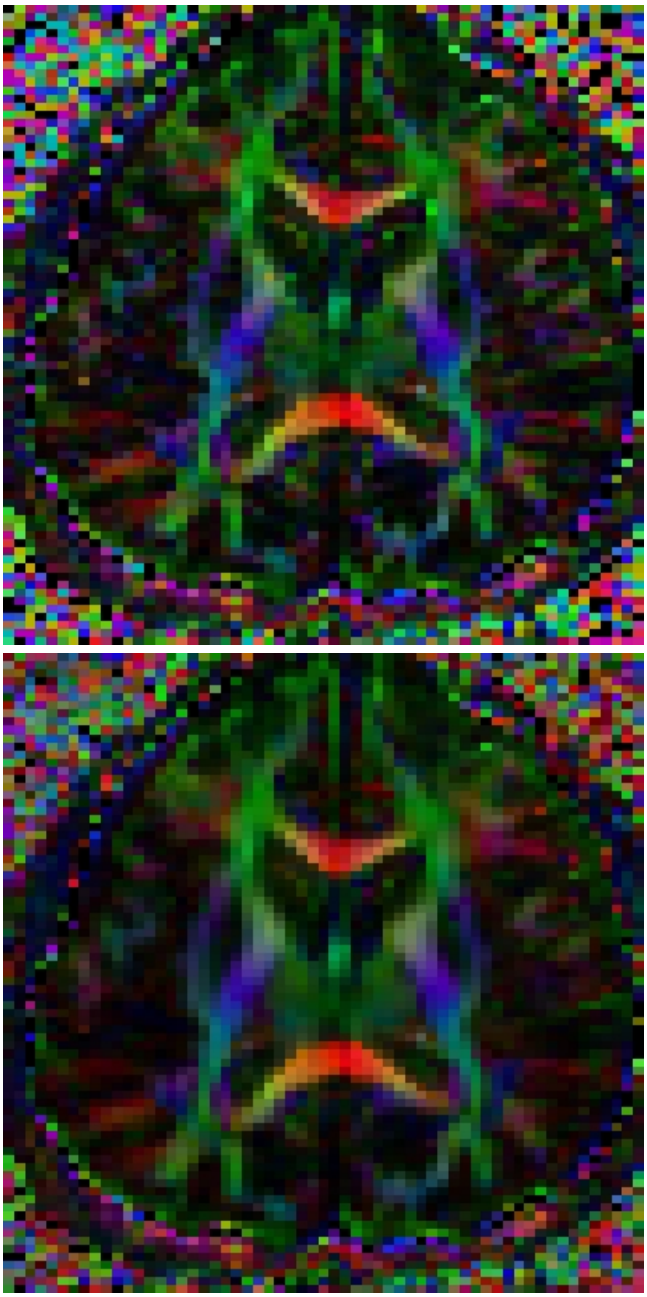

Fig. 6. In this example we show a color coded FA plot of a slice of a noisy (upper) and denoised (lower) DTI dataset from a real human brain.