Josef Michael KOLLER

Image Shading Correction via TGV and DCT with Application to MRI

Master Thesis

Graz University of Technology

Institute of Medical Engineering
Stremayrgasse 16/III, A - 8010 Graz
Head: Univ.-Prof.Dipl.-Ing.Dr.techn. Rudolf Stollberger

Supervisor: Univ.-Prof.Dipl.-Ing.Dr.techn. Rudolf Stollberger

Graz, October 2016

Image Shading Correction via TGV and DCT with Application to MRI

Abstract

Magnetic Resonance Imaging is a measurement method which produces representative data of the insight of human bodies. In this work a retrospective correction method for the inhomogeneity artifact is composed of other approved image processing techniques, extended and evaluated.

The proposed method performs denoising and simultaneous bias estimation by the TGV-L^1 Primal-Dual algorithm for volumetric data. Bias correction is done by solving a Poisson Equation via a direct form solution in the cosine domain. The algorithm is implemented for general TGV order and efficiently calculates most operations in parallel.

Several image types are processed including 3D MR measurement data. The results include a quantitative comparison to ground truth data and metric values. Additionally a qualitative evaluation by intensity profile line plots and an estimate of the probability density functions is given.

Under the assumption of piecewise constant objects of interest and a slowly and smoothly varying bias field the proposed method successfully estimates higher-order bias fields. The method outperforms the reference method N4ITK in several aspects and may improve the performance of other imaging tasks, and could be applied to several other imaging modalities.

Keywords: MR Inhomogeneity correction, bias, shading, TGV, DCT

Bildbeleuchtungskorrektur via TGV und DCT mit Anwendung auf MRI

Zusammenfassung

Magnetresonanzbildgebung ist eine Messmethode, welche representative Daten vom inneren des menschlichen Körpers produziert. In dieser Arbeit wird eine retrospektive Korrekturmethode für das Inhomogenitätenartefakt aus anderen bewährten Bildverarbeitungstechniken zusammengestellt, erweitert und evaluiert.

Die vorgeschlagene Methode entfernt gleichzeitig Signalrauschen und schätzt das Inhomogenitätenfeld mittels TGV-L^1 Primal-Dual Algorithmus für Volumsdaten. Die Inhomogenitätenkorrektur wird durch die Lösung einer Poisson-Gleichung in der Kosinusdomäne durchgeführt. Der Algorithmus ist für eine generelle Ordnung von TGV implementiert und die meisten Operationen werden parallel ausgeführt.

Verschiedene Bildtypen – auch 3D MR Messdaten – werden verarbeitet. Die Resultate enthalten einen quantitativen Vergleich mit Ground Truth Daten und die Berechnung von mehreren Metrikwerten. Weiters wird eine qualitative Evaluierung durch visuelle Darstellung von Intensitätsprofilen entlang einer Linie und der Wahrscheinlichkeitsdichtefunktion durchgeführt.

Unter der Annahme von stückweise konstanten Objekten und einen sich langsam und stetig ändernden Inhomogenitätenfeldes schätzt die vorgeschlagene Variationsmethode erfolgreich Inhomogenitäten höherer Ordnung. Die Methode übertrifft in einigen Aspekten die Referenzmethode N4ITK, und könnte die Leistung von anderen Bildverarbeitungsaufgaben verbessern und auch auf andere Bildmodalitäten angewendet werden.

Schlüsselworte: MR Inhomogenitätenkorrektur, Trift, Schatten, TGV, DCT

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

________________ ________________ ____________________

Place Date Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

________________ ________________ ____________________

Ort Datum Unterschrift

Danksagung

Ich danke meinen Eltern, welche mir eine wunderbare Ausbildung ermöglicht haben, und mir die notwendige Freiheit gaben, um mich selbstständig zu entwickeln. Bei meiner gesamten Familie will ich mich für vielseitige Unterstützung bedanken. Meinen Freunden danke ich für das richtige Maß an Ablenkung, Geduld und Aufmunterung. Besonders danke ich Monika Gabernig, MSc. die mich liebevoll durch das Studium begleitet hat.

Für gute Zusammenarbeit im Zuge des Standard-MRT Projekts will ich mich bei allen Kollegen am LBI und IMT bedanken. Hierbei gilt ein besonderer Dank DI Dr. Martin Urschler und DI Dr. Alexander Bornik, welche mir bei der Vorarbeit zu dieser Arbeit maßgeblich geholfen haben. Auch an dieser Stelle half mein Betreuer Univ.-Prof. DI Dr. Rudolf Stollberger bei anregenden Diskussionen, kritischen Fragestellungen und der Ausarbeitung von Lösungswegen.

Ich fühle mich geehrt, diese interessante und spannende Aufgabe mit Experten auf dem Gebiet der Bildverarbeitung von medizinischen Daten erarbeiten zu dürfen. Weiters danke ich Christof Sirk, BSc. und DI Andreas Schwarzl für eine tolle gemeinsame Studienzeit und das Korrekturlesen dieser Diplomarbeit.

Diese Arbeit wurde zum Teil durch Forschungsförderungsmittel des Landes Steiermark (HTI:TechForMed ABT08-22-T-7/2013-13) unterstützt.

Contents

List of Abbreviations and Symbols

Abbreviations	**Description**
CG	Conjugate Gradient
CV	Coefficient of Variation
DCT	Discrete Cosine Transform
DICOM	Digital Imaging and Communications in Medicine
FT	Fourier Transform
GPU	Graphic Processing Unit
HSV	Hue Saturation Value
KDE	Kernel Density Estimation
MR	Magnet Resonance
PDE	Partial Differential Equation
PDF	Probability Density Function
RMSE	Root Mean Square Error
TE	Echo Time
TGV	Total Generalized Variation
TI	Inversion Time
TR	Repetition Time
TV	Total Variation
T1	Spin-Lattice Relaxation Time
T2	Spin-Spin Relaxation Time

Table 1: List of Abbreviations

Symbol	**Meaning**
k	TGV order
$\|x\|$	L^1-norm of x
α_i	maximum step size for the i-th derivative
λ	optimization weight for the data term
N_{voxel}	voxel count
σ_k	kernel bandwidth for KDE
σ_s	parameter for the spatial distance for Bilateral Filter
σ_i	parameter for the intensity difference for Bilateral Filter
∇_i	gradient operator of order i
div_i	divergence operator of order i
$\triangle$	Laplace Operator
ρ_H	proton density
$f_{downsampling}$	downsampling factor
μ	mean voxel intensity of an image

Table 2: List of Symbols

1 Introduction

1.1 Problem Description

Images of the inside of the human body provide crucial information for medical diagnosis, therapy and research. Magnetic Resonance Imaging is one non-invasive measurement method to produce such information. Besides MR, todays medicine benefits from other imaging modalities like microscopy, computer tomography and ultrasound. A challenging problem – which is common to these techniques – is the intensity inhomogeneity. This phenomena is discussed in the review of Vovk et al. 2007 [1].

It is an undesired signal component due to an inhomogeneous excitation field or coil sensitivity profile. In MR it especially appears with increasing field strength. This artifact is commonly defined to be slowly and smoothly varying. In other words the spatial intensity gradient is low and it does not contain sharp edges.

Particularly for quantitative analysis intensity values must not depend on the location of the object of interest within the measurement device. Furthermore a bias correction method should preserve the absolute intensity values of the image. Inhomogeneity correction improves the performance of other image processing tasks like segmentation, registration and classification as well as visual inspection.

1.2 Common Solutions

1.2.1 Prospective Methods

Several methods to estimate and correct the coil sensitivity profile have been developed. During device manufacturing inhomogeneities of the strong static field are minimized by passive shimming. The coil field arises due to the current flow through the circuit paths of the conductor board. Additionally all other paramagnetic elements interact with the field. Therefore the field inhomogeneity is an important criterion in device design. Active shimming is an automatic step which is usually performed once the measurement object changes. The system measures the inhomogeneity and optimizes the coil parameters.

Additionally other retrospective actions are performed. The field can be calibrated by ground truth data. Phantoms are accurately manufactured measurement objects. For example doped water, pure oils or gels in a test-tube. The inhomogeneity field can be approximated by the measurement of multiple equal samples split across the image, segmentation of the content, averaging and spline interpolation.

MR scan parameters influence the strength of this bias artifact. Belaroussia et al. 2006 [2] mention the influence of the slice distance, repetition time and number of echoes. The authors list some correction methods and propose a protocol for the evaluation of these. Special pulse sequences – for example inversion recovery – compensate the effect of inhomogeneities partly. The coil sensitivity estimation is also an important part in

Parallel Imaging. This technique uses multiple coils for the image acquisition. In Knoll et al. 2012 [3] the TGV algorithm is used for the reconstruction of such data.

Inhomogeneities additionally originate due to the measurement object. Certainly this is intended since the signal should represent object structures. Nevertheless there are additional slow variations which are object-induced.

1.2.2 Retrospective Methods

Retrospective inhomogeneity correction methods aim to decompose the measured image. This is an underdetermined problem. Each voxel intensity value of the bias and object volume is a degree of freedom and only the measured intensities are known. Such a problem can only be solved with prior knowledge or assumptions about the imaging sequence, field distribution or the imaged objects. The common signal model of MR defines a multiplicative bias field and an independent additive noise term [1]. It is frequently found in literature that the multiplicative composition is transformed to an additive one by taking the logarithm [1, 4–13].

The common assumptions used in retrospective methods is that the bias field is slowly and smoothly varying and the imaged objects are piecewise constant. Hence it follows that sharp edges in the measured image correspond to the objects of interest. Usually the shape of the bias field is controlled by the parameters of the correction method.

The simplest method for the estimation of the bias field is a lowpass filter. This technique assumes the bias to contain low frequency information and that a cutoff frequency exists, which splits the bias from object information. The lowpass can be implemented in the Fourier domain by an ideal or a butterworth filter.

Homomorphic Unsharp Masking first blurs the image by a Gaussian filter. This low frequency image is subtracted in the logarithm domain. The resulting high frequency information is multiplied by a constant factor and added to the source image. This technique weights edge information stronger and thus reduces slow variations. Axel et al. 1987 [14] do not use the logarithm transform and apply this filter for intensity correction of surface-coil MR images.

The N3 algorithm by Sled et al. 1998 [15] is an implementation of an inverse filter for MR inhomogeneity correction. It approximates the bias field in the logarithm domain by spline interpolation and iteratively performs deconvolution of the intensity histogram. A multi-scale extension of N3 is the N4ITK method by Tustison et al. 2010 [16]. It uses a Gaussian Pyramid of the source image and successively fits the bias field on increasing scales. Manjón et al. 2007 [17] also use spline interpolation with a multi-scale approach to estimate the bias field, and optimize their parameters by an entropy related performance measure.

In Land and McCann 1971 [18] the retinex model is introduced and Horn 1974 further describes the theoretical background. Retinex is an acronym for retina and cortex. The human visual perception does somehow include an illumination correction method. We recognize equal objects under different illumination situations. This phenomena is denoted by the term color constancy. Nevertheless the retinex algorithms – which are described in the following paragraphs – basically only have the idea of a pixel-wise composition in common. Most methods process the input image in the logarithm domain. In that way a multiplicative composition becomes additive.

Similar to Homomorphic Unsharp Masking the Single Scale Retinex method uses a lowpass kernel to extract shading information. Multiscale Retinex contains a linear combination of multiple kernels. These two methods are used by Shen and Hwang 2009 [19], Chao et al. 2012 [20], Wang and Huang 2014 [21] as well as Morel et al. 2014 [22]. Zhao et al. 2012 [5] derive a close-form solution for a quadratic energy functional based on retinex.

A method commonly found as a preprocessor in face recognition tasks is Empirical Mode Decomposition. Xie 2014 [23] describe this method which includes special basis functions known as Intrinsic Mode Functions. The author explains the method for one-dimensional data, combines it with the retinex idea and applies it for face recognition. Damerval et al. 2005 [24] extended Empirical Model Decomposition to two-dimensional images. In Liang and Si 2015 [25] this method is used for MR inhomogeneity correction.

Variational image processing methods are widely applied and approved today. Tikanov et al. 1992 [26] describe PDE based methods to solve imaging tasks by the minimization of a Lagrange function. Maximum a posteriori probability (MAP) methods include a model of the probability distribution of the intensity values. Gaussian, Rayleigh, Gibbs and Rician distributions are used. Fu et al. 2015 [13] show how a MAP problem is transformed into an energy minimization problem. Another probabilistic approach is used by Wang et al. 2014 [27] and compared to several other retinex methods. The energy functional usually consists of multiple weighted integral terms of the image data. These include the L^1- or L^2-norm of image components (source, denoised, bias, deshaded), the gradient or higher order derivatives of those.

The pioneer work of Kimmel et al. 2003 [6] is based on an energy function which uses the L^2-norm of the gradient of the illumination, the deshaded image and the gradient of the deshaded image. The authors choose a special Steepest Descent algorithm to minimize their cost function. Their work shows that bias correction can be done with few assumptions and parameter values on a PDE basis. Ma and Osher 2010 [7] introduce the L^1-norm of the resulting image in the energy function. Their motivation is to preserve edges in the resulting image similar to the ROF model by Rudin et al. 1992 [28]. This requires a more complex optimization routine, but highly improves the quality of the result. They use the Bregman methods to solve the minimization problem. In Morel et al. 2010 [8] a much simpler optimization algorithm is introduced, which calculates the bias by the thresholded gradient and the FT. This method is further improved by Limare et al. 2011 [29]. Ma et al. 2011 [9] include the L^1-norm of the deshaded image in the energy functional. They extended their optimization method to use the thresholded gradient to split bias from object information. Their method is evaluated with MR medical images and compared to the N3 algorithm by Sled et al. 1998 [15].

Liang and Zhang 2015 [11] add additional terms to the energy functional. Their model includes the deshaded and the bias component. They use the split inexact Uzawa method which is a variant of the alternating direction method of multipliers (ADMM). Similarity to TGV^2-L^1 by Bredies et al. 2010 [30] is also discussed by these authors.

MR inhomogeneity correction is also combined with simultaneous segmentation by variational level set methods. Such algorithms are used by Verma et al. 2012 [31], Shahvaran et al. 2012 [32] and Ivanovska et al. 2016 [33] for the segmentation of biological tissue.

Gilboa and Osher 2009 [34] introduce non-local differential operators to variational image processing. Non-local operators were described earlier by Yaroslavsky 1994 [35]. Zhang et al. 2013 [10] combine non-local differentiation and retinex and Zosso et al. 2015 [36] further generalize this idea.

2 Methods

In this work an additive decomposition method is evaluated. The fundamental cost function is based on the idea of Liang and Zhang 2015 [11]. It is optimized by the Primal-Dual algorithm developed by Chambolle and Pock 2010 [37]. This algorithm is extended to second order TGV in Knoll et al. 2011 [38] and applied to the problem of image denoising and MR reconstruction.

The first step of the proposed PDE inhomogeneity correction method is to minimize the functional stated in Equation 1. It only differs to the formulation of the image denoising problem by Knoll et al. 2011 [38] in the norm of the data term, and the parameters α_0 and α_1. The stated cost function consists of three parts including the L^1-norm of scalar fields and vector fields, which are weighted by control parameters. First of all it minimizes the distance from the denoised image u to the measured image f. According to the notation of Chambolle and Pock 2010 [37] this term is weighted by the parameter λ. Low values of this parameter allow the denoised image to highly differ from the input image. Higher values of λ increase the weight of the difference in the overall cost function and therefore result in more similar images and less denoising.

Next the regularization functional includes the L^1-norm of the gradient of the denoised image subtracted by a vector field v. The subtracted vector field v is not just totally equal to the gradient of the denoised image. The last term minimizes the second order gradient of this vector field. These two terms counteract in the properties of the final vector field v. The compromise is balanced by the specific values chosen for the elements of the parameter vector α. High values of α_1 result in less edges, which is equal to a smoother field. If the previous term is weighted stronger the field v will contain the edge information of u. Knoll et al. 2011 [38] prevent the image from containing the staircasing artifact which usually evolves by using the TV-norm of the gradient.

$$\underset{u,v}{min} \left\{ \lambda \int_{\Omega} |u - f| \, dx + \alpha_0 \int_{\Omega} |\nabla u - v| \, dx + \alpha_1 \int_{\Omega} |\nabla_2 v| \, dx \right\} \tag{1}$$

This cost function in Equation 1 describes two properties of the denoised image u. The vector field v is introduces as an auxiliary variable to contain piecewise smooth image information and to prevent the TV-norm from minimizing all gradients of the denoised image.

Liang and Zhang 2015 [11] have invented a PDE retinex method. The authors describe the similarity of their cost function to the TGV-L^2 denoising functional of Bredies et al. 2010 [30]. They interpret the vector field v as the gradient of the inhomogeneity field l, and perform inhomogeneity correction of MR images by using the simple composition model stated in Equation 3. While these authors estimate the images r and l at the same time, TGV-L^1 (Equation 1) results in the gradient of the bias v and the denoised image u. Liang and Zhang 2015 [11] mention the idea to solve a Poisson Equation for the transformation of the vector field v to the bias scalarfield l. This results from the derivative of the Lagrange Equation of minimizing the L^2-norm of the difference of the gradient of the bias image l and this vector field v (Equation 2). The Poisson solver – which is implemented in this work – is based on the idea of Limare et al. 2011 [29] and described in Section 2.3 in more detail.

$$\underset{l}{min}\left\{\int_{\Omega} \|\nabla l - v\| \, dx\right\} \tag{2}$$

$$u = r + l \tag{3}$$

After solving the minimization of Equation 1 and then Equation 2, the estimate of the bias-free image r can be calculated by subtracting the estimated bias l from the estimated noise-free image u. For better understanding it is mentioned at this point, that the amount of denoising is controlled by the parameter λ and the smoothness of the bias by α_1.

Prior experiments have shown that the logarithm - which is usually used to transform the multiplicative combination of r and l to an additive one - is not necessary and omitted in this work. The MR inhomogeneity is therefore not approximated by an multiplicative term in the logarithm domain, but directly as an additive one.

The following sections describe each single step of the inhomogeneity correction method under investigation. For some presented samples in Section 3 prior denoising is applied to the data to simplify the interpretation of the results. Bias estimation is done based on the TGV-functional which is described in Section 2.2 in detail. The DCT Poisson solver is derived step-by-step in Section 2.3.

With these components an inhomogeneity correction method is composed. In Section 2.4 this method is extended by the idea of processing the algorithm on lower scales. This additionally controls the shape of the estimated bias field and highly improves the required computation time and memory usage. Another extension of the method is described in Section 2.5. This powerful Primal-Dual improvement is very specific to the implementation choosen and allows to fully exploit the hardware resources.

Pseudo code of the overall method and its modules is listed in Section 2.6. These algorithms are implemented in CUDA (Nvidia Corporation, Santa Clara, USA) to efficiently estimate inhomogeneity components of 3D data. The proposed method is qualitatively evaluated by visual inspection of the resulting images, histograms and line profile plots. Quantitative evaluation is done based on several metric values, which are further described in Section 2.7.

2.1 Preprocessing

The following procedure outlines the general way how preprocessing is done. If the pipeline for a specific sample differs from this description the reason will be explained in Section 3. First a region of interest of the measurement data is focused. This includes the extraction of in planar coordinates and the desired slices. Then the intensity values are normalized to the range 0 to 1. The prepocessing filter, described in Section 2.1.2, requires parameters which depend on the absolute intensity spectrum values.

The proposed method performs denoising and shading extraction simultaneously. Nevertheless evaluation is concentrated on the deshading properties. Therefore prior denoising is done. First TGV-L^1 removes most of the small structures and then a Bilateral Filter is applied. With less details the resulting line profile plots are easier to analyze. For example Figure 11 includes such a profile plot, which representatively shows the effect of the proposed inhomogeneity correction algorithm.

2.1.1 Total Generalized Variation Denoising

Since TGV-L^1 is part of the overall method no additional implementation is necessary to use this procedure for prior denoising. The implemented bias correction method can therefore additionally be used for prior denoising, because it outputs the estimated noise and bias free volumes. The denoising problem is stated in Equation 4 as a minimization of the L^1-data term and the TGV. The data term is scaled by the factor λ and the regularization term is defined in Equation 8. The parameter λ balances the regularization and the data term. This parameter is adjusted to the level of noise and desired scale of interest.

$$\underset{u}{min}\left\{\lambda\int_{\Omega}|u-f|\,dx+TGV_{\alpha}^{k}\left(u\right)\right\} \tag{4}$$

A pseudo-code for TGV-L^1 of third and higher order is listed in Algorithm 4. Based on this code the algorithm is implement in CUDA for 3D images. The TGV functional is described in Section 2.2 and discussed in Section 4.1.7.

2.1.2 Bilateral Filtering

Additional denoising is done using a Bilateral Filter. This filter is described by Tomasi and Manduchi 1998 [39]. Parameters are the kernel size, the bandwidth of the exponential term for the spatial distance σ_s and the bandwidth of the second exponential term for the intensity difference σ_i. Equation 5 shows the underlying model. The denoised intensity value at the position x is calculated by the summation of weighted pixel values in a neighbourhood N_x around x. Also the factor K_x depends on the location of the pixel. It normalizes the sum of the weights for each neighbourhood.

$$I_{denoised}\left(x\right) = \frac{1}{K_x} \sum_{k}^{N_x} I_k \, e^{\frac{(x_k - x)^2}{\sigma_k}} \, e^{\frac{(I(x_k) - I(x))^2}{\sigma_i}} \tag{5}$$

This algorithm is an extension of the Gaussian Filter. The second exponential term controls the contrast of edges which should not be blurred. It controls which edges are preserved in the denoised image. The implementation is done in CUDA since the resulting intensity value of each pixel can be calculated in parallel.

Bilaterial Filtering is also used as an edge-preserving lowpass filter for Multiscale Retinex by Chang and Bai 2015 [12] and as the solution of a quadratic minimization problem by Elad 2005 [40]. In this work it is only used for denoising purposes.

2.2 Shading Estimation

2.2.1 Total Generalized Variation

The TGV functional defined by Bredies et al. 2010 [30] is shown in Equation 6. This regularization term is generally formulated for d-dimensional real data. It includes the supremum of the integral of the product of the data u and the k-th order divergence of a vector field v. Constraints are defined by the maximum norm of symmetric tensors of v of order up to $k - 1$.

$$\begin{aligned} &TGV_{\alpha}^{k}\left(u\right) = \\ &= sup \left\{ \int_{\Omega} u \, div^k v \, dx \,|\, v \in C_c^k\left(\Omega, Sym^k\left(\mathbb{R}^d\right)\right), \left\|div^l v\right\|_{\infty} \leq \alpha_l, l = 0, \ldots, k-1 \right\} \end{aligned} \tag{6}$$

TGV is successfully used for several imaging tasks like denoising, reconstruction, zooming, inpainting and compression. This regularization term minimizes the L^1-norm of higher order derivatives. For digital image processing quantization to discrete spatial positions is usually done. The following sections will describe this functional for first, second, third and higher order in more detail.

2.2.2 First Order Total Generalized Variation

First order TGV is equal to the TV term weighted by the factor α. It is defined as the L^1-norm of the gradient according to Equation 7. For two-dimensional images the gradient produces a vector field with two elements in each field point. The L^1-norm is the magnitude of this vector. Integrating all these magnitudes gives overall information about intensity changes in the image. For example zero-based normally distributed noise or salt-and-pepper noise increases the TV. Such noise variants can be successfully removed by the Primal-Dual algorithm by Chambolle and Pock 2010 [37].

$$TGV_{\alpha}^{1}(u) = \alpha TV(u) = \alpha \int_{\Omega} |\nabla u| \, dx \tag{7}$$

TV is usually combined with a data term. It contains the L^1- or L^2-norm of the difference to the input data (for example Equation 4). The data term and the regularization are usually weighted by parameters. These parameters balance between the desired properties. In case of first order TGV these properties are the similarity to the input image and a low TV. Implementations of this minimization extract piecewise constant objects. For input images containing a slowly and smoothly varying bias the algorithm outputs a compromise which usually includes the staircasing artifact. This first order TGV functional produces piecewise constant objects, but such model fitting is maybe not the indented result.

2.2.3 Second Order Total Generalized Variation

In the work of Knoll et al. 2011 [38] the second order TGV functional is used for MR image denoising and reconstruction. This functional extracts piecewise linear objects. It is shown in Equation 8. In comparison to the previously stated Equation 7 it includes a vector field v which is subtracted from the gradient before the L^1-norm and integration is performed. Furthermore the functional aims to minimize the L^1-norm of the second order gradient of this vector field v. The two energy terms are weighted by the factors α_0 and α_1 respectively.

$$TGV_{\alpha}^{2}(u) = \underset{v}{min} \left\{ \alpha_0 \int_{\Omega} |\nabla u - v| \, dx + \alpha_1 \int_{\Omega} |\nabla_2 v| \, dx \right\} \tag{8}$$

Starting at second order, the TGV energy functional itself is a minimization. The Primal-Dual algorithm alternates between minimization of the TGV and the data term. In the Primal-Dual implementation the maximum step sizes α_i are used in the projection subroutine which is part of Algorithm 4. Each point in the vector field is normalized separately. If the magnitude of the vector divided by the maximum step size is greater than 1, the vector is shrinked. This enforces the constraint defined in Equation 6.

Second order TGV solves the problem of staircasing artifacts. It does not prefer piecewise constant images for u which would directly minimize the TV. This method rather uses the additional vector field v to hold the non-piecewise constant information and therefore indirectly minimizes the TV. A more detailed discussion can be found in Section 4.1.7.

2.2.4 Third Order Total Generalized Variation

Third order TGV is used for denoising in Bredies et al. 2010 [30]. The resulting images contain piecewise quadratic objects. The implementation of TGV^3-L^1 of this work is based on the energy functional in Equation 9. This definition is based on the general TGV minimization functional stated in Holler and Kunisch 2014 [41]. An additional vector field w is introduced and the L^1-norm of the third order gradient is weighted by α_2.

$$TGV_{\alpha}^{3}(u) = \underset{v}{min} \left\{ \alpha_1 \int_{\Omega} |\nabla u - v| \, dx + \alpha_0 \int_{\Omega} |\nabla_2 v - w| \, dx + \alpha_2 \int_{\Omega} |\nabla_3 w| \, dx \right\} \tag{9}$$

In comparison to TGV^2-L^1 the 3D Primal-Dual implementation of TGV^3-L^1 introduces 30 additional variables of the size of the input image. These are necessary for the gradient, divergence, projection and overrelaxation operation. Analysis of the memory usage is done in Section 3.3.1. The resulting denoised image u and the divergence of the vector field v are used in the next step for shading extraction. That means shading extraction does not differ between second and higher order TGV-DCT. The difference is how the values of the vector field v are calculated. Note that if the input image would contain data which results in a zero vector field for w (by the minimization of the L^1-norm of the third order gradient) the result would be equal to the one produced by second order TGV.

This method requires more memory and computation time per iteration. But it produces good results in less iterations. Surely there exists a problem usually known as overfitting. So it is clear at this point that increasing the order of TGV will not generally end up in better performance of the algorithm.

Based on the second order TGV^2-L^1 Matlab (Mathworks Inc., Natick, USA) implementation for 2D images by Chambolle and Pock 2010 [37] the algorithm is extended for 3D data. Then a third order version is implemented based on Equation 9. This requires additional primal and dual variables, but most of the subroutines needed have already been implemented at this time. The main extension is the third order gradient and divergence operation. Pseudo code for these operations is listed in Algorithms 8 and 9.

2.2.5 Third and Higher Order Total Generalized Variation

By increasing the order of the TGV definition and the Primal-Dual implementation a general pattern becomes visible. Based on the assumption that the gradient and divergence are symmetric, any higher order derivative is identical to the one for the third order. Those functionals transform a six-dimensional image to a vector in the same domain. Subroutines for the gradient, divergence, primal and dual update can be reused. Similar to the step from TGV^2 to TGV^3 increasing the order by one requires 30 additional image volume variables. The general minimization function is shown in Equation 10 based on Holler and Kunisch 2014 [41]. The third and higher order TGV Primal-Dual pseudo-code is shown in Algorithm 4. By default the elements of the maximum step size vector α_i are set to $(i+1)$ for $i = 0 \ldots k-1$. Thus higher order information is added up faster.

This choice for the parameter α is discussed later in Section 4.1.7. These values for the maximum step size lead to acceptable convergence for the processed images.

$$\begin{aligned} & TGV_{\alpha}^{k}(u) = \\ & = \min_{v} \left\{ \alpha_0 \int_{\Omega} |\nabla u - v| \, dx + \alpha_1 \int_{\Omega} |\nabla_2 v - w_0| \, dx + \sum_{i=0,\, w_{k-2}=0}^{k-3} \alpha_{i+2} \int_{\Omega} |\nabla_3 w_i - w_{i+1}| \, dx \right\} \end{aligned} \tag{10}$$

The first term in Equation 10 – which is weighted by α_0 – shows that the algorithm aims to minimize the TV of the image. If it would just minimize the TV – as is done by first order TGV in Equation 7 – piecewise constant images would be approximated. For biased images this would lead to the staircasing artifact. Therefore higher order derivatives are subtracted successively.

Second order TGV in Equation 8 corrects the TV term by the second order derivative only. This is sufficient for purely linear bias fields. In other words, if the shape of the bias field could be described by a linear polynomial with respect to the spatial dimensions, the second order method would successfully perform inhomogeneity correction. In Equation 10 the second term includes the second order derivative and additionally a vector field – which holds third and higher order information – is subtracted.

The third term in Equation 10 describes successive correction of higher order derivatives in a general form. This is possible because it is assumed that the higher order gradient and divergence operation can be approximated by the respective symmetrized third order operation. Section 2.6 contains a more detailed description of this assumption. Note that these higher order vector fields have the same variable name w with an index for further identification.

Finally TGV$^{\mathrm{k}}$ contains k parameters for the maximum step size which weight k terms of the energy functional. These terms contain the L^1-norm of corrected gradients of order up to k-1. Minimization is done by the Primal-Dual algorithm which alternates between primal and dual update of all included terms. Additionally a L^1-data term is added to this optimization. The resulting TGV$^{\mathrm{k}}$-L^1 procedure is listed in Algorithm 4.

By using the regularization functional in Equation 10, image denoising and shading estimation is performed simultaneously. Noise is assumed to be uncorrelated and additive. TGV$^{\mathrm{k}}$-L^1 approximates piecewise regions of order k-1 to the image and therefore removes such noise which becomes part of the residual.

Shading estimation is done under the following conditions. First the objects of interest are assumed to be piecewise constant. In other words, the measured objects consist of homogeneous regions which are connected by sharp edges. An artificial example of such an image is shown in Figure 1. Second the bias field is assumed to contain all higher order information. The samples in Section 3 demonstrate the performance of this shading estimation routine. Interpretation and discussion is done in Section 4.

Note that the energy functional above (Equation 10) is the minimization of the L^1-norm of the gradient of the denoised image u subtracted by a vector field v. It does neither include the estimated bias-free image nor the bias field directly. Only the gradient of the bias is approximated. For inhomogeneity correction – which is interpreted as the decomposition of piecewise constant regions and higher order information in this work – there are additional steps necessary. The following Section 2.3 contains two possible solutions for shading extraction based on $\mathrm{TGV}^{\mathrm{k}}$-$\mathrm{L}^1$. These alternatives are evaluated and compared later in Section 3.3.7.

2.3 Shading Extraction

2.3.1 Poisson Equation

As described above the gradient of the bias field l is approximated by the vector field v. Equation 2 states this approximation in the form of the minimization of the L^2-norm of the difference of those two variables. Derivation of the Lagrange Equation with respect to the spatial dimensions leads to a Poisson Equation which is shown in Equation 11. The extraction of the bias field l is therefore done by the inversion of the Laplace Operator (Equation 12).

$$\triangle l = div\, v \tag{11}$$

$$l = \triangle^{-1} \left(div\, v \right) \tag{12}$$

2.3.2 Laplace Operator

The first step to numerically solve the Poisson Equation is the discretization of the Laplace Operator. According to the first Primal-Dual algorithm by Chambolle and Pock 2010 [37] the gradient is approximated by the forward difference with Neumann boundary conditions and the divergence by the backward difference with Dirichlet Boundary-Conditions. The Laplace Operator is approximated by the divergence of the gradient.

2D images are stored into a vector in row-major order. 3D images are reshaped and contain the 2D slices one after another. The Laplace convolution kernel shown in Table 3 transforms this image vector. At the boundaries of the image the center value of the kernel contains the number of neighbouring pixels which are inside the image and have non-zero values in the kernel. This results from the boundary conditions previously mentioned. In the corners of a 2D image there are two and at the other boundaries there are three neighbours containing non-zero values.

0	-1	0
-1	4	-1
0	-1	0

Table 3: The discrete Laplace Operator kernel for a 2D image.

For 3D volumes the Laplace Operator is performed by the convolution with a 3x3x3 kernel. For each resulting voxel – except at the boundaries – the intensity values of six neighbour voxels are taken under consideration.

2.3.3 Fourier Transform

A direct form solution can be stated for the inversion of the Laplace Operator by using the differentiation rule of the Fourier Transformation. Equation 13 defines the forward difference in the spatial and Fourier domain. For the one-dimensional case the next Equation 14 performs the Laplace Operator in the Fourier Domain and simplifies the two exponential terms by the cosine function. According to these identities Equation 15 performs the inversion of the Laplace Operator for 3D images.

$$u\left[x+1\right]-u\left[x\right]=\nabla_x u=FT^{-1}\left(FT\left(u\right)\,\left(e^{ik_x}-1\right)\right) \tag{13}$$

$$\begin{aligned}2\,u\left[x\right]-u\left[x+1\right]-u\left[x-1\right]=\triangle_x u=FT^{-1}\left(FT\left(u\right)\,\left(2-e^{ik_x}-e^{-ik_x}\right)\right)=\\=FT^{-1}\left(FT\left(u\right)\,\left(2-2\,cos\left(k_x\right)\right)\right)\end{aligned} \tag{14}$$

$$\triangle^{-1}=FT^{-1}\left(\frac{FT\left(u\right)}{6-2\,\left(cos\left(k_x\right)+cos\left(k_y\right)+cos\left(k_z\right)\right)}\right) \tag{15}$$

2.3.4 Discrete Cosine Transform

Another simplification can be done by the assumption of symmetric images. In other words the image is assumed to extend symmetrically across the boundaries. Even real data does not contain an imaginary part in the Fourier Domain. Laplace Operator inversion is therefore implemented according to Equation 15 by using the cosine transform instead of the complex Fourier Transform. A similar step is included in the 2D shading correction algorithm of Limare et al. 2011 [29]. The authors use the FFTW-library by Frigo and Johnson 2005 [42] for the cosine transform.

In this work the transformation is implemented in CUDA for 2D and 3D data. The operation is separated to each single dimension to achieve better performance. The forward transform is a DCT-II and the inverse transform a DCT-IV. Additionally a normalization factor is introduced to the inverse transform. Sequential forward and backward transformation does not scale the image's intensity values. The pseudo-code for these transformations is listed in Algorithms 5 and 6.

Only the gradient of the bias v is known from solving Equation 1. There are infinite possible solutions to Equation 2, because any constant image added to l does not change the gradient. Therefore the proposed correction method can estimate the shape of the inhomogeneity, but a global constant offset is missing in the result.

The scalar bias field can be calculated by using Equation 15. Division by zero is omitted by setting the first voxel value in the Fourier Domain to zero [29]. In the spatial domain this sets the mean intensity value to zero. Finally the bias component is subtracted from the denoised image.

2.3.5 Alternative Poisson Solver

An iterative reference method for the DCT Poisson solver is additionally developed. With the boundary conditions mentioned above the Laplace Operator is written in matrix form. The Conjugate Gradient algorithm is implemented in CUDA to invert the linear equation system. Hestenes and Stiefel 1952 [43] developed this numerical method. The matrix multiplication is done by simply performing the forward model (Laplace Operator) using a CUDA kernel. The pseudo-code for this method is listed in Algorithm 7.

2.4 Downsampled TGV-DCT Bias Correction

The algorithm is further extended by using the fundamental assumption that the bias field is slowly and smoothly varying. Similar to an option of the reference method N4ITK by Tustison et al. 2010 [16] bias correction is done on a lower scale. The input image size is decreased for processing. The downsampled image size becomes the original image size multiplied by a downsampling factor $f_{downsampling}$. This further decreases the computation time and makes the estimation of the maximum step size parameter α easier. For example it is noticed during experiments that a general factor for α of 1 leads to good convergence for image sizes up to 128 pixels and a factor of 0.1 for image sizes of about 256. The downsampling step generally has lowpass character. Therefore focusing of low frequency information is done earlier.

After the TGV-DCT algorithm converges with the downsampled image (and mask) volume, the resulting bias field is upsampled. The algorithm additionally ensures that the upsampling procedure ends in the same image size as the one of the original image. This prevents rounding errors with the sampling factor. The upscaled bias field is then subtracted from the input image. Note that this procedure does not perform denoising.

The implementation performs cubic B-spline interpolation between existing voxels to resample the input volume. These splines contain three conditions at every node. The intensity values, the first and the second spatial derivative of connected polynomials are equal respectively. This additionally improves smoothness of the estimated bias field.

The downsampled version of the bias correction algorithm introduces one additional parameter $f_{downsampling}$, but highly reduces computation time and the amount of required memory. Therefore processing higher order TGV-DCT algorithms and bigger volumetric data becomes feasible.

2.5 Masked TGV-DCT Bias Correction

Another extension of the algorithm is introduced to further improve the performance. During preprocessing focusing on a region of interest is usually done. First this involves the extraction of a volume of interest. A start- and an end-index for each spatial dimension is chosen. Next the calculation of a foreground mask is performed.

In this work mask calculation is done by a region growing segmentation algorithm. Several segments with multiple seed points are the input of the algorithm. Each seed point can define a tolerance value. The algorithm recursively grows to unsegmented left, right, top, bottom, front and back direct neighbours until it finds a value greater than the tolerance. To omit stack overflows a maximum recursion depth is used. If this depth is reached the current voxel is added to a queue and later handled as a new seed point of the current segment. Mask generation is done based on the input image, the non-local gradient of the input image and also based on region growing segmentation results, inverted and dilated images.

Some volumes have millions of voxels, but only half of them hold foreground information. Biological tissue is not aligned on straight lines and because of that the slices expand differently and the overall volume includes many background voxels. The proposed algorithm is implemented in CUDA. Algorithm 4 describes all operations in detail. To calculate only necessary foreground values the following extension is done.

The algorithm loops over all voxels of the mask and builds up several index-vectors. First it searches all indices inside of the mask. Next multiple subsets of these indices are additionally stored. To efficiently calculate the gradient- and divergence-operations prior determination of boundary voxels is done. The forward and backward difference in 3D requires $2x2x3$ index vectors. For example the forward difference in x-direction is calculated for voxels inside of the mask which are not located at the right boundary. Another subroutine applies the Neumann zero flux boundary condition in parallel.

Other operations than the gradient and divergence (i.a. projection, overrelaxation...) are done for all voxels inside of the mask. The implementation therefore stores 13 index-vectors built out of the mask volume. The kernel functions are specially designed and do not contain any branches. Consequently the threads are in sync, which fully exploit the resources of the graphic card.

2.6 Algorithms

The overall method is formulated in Algorithm 1. For a choice of the step size vector α and denoising parameter λ the TGV-L^1 Primal-Dual algorithm of order $k \geq 2$ is processed. The output of this subroutine is the denoised image u and vector field v. Next the inversion of the Laplace Operator is done. Finally the bias field is subtracted from the denoised image. The second step is either done by Algorithm 2 or Algorithm 3. In Section 3.3.7 these two alternatives are compared in detail.

The third or higher order TGV-L^1 Primal-Dual pseudo-code is listed in Algorithm 4. In this work the TGV-L^1 Primal-Dual algorithm of general order and the DCT and CG Bias Extraction algorithms are implemented for 3D images in CUDA. The forward transform of the DCT is listed in Algorithm 5 and the inverse transform in Algorithm 6. Pseudo-code for CG is shown in Algorithm 7. The variable N_D is used to denote the size of the image in the dimension D.

Algorithm 1 TGV Bias Correction

```
choose $k \geq 2,\ \alpha_i > 0,\ i = 0 \ldots k-1,\ \lambda > 0$
$u, v \leftarrow TGV_\alpha^k L_1\,(f,\ \lambda)$
$l \leftarrow \triangle^{-1}\,(div\,v)$
$r \leftarrow u - l$
```

Algorithm 2 DCT Bias Extraction

```
$l \leftarrow DCT^{-1}\left(\frac{DCT(div\,v)}{6-2\,(cos(k_x)+cos(k_y)+cos(k_z))}\right),\ k_D \leftarrow \pi\,\frac{i_D}{N_D},\ i_D = 1 \ldots N_D - 1,\ D = \{x, y, z\}$
```

Algorithm 3 CG Bias Extraction

```
choose $\epsilon_{convergence} > 0$
define the forward model $M_{laplace}$
$l \leftarrow CG\,(div\,v,\ M_{laplace},\ \epsilon_{convergence})$
```

Algorithm 4 $TGV_\alpha^k - L_1$ Primal-Dual of order $k \geq 3$ for 3D images

```
function $projection\,(p,\,\alpha_i)$
  $p \leftarrow \frac{p}{max\left(1,\frac{\|p\|}{\alpha_i}\right)}$
end
function $dataL_1\,(u,\,f,\,\tau,\,\lambda)$
  $u \leftarrow u - \tau\lambda,\; u - f > \tau\lambda$
  $u \leftarrow u + \tau\lambda,\; u - f < \tau\lambda$
  $u \leftarrow f,\; \|u - f\| < \tau\lambda$
end
function $overrelaxation\,(u,\,u_{previous})$
  $\bar{u} \leftarrow 2\,u - u_{previous}$
end
function $TGV_\alpha^k L_1\,(f,\,\lambda)$
  $u,\,\bar{u} \leftarrow f$
  $p,\,v,\,\bar{v} \leftarrow \mathbf{0} \in \mathbb{R}^{N_{voxel}\cdot 3}$
  $q,\,w_i,\,\bar{w_i},\,r_i \leftarrow \mathbf{0} \in \mathbb{R}^{N_{voxel}\cdot 6},\; i = 0\dots k-3$
  choose $\tau, \sigma > 0$
  repeat
    $p \leftarrow projection\,((p + \sigma\,(\nabla\bar{u} - \bar{v}))\,,\,\alpha_0)$
    $u_{previous} \leftarrow u$
    $u \leftarrow dataL_1\,((u + \tau\,div\,p)\,,\,f,\,\tau,\,\lambda)$
    $\bar{u} \leftarrow overrelaxation\,(u,\,u_{previous})$
    $q \leftarrow projection\,((q + \sigma\,(\nabla_2\bar{v} - \bar{w_0}))\,,\,\alpha_1)$
    $v_{previous} \leftarrow v$
    $v \leftarrow v + \tau\;(p + div_2\,q)$
    $\bar{v} \leftarrow overrelaxation\,(v,\,v_{previous})$
    $r_i \leftarrow projection\,((r_i + \sigma\,(\nabla_3\bar{w_i} - w_{\bar{i+1}}))\,,\,\alpha_{i+2}),\; i = 0\dots k-3,\; w_{k-2} = 0$
    $w_{i,previous} \leftarrow w_i,\; i = 0\dots k-3$
    $w_0 \leftarrow w_0 + \tau\;(q + div_3\,r_0)$
    $w_i \leftarrow w_i + \tau\;(r_{i-1} + div_3\,r_i),\; i = 1\dots k-3$
    $\bar{w_i} \leftarrow overrelaxation\,(w_i,\,w_{i,previous}),\; i = 0\dots k-3$
  until convergence of $u$
end
```

Algorithm 5 DCT for 3D images using separability in each dimension

```
function $DCT_x\left(f_{input}\right)$
 for $x = 0\ldots N_x - 1,\ y = 0\ldots N_y - 1,\ z = 0\ldots N_z - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow 2\, f_{input}\left[x,\ y,\ z\right]\ cos\left(\pi\ \left(k_x + \frac{1}{2}\right)\ \frac{x}{N_x}\right),\ k_x = 0\ldots N_x - 1$
end
function $DCT_y\left(f_{input}\right)$
 for $x = 0\ldots N_x - 1,\ y = 0\ldots N_y - 1,\ z = 0\ldots N_z - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow 2\, f_{input}\left[x,\ y,\ z\right]\ cos\left(\pi\ \left(k_y + \frac{1}{2}\right)\ \frac{y}{N_y}\right),\ k_y = 0\ldots N_y - 1$
end
function $DCT_z\left(f_{input}\right)$
 for $x = 0\ldots N_x - 1,\ y = 0\ldots N_y - 1,\ z = 0\ldots N_z - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow 2\, f_{input}\left[x,\ y,\ z\right]\ cos\left(\pi\ \left(k_z + \frac{1}{2}\right)\ \frac{z}{N_z}\right),\ k_z = 0\ldots N_z - 1$
end
function $DCT\left(f_{input}\right)$
 $f_{output} \leftarrow DCT_x\left(f_{input}\right)$
 $f_{output} \leftarrow DCT_y\left(f_{output}\right)$
 $f_{output} \leftarrow DCT_z\left(f_{output}\right)$
end
```

Algorithm 6 Inverse DCT for 3D images using separability in each dimension

```
function $DCT_x^{-1}\left(f_{input}\right)$
 for $x = 0\ldots N_x - 1,\ y = 0\ldots N_y - 1,\ z = 0\ldots N_z - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow f_{input}\left[x,\ y,\ z\right]\ cos\left(\pi\ \left(x + \frac{1}{2}\right)\ \frac{k_x}{N_x}\right),\ k_x = 1\ldots N_x - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow 2\, f_{output}\left[x,\ y,\ z\right] + f_{input}\left[0,\ y,\ z\right]$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow \frac{f_{output}\left[x,\, y,\, z\right]}{2\, N_x}$
end
function $DCT_y^{-1}\left(f_{input}\right)$
 for $x = 0\ldots N_x - 1,\ y = 0\ldots N_y - 1,\ z = 0\ldots N_z - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow f_{input}\left[x,\ y,\ z\right]\ cos\left(\pi\ \left(y + \frac{1}{2}\right)\ \frac{k_y}{N_y}\right),\ k_y = 1\ldots N_y - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow 2\, f_{output}\left[x,\ y,\ z\right] + f_{input}\left[x,\ 0,\ z\right]$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow \frac{f_{output}\left[x,\, y,\, z\right]}{2\, N_y}$
end
function $DCT_z^{-1}\left(f_{input}\right)$
 for $x = 0\ldots N_x - 1,\ y = 0\ldots N_y - 1,\ z = 0\ldots N_z - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow f_{input}\left[x,\ y,\ z\right]\ cos\left(\pi\ \left(z + \frac{1}{2}\right)\ \frac{k_z}{N_z}\right),\ k_z = 1\ldots N_z - 1$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow 2\, f_{output}\left[x,\ y,\ z\right] + f_{input}\left[x,\ y,\ 0\right]$
  $f_{output}\left[x,\ y,\ z\right] \leftarrow \frac{f_{output}\left[x,\, y,\, z\right]}{2\, N_z}$
end
function $DCT^{-1}\left(f_{input}\right)$
 $f_{output} \leftarrow DCT_x^{-1}\left(f_{input}\right)$
 $f_{output} \leftarrow DCT_y^{-1}\left(f_{output}\right)$
 $f_{output} \leftarrow DCT_z^{-1}\left(f_{output}\right)$
end
```

Algorithm 7 Conjugate Gradient

```
function $CG\,(f,\, A,\, \epsilon)$
 $x \leftarrow 0$
 $r \leftarrow A\,x - f$
 $p \leftarrow r$
 $\rho_0 \leftarrow \|r\|$
 stop if $\rho_0 < \epsilon^2$
 for $k = 0 \ldots N_{voxel} - 2$
  $s \leftarrow A\,p$
  $\sigma \leftarrow (s,\, p)$
  $a \leftarrow \frac{\rho}{\sigma}$
  $x \leftarrow x - a\,p$
  $r \leftarrow r - a\,s$
  $\rho_{k+1} \leftarrow \|r\|$
  stop if $\rho_{k+1} < \epsilon^2 \rho_0$
  $b \leftarrow \frac{\rho_{k+1}}{\rho_k}$
  $p \leftarrow r + b\,p$
end
```

The CG Algorithm 7 solves a linear equation system which is described by the vector of the right side f and the symmetric and positive definite matrix A. Note that the initial guess of $x = 0$ is usually not part of the CG algorithm itself. The third parameter ϵ is the convergence threshold which is compared to the L^2-norm of the residual vector r. The algorithm iteratively performs the forward model with the current search direction p and calculates a step size a via the inner-product. In the worst case it converges after N_{voxel}-1 iterations. In this work the CG algorithm is used to invert the discrete Laplace Operator. Section 3.3.7 contains results of this algorithm and comparison to the DCT Poisson solver. In that way the performance is evaluated.

In general all pixel-wise operations are performed in CUDA. The forward DCT in Algorithm 5 sequentially transforms the input image according to each dimension. Therefore after DCT_x and DCT_y a CUDA thread synchronization is necessary. The same applies to the inverse DCT in Algorithm 6. These implementations are based on the description on the FFTW website [44].

The $TGV_\alpha^k - L_1$ Primal-Dual method in Algorithm 4 performs most operations in parallel. Synchronization is basically only needed before and after each gradient and divergence subroutine call. The resulting voxels are dependent on their neighbourhood and therefore multiple Primal-Dual iterations can not be calculated in parallel. The three subroutines *projection*, *dataL*$_1$ and *overrelaxation* do not need a synchronization point. For the variables w_i, r_i and all auxiliary ones a dynamic array of images is allocated. The higher order gradient and divergence operations are described in the following paragraph.

According to Bredies et al. 2010 [30] the TGV discretization is based to the forward δ_+ and backward δ_- difference. Depending on the order, the higher order gradient and divergence alternate between those two differences. For example the third order gradient uses forward difference and the fourth order gradient performs backward difference operations. For three spatial dimensions there are six image variables if it is assumed that other cross terms are nearly zero. The gradient operation for even orders (backward difference) is listed in Algorithm 8 and the divergence in Algorithm 9. This makes it possible to efficiently implement the $TGV_\alpha^k - L_1$ Primal-Dual algorithm of order three and above.

Algorithm 8 Gradient operation for 3D images for even order ($k = 4, 6 \ldots$)

```
$\nabla : \mathbb{R}^{N_{voxel} \cdot 6} \rightarrow \mathbb{R}^{N_{voxel} \cdot 6}$
$\nabla_x := \delta_{x-} \left( \bar{w}_x \right)$
$\nabla_y := \delta_{y-} \left( \bar{w}_y \right)$
$\nabla_{xy} := \frac{1}{2} \left( \delta_{x-} \left( \bar{w}_{xy} \right) + \delta_{y-} \left( \bar{w}_{xy} \right) \right)$
$\nabla_z := \delta_{z-} \left( \bar{w}_z \right)$
$\nabla_{xz} := \frac{1}{2} \left( \delta_{x-} \left( \bar{w}_{xz} \right) + \delta_{z-} \left( \bar{w}_{xz} \right) \right)$
$\nabla_{yz} := \frac{1}{2} \left( \delta_{y-} \left( \bar{w}_{yz} \right) + \delta_{z-} \left( \bar{w}_{yz} \right) \right)$
```

Algorithm 9 Divergence operation for 3D images for even order ($k = 4, 6 \ldots$)

```
$div : \mathbb{R}^{N_{voxel} \cdot 6} \rightarrow \mathbb{R}^{N_{voxel} \cdot 6}$
$div_x := \delta_{x+} \left( r_x \right)$
$div_y := \delta_{y+} \left( r_y \right)$
$div_{xy} := \delta_{x+} \left( r_{xy} \right) + \delta_{y+} \left( r_{xy} \right)$
$div_z := \delta_{z+} \left( r_z \right)$
$div_{xz} := \delta_{x+} \left( r_{xz} \right) + \delta_{z+} \left( r_{xz} \right)$
$div_{yz} := \delta_{y+} \left( r_{yz} \right) + \delta_{z+} \left( r_{yz} \right)$
```

2.7 Evaluation

Qualitative evalution is done by visual inspection of the image, a representative line profile plot and the histogram of the foreground pixels. The coefficient of variation, total variation and entropy are calculated to present quantitative measures of the performance. These evaluation methods for bias correction algorithms are discussed in Arnold et al. 2001 [45], Belaroussi et al. 2006 [2] and Vovk et al. 2007 [1].

2.7.1 Kernel Density Estimation

To calculate meaningful entropy values the probability density function of the image is approximated by Kernel Density Estimation. A quadratic Epanechnik-Kernel with bandwidth $\sigma_k = 3\%$ of the intensity spectrum is generally used and the spectrum is quantized to $\sqrt{N_{voxel}}$ steps. This KDE kernel is analyzed in detail by Huang and Kong 2012 [46]. For each spectrum value a CUDA function loops over the image and sums up the voxel values weighted by the kernel. Additionally uniform, Gaussian and cosine kernels are implemented. KDE results in a smoother and better comparable PDF. An example is included in Figure 13.

Background voxels are usually not taken into account for KDE estimation. The background can be ignored by specifying an intensity window or a mask image.

2.7.2 Entropy Calculation

Based on the probability density function the entropy H is calculated according to Equation 16. The implementation only takes probability values p greater than 10^{-7} for the summation of the intensity values a.

$$H = -\sum_{a} \left(p\left(a\right)\, log_2\left(p\left(a\right)\right)\right) \tag{16}$$

From the viewpoint of the histogram narrow peaks have low entropy. The bias field is assumed to make peaks wider and thus increase entropy. In the extreme case the correction algorithm minimizes the information of the image to a single narrow peak for the foreground pixels.

The following comparisons in Section 3 generally apply a mask image and a window containing all intensities of voxels inside of the mask.

2.7.3 Coefficient of Variation

Another metric commonly used for the quantitative evaluation of bias correction is the coefficient of variation. It is calculated due to Equation 17 as the fraction of the standard deviation and the mean μ of the intensity values.

$$CV = \frac{std}{\mu} \tag{17}$$

Bias correction is assumed to minimize the standard deviation. To make the metric value less dependent on the absolute intensity values normalization by the mean is done.

2.7.4 Total Variation

The third metric used is the TV. The implementation performs a gradient operation and sums up the magnitude value in each voxel. This metric is used to evaluate the degree of piecewise-constant objects of the image. Homogeneous regions contain low TV values. The bias field also increases this metric value.

2.7.5 Local Contrast

The previously described metrics entropy, CV and TV have a theoretical minimum for a single intensity value in the image. However the image is assumed to contain multiple piecewise-constant regions. One may perform segmentation during the evaluation and calculate the metric values for each region separately. But the performance of segmentation would highly influence the result. And the segmentation performance may depend on the previously processed algorithm. In this work another approach is used. Additionally to the metrics entropy, CV and TV the local contrast of several tissue classes is analyzed. It is simply the intensity difference of neighbouring regions and is approximated based on the line profile plot.

The bias correction method should therefore reduce the values of CV, TV and entropy. Furthermore contrast between several piecewise-constant regions should be preserved. In other words it should simply remove the bias without destroying the information of the objects of interest. The combination of the listed metrics quantifies this objective.

2.7.6 Trivial Solutions of Inhomogeneity Correction

Small Constant Factor If an inhomogeneity correction method would just multiply all voxels by a constant much smaller than 1, the metric values described above would change in the following way. The entropy would not be changed, if the applied window is also scaled. The CV would also not be altered, because the mean and the standard deviation are multiplied by the same factor. But such a global operation would scale the TV and the local contrast.

An Additive Constant Another method may just add a global constant to all voxels. In fact this would decrease the CV, because the mean increases and the standard deviation remains the same. Again the entropy is not altered, if the window is shifted by the same constant. And the TV also remains the same, because the gradient operation removes such global additive constant.

Subtract or Divide the Input Image Another trivial solution is to subtract the input from itself. Dividing the input by itself is similar. Surely that results in a perfectly uniform image. At this point this trivial solution may seem to be total nonsense.

But in fact many inhomogeneity methods roughly perform such operations. The trick is to filter specific information. For example Homomorphic Unsharp Masking [14] removes low frequency components. If one does not well configure the proposed algorithm of this work, it may result in a nearly uniform image. The parameter α_1 in Equation 1 could be configured too low. The last regularization term would not have any effect, and the estimated bias field would be very similar to the denoised image.

Avoid Trivial Solutions To avoid the trivial solutions previously mentioned in the quantitative evaluation of inhomogeneity correction methods, a combination of the listed metrics is used. For qualitative evaluation the pixel values of images shown in this work are always rescaled from the minimum to the maximum intensity value. A white pixel corresponds to the maximum and a black pixel to the minimum. The histogram is based on an intensity-windowed spectrum as described above. And line profile plots include the input and output image with the same axis scaling.

Some authors additionally perform some kind of statistic normalization. For example Limare et al. 2011 [29] attempt to rescale the output image to preserve the mean and the standard deviation. The authors argue that the 'global contrast' remains the same in that way. This is not necessary with the actions described in the previous paragraph. And in this work the CV is used to quantify the amount of inhomogeneity.

Trivial additive constants could be avoided by preserving the mean only. Mean preserving can either be done by an additive constant or a global factor. Additive mean preserving has the drawback, that a method could apply a global factor smaller than 1, which result in a decreasing CV. Multiplicative mean preserving would change the local contrast and the TV.

Quantitative evaluation avoids trivial global factors and additive constants by performing KDE as described in Section 2.7.1 and no additional transformation is done.

3 Results

The method is applied to 2D simulated images, 3D simulated MR volumes, standard test images, color photographs and 3D MR tissue measurements. The following sections include such data which is processed by the methods described above. The results are later discussed in Section 4.

3.1 Simulated Images

3.1.1 Ground Truth Evaluation

Ground truth data is essential for a proper evaluation. With known bias field and object component the result of the algorithm can be evaluated with a quantitative metric. The first sample in Figure 1 is a piecewise constant composition of geometrical objects with an additive field. There are triangles, squares and circles which contain a single intensity. The bias is generated using the formula and parameters of Keeling et al. 2011 [47]. It contains linear and quadratic terms with respect to the spatial dimensions.

Rescaling the intensity values is omitted for this sample. The calculation of the RMSE is done based on the difference of the ground truth to the estimate. Because of the missing constant offset – which is described in Section 2.3.4 – the zero-frequency is removed from this error. In this work this metric is used to evaluate the error of the shape of the bias field only. Additionally no denoising is done for this sample. The estimated bias is subtracted from the input image, instead of using the denoised image.

The TGV^2-DCT algorithm nearly perfectly estimates and corrects the bias field. Parameters are set to $\lambda = 1$, $\alpha_0 = 0.1$ and $\alpha_1 = 0.2$. This line profile plot of Figure 1 contains the input in blue and output in green. The line starts at the top-left and reaches to the bottom-right corner. It is additionally painted as an overlay in the two corresponding images. Image dimensions are $256x256$ and the intensity values are in the range of 0 to 2. The RMSE of the image and the bias is 0.153e-3.

Sample 2 in Figure 2 evaluates the performance with respect to a multiplicative inhomogeneity component which spawns between 0.4 and 1.6. The input image has the same intensity range than sample 1. The previously used algorithm with the same parameter values is used. Inhomogeneity reduction is clearly visible in the line profile plot of Figure 2.

For comparison of the estimated bias field the ground truth image is subtracted from the input to get an additive bias. The RMSE of the image and the bias is 0.963e-1.

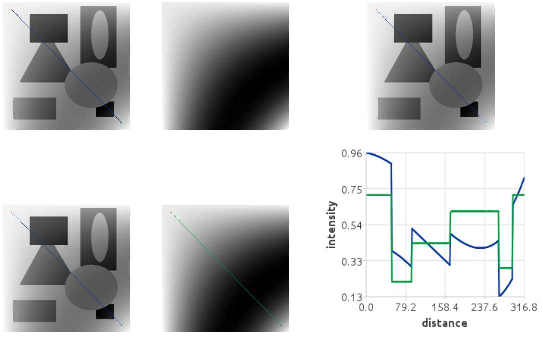


Figure 1: Sample 1: From top-left to bottom-right: estimated bias-free image, bias field, algorithm input, output, estimated bias field, line profile plot of input (blue) and output (green).

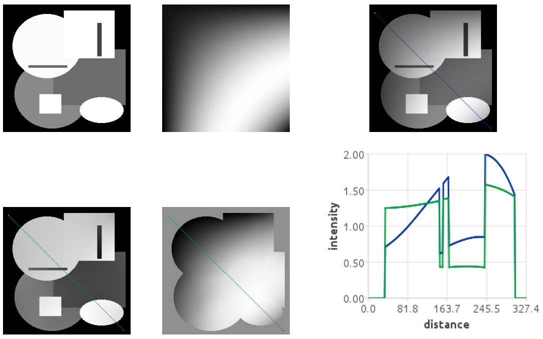


Figure 2: Sample 2: From top-left to bottom-right: estimated bias-free image, multiplicative bias field, algorithm input, output, estimated additive bias field, line profile plot of input (blue) and output (green).

These samples are additionally processed by the reference method N4ITK. This method is developed by Tustison et al. 2010 [16] and corrects inhomogeneities of MR images. The results for the two artificial samples do not look very promising and are neither listed nor compared to the proposed method. Evaluation and comparison is later done using MR samples in Section 3.3.

The two samples above demonstrate the algorithms performance for the simple case of perfectly piecewise constant measurement objects and a bias field which is generated by a single polynomial function. The second sample is corrupted by a multiplicative field which is a bit more similar to real MR measurements. The following samples will continue to get more complex and therefore gradually approximate MR volumes of biological tissue. In that way the algorithms performance is evaluated and can be compared to other methods which process the same or similar samples.

3.1.2 Simulated MR volumes

MR phantom data is simulated using the BrainWeb online interface version 1.4 by Cocosco et al. 1997 [48]. Sample 3 is generated with the following parameters: slice thickness $1\,mm$, scan technique $SFLASH$, TR $18\,ms$, TE $10\,ms$, flip angle 30°, one echo, magnitude image, zero additional noise, bias field A and bias strength 100%. Preprocessing includes volume extraction (slice 59 to 125) and further focusing. A foreground mask is generated by region-growing-segmentation based on the non-local gradient. The resulting volume contains $180x216x67$ voxels.

Shading correction is done by the TGV^2-DCT algorithm with $\alpha_0 = 1$, $\alpha_1 = 2$ in 1300 iterations. To preserve small lobs in the image the denoising parameter is increased to $\lambda = 2$. Figure 3 shows slice 34 of this sample. The line profile plot of this sample demonstrates that the intensity of the white matter becomes more homogeneous. For quantitative evaluation Table 4 includes the CV, TV and entropy of the input and output volume. Entropy calculation is done for an intensity window from 0.1 to 1 with 128 histogram bins. All three metric values decrease and therefore confirm inhomogeneity reduction.

Sample 4 demonstrates simultaneous bias correction and denoising. The brain phantom is generated using the same parameters as for sample 3, except the noise level is 3%. To eliminate this additional noise the parameter λ is set to 1.2. Figure 4 qualitatively presents the results for this sample. The corrected image has reduced noise and inhomogeneity. Table 5 shows the metric values of the input and output image. The relative difference of the TV is greater than 45% and also the entropy is highly reduced. The CV value also decreases.

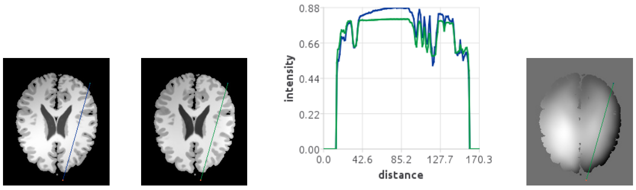


Figure 3: A representative slice of sample 3: algorithm input, output, line profile plot of input (blue) and output (green) and bias field.

Sample 3		
CV	TV	Entropy
1.207	0.552e-1	7.629
1.192	0.463e-1	7.045
1.267 %	16.104 %	7.657 %

Table 4: CV, TV and entropy of the input and output image of sample 3. The last row shows the relative difference of the corresponding metric value.

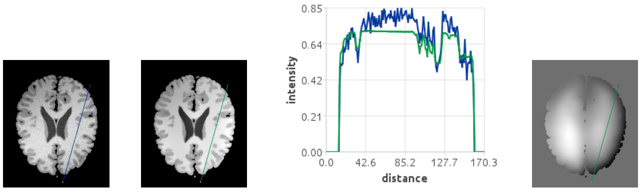


Figure 4: Sample 4: algorithm input, output, line profile plot of input (blue) and output (green) and bias field. This sample is equal to the one in Figure 3 above expect additional noise is added to the input image.

Sample 4		
CV	TV	Entropy
1.21	0.509-1	7.4
1.189	0.263-1	6.996
1.705%	48.1937%	5.46%

Table 5: CV, TV and entropy of the input and output image of sample 4. The last row shows the relative difference of the corresponding metric value.

In comparison to the previously shown samples the simulated MR volumes demonstrate the correction of 3D data sets. Because no ground truth data is available for these samples the metrics CV, TV and entropy are used to quantitatively evaluate the algorithm's performance. Again the second order algorithm is chosen because it produces promising results. It is therefore not necessary to increase the order in these cases.

3.1.3 Standard Test Images

The next sample in Figure 5 shows the Adelson-Checkboard [49]. This color image is transformed to the HSV space using the OpenCV framework [50] version 3.1. Only the value channel is processed, rescaled and written back to the color image file. Sample 5 includes two regions marked by an A and B. The line profile plot beneath this image shows that those contain the same intensity value. Again the second order algorithm TGV^2-DCT with $\lambda = 1$, $\alpha_0 = 1$, $\alpha_1 = 2$ is processed. Entropy calculation is based on KDE as described in Section 2.7.1. In the corresponding line profile plot contrast improvement between the region A and B is visible. The three metric values in Table 6 underpin a reduction of inhomogeneity.

A similar example is the Logvinenko-Illusion shown in Figure 6. Again the local contrast is enhanced by the extraction of the illumination. Table 6 includes the metric values for this sample.

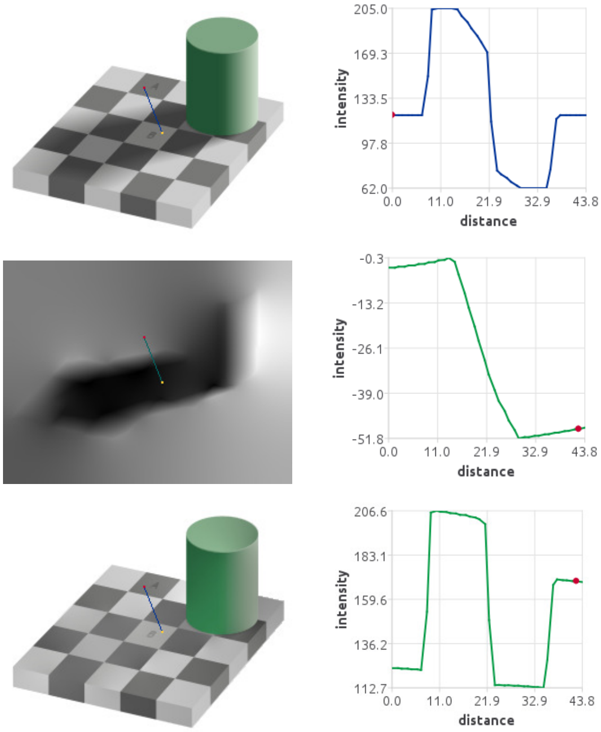


Figure 5: Image and line profile plot of sample 5. In the first row the denoised image is shown. The position of the line profile is visible in the image. It spreads from the red square (top left) to the yellow one (bottom right). The next row contains the extracted shading component and the third one the corrected data.

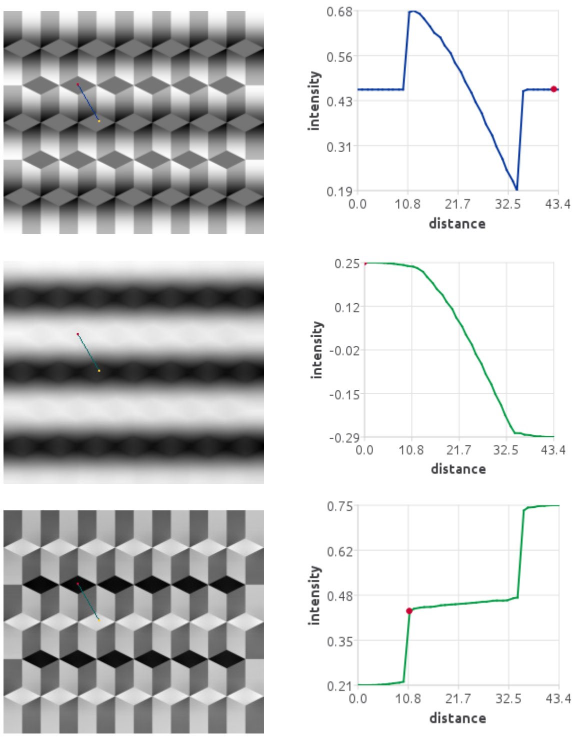


Figure 6: Image and line profile plot of sample 6. In the first row the denoised image is shown. The position of the line profile is visible in the image. It spreads from the red square (top left) to the yellow one (bottom right). The next row contains the extracted shading component and the third one the corrected data.

Sample 5			Sample 6		
CV	TV	Entropy	CV	TV	Entropy
23.8	4.53	6.5	17.7	10.3	7.22
17.9	4.43	6.69	8.76	7.01	6.93
25%	2.2%	2.85	51%	32%	4%

Table 6: CV, TV and entropy for sample 5 and 6. The first row contains the metric values of the denoised data. The second row shows the values for the shading corrected image and the third row the relative difference.

3.2 Natural Images

3.2.1 Standard Color Photographs

The next two test images are published in Fu et al. 2015 [13] and used to evaluate and compare several shading correction algorithms. In comparison to those methods the proposed algorithm does not include a gamma correction step. For the sample in Figure 7 the TGV^3-DCT algorithm is processed with $\lambda = 2$, $\alpha_0 = 1$, $\alpha_2 = 2$ and $\alpha_3 = 3$. The darker areas of the image become brighter and the overall shading is partly compensated. Figure 8 shows the next sample which is corrected by TGV^k-DCT of order 2 to 8 with the α value described in Section 2.2.5. With increasing order the gold globe becomes more and more homogeneous.

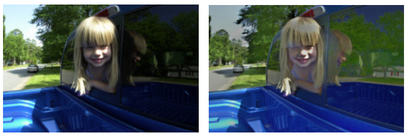


Figure 7: Input and output image of sample 7.

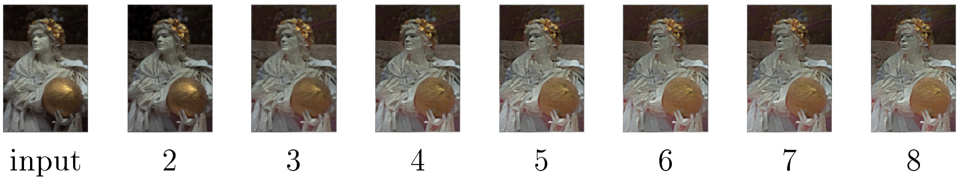


Figure 8: Input and output images of sample 8. The number beneath the output images is the TGV-DCT order.

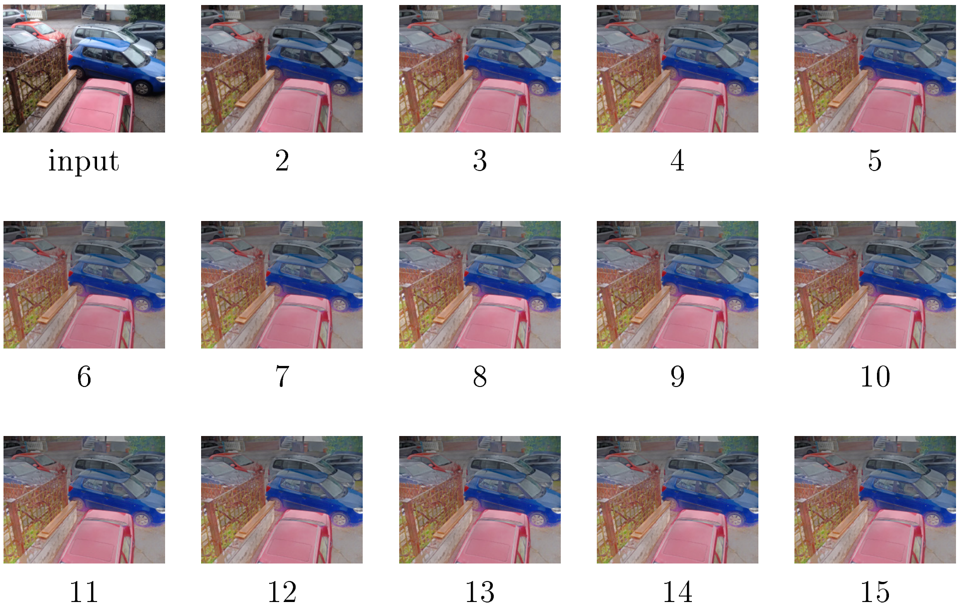
This section demonstrates the shading correction performance of the algorithm for color photographs. Higher order TGV-DCT is processed to extract piecewise constant regions of the image. The two samples show outdoor scenarios which are illuminated by sunlight. Light reflection and refraction occur at material boundaries. The shape of the shading field might be much more complicated than the bias field in MR measurements. The samples are included here to make it possible to qualitatively compare the algorithm to other variational methods found in literature.

Note that increasing the order does not lead to more parameters to adjust, because default values are used. Therefore shading extraction is simply done by choosing a value for the denoising parameter λ and the order of the TGV functional.

3.2.2 Color Photographs

The effect of increasing TGV$^{\mathrm{k}}$-DCT order is additionally analyzed in sample 9. The algorithm is processed with the default parameter values up to the order of 15. Results are presented in Figure 9. With increasing order the image becomes more and more piecewise constant as the algorithm fits a piecewise polynomial of order k-1 to the image and extracts only the constant information. Although the difference becomes smaller there are details which change. The car in the top-left corner becomes darker. This is the brightest region of the estimated shading image. With increasing order the algorithm fits an illumination field with pieces of smaller size.

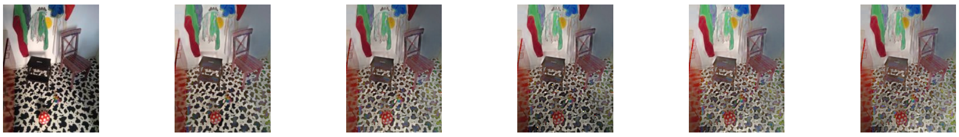
Figure 10 includes another sample which contains approximately piecewise constant objects. The black and white pattern of the floor, red and white ball and cotton tissues on the wall consist nearly of single colors in reality. With increasing order shadows on the wall and floor disappear. Contrast of the pattern on the floor decreases. The behaviour of the algorithm for increasing order is evaluated and described later in more detail.


input 2 3 4 5

6 7 8 9 10

11 12 13 14 15

Figure 9: Input and output images of sample 9. The number beneath the output images is the TGV-DCT order.



input 5 10 15 20 25

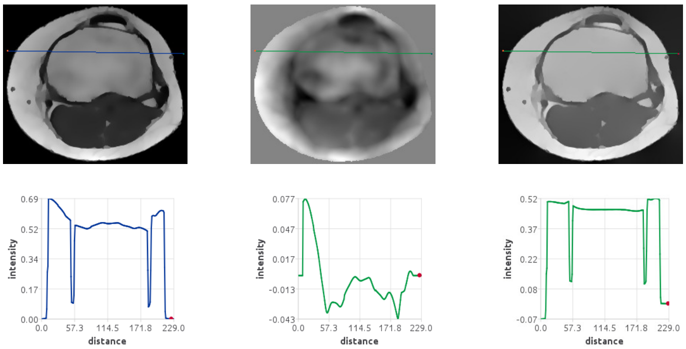
Figure 10: Input and output images of sample 10. The number beneath the output images is the TGV-DCT order.

The samples above are processed by the algorithm of order up to 25. These results further demonstrate the effect of increasing order for color images and are discussed later in Section 4.1.1. The proposed algorithm primarily aims to correction inhomogeneities of MR images. The following sections demonstrate the performance for such samples and include comparison to a reference method.

3.3 3D MR Measurements

MR measurements have been performed using a Skyra (Siemens AG Österreich, Wien) 3T device at the IMT Graz. The data include images of the right knee done by a knee coil. Sample 11 is a volume of $384x384x60$ voxels. In the physical space the voxel size is $0.49\,mm \cdot 0.49\,mm \cdot 1.4\,mm$. TR is set to $7790\,ms$ and TE to $10\,ms$. This turbo spin echo sequence generates ρ_H- and T2-weighted intensity values. To omit spatial details – and therefore make the line profile plots easier to compare – the volume is denoised in a preprocessing step. First TGV^2-L^1-Primal-Dual denoising with $\lambda = 1$ is processed. Additionally a Bilateral-Filter with a convolution kernel of 32^{iso} is applied. The parameter for the spatial distance is $\sigma_s = 25$ to remove details (wide neighbourhood) and the one for the intensity difference is $\sigma_i = 0.01$ (low gradients) to preserve edges. Figure 11 shows the results for this sample. Bias correction is done by TGV^2-DCT with $\lambda = 2$, $\alpha_0 = 1$, $\alpha_1 = 2$. The shading image partly contains the structure of the bone, muscle and surrounding tissue. In the line profile plot the resulting image does show piecewise constant regions. The metric values in Table 7 decrease due to the correction process.

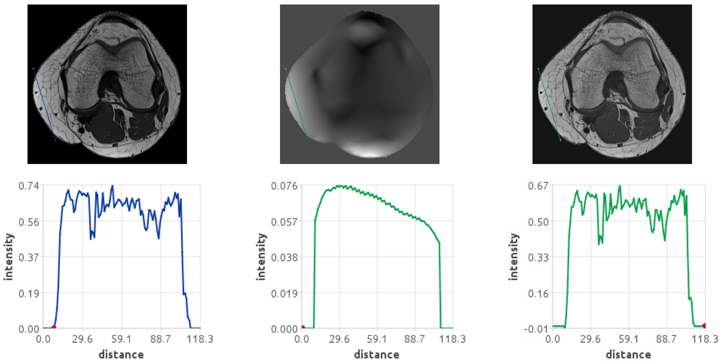
Sample 12 has equal dimensions and parameter values despite of the TR ($625\,ms$) and TE ($18\,ms$). This generates T1-weighted intensity values. No prior denoising is done for this sample. Figure 12 shows slice 27 of this volume and Table 8 the metric values.



Sample 11		
CV	TV	Entropy
0.233	0.299e-1	7.32
0.201	0.229e-1	7.05
13%	23%	3.7 %

Table 7: CV, TV and entropy for sample 11. Rows: denoised data, deshading and relative difference.

Figure 11: Results for sample 11. The arrangement of the images is the same as in Figure 5.



Sample 12		
CV	TV	Entropy
1.061	0.635e-1	7.394
1.01	0.633e-1	7.256
4.807%	0.315%	1.866%

Table 8: CV, TV and entropy for sample 12. Rows: denoised data, deshading and relative difference.

Figure 12: Results for sample 12. The arrangement of the images is the same as in Figure 5.

Figure 11 demonstrates inhomogeneity correction for previously denoised MR measurements. The resulting image contains piecewise constant regions for several tissues. In Figure 12 smaller structures are focused and preserved by the method. For these samples the second order algorithm is processed and improves the quality of the images as can be seen by the reduction of the metric values in Tables 7 and 8.

Next sample 13 contains only slice 11 of the second measurement volume. A foreground mask is generated by region-growing-segmentation and applied to the input and output image. TGV$^{\mathrm{k}}$-DCT of order 2 to 8 with the α value described in Section 2.2.5 is processed. Figure 13 shows that with increasing order the bias field includes more and more of the tissue structure, the histogram peak around 0.45 becomes narrower and the profile line straighter. The histogram peak around 0.11 and the profile line of the muscle are less modified.

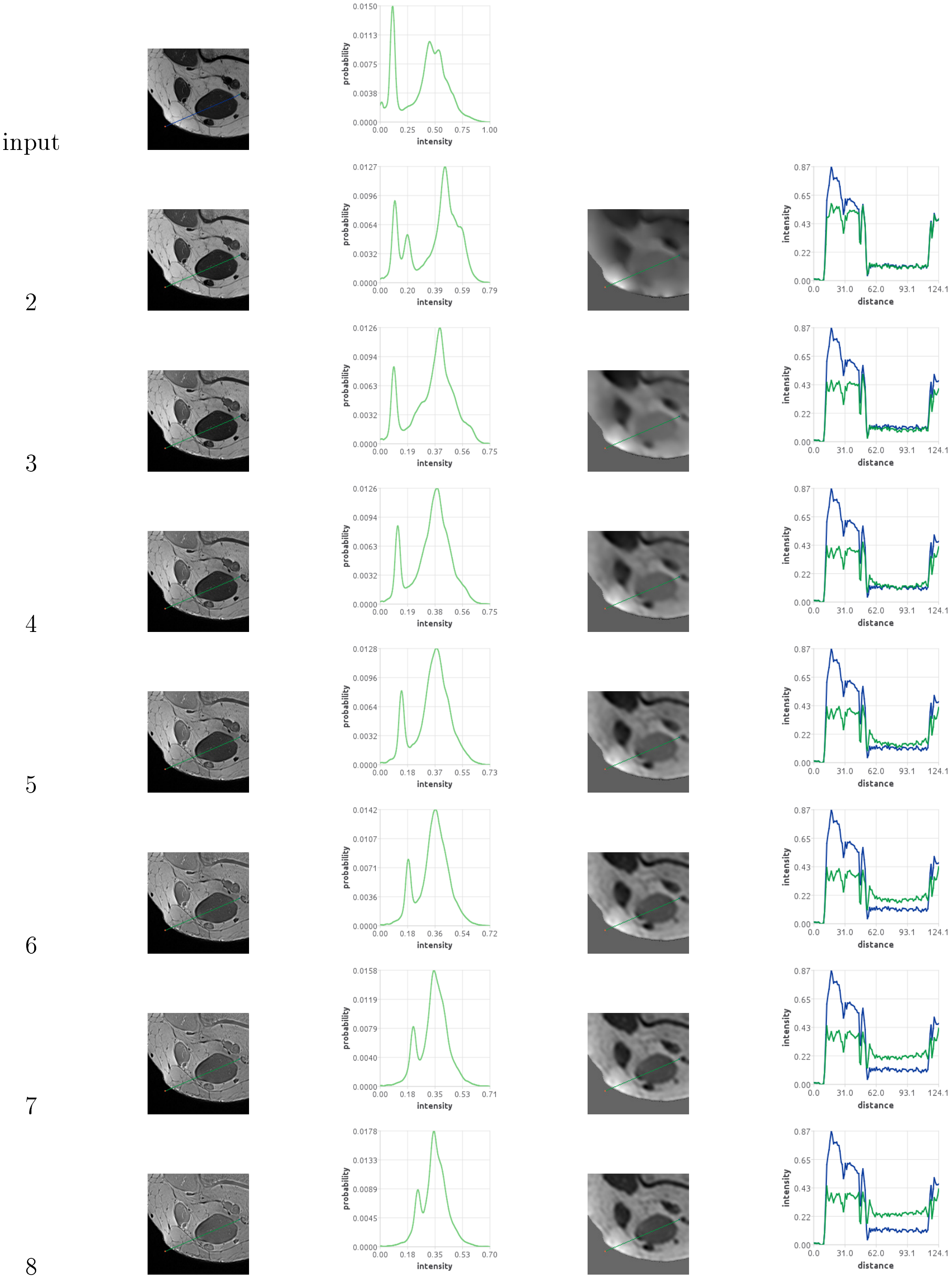


Figure 13: The first row includes the input image sample 13 and the corresponding histogram. The following rows include the TGV-DCT order, output image, histogram of the output image, bias field and line profile plot of the input image (blue) and output image (green).

3.3.1 Downsampled TGV-DCT Bias Correction

For the evaluation of the downsampled version of the TGV-DCT bias correction algorithm the following MR volume is processed. After focusing a region of interest sample 14 has the dimensions $200x186x65$. It shows a transversal magnitude image resulting from a 3D gradient inversion recovery sequence (MP-RAGE) with TE set to $2.66\,ms$, TR to $1430\,ms$ and TI to $900\,ms$. This results in T1 weighted intensity values.

This volume is processed by the downsampled TGV-DCT bias correction algorithm with multiple values for the downsampling factor $f_{downsampling}$ and the TGV order k. For better comparison the number of TGV iterations is set to $1e4$ for all combinations of the input parameters. Negative intensity values of the resulting volume are clamped to a value of zero. The absolute computation time is compared between several downsampling factors. Computation is done on a computer with a Geforce GTX 1070 (Nvidia Corporation, Santa Clara, USA) dedicated graphic card with $8\,GB$ memory, a quadcore i5-4690U (Intel Corporation, Santa Clara, USA) with $3.5\,GHz$ and $16\,GB$ working memory. During processing nearly $100\,\%$ of the processors capacity is used.

The masked input volume has the following metric values: CV is 0.987, TV is 0.068 and entropy (calculated with 256 intensity bins and a minimum threshold of 0.01) is 6.963. Figure 14 shows slice 33 of the input and shading-corrected volumes. A set of three values for k and $f_{downsampling}$ is processed which results in nine estimated bias-free volumes (Figure 14) and bias fields (Figure 15). The bias fields shown include more details with increasing order and less details with decreasing downsampling factor $f_{downsampling}$.

Downsampling reduces the computation time and GPU memory usage which is shown in Table 9. As described above in Section 2.2.5 the memory usage increases with increasing TGV-DCT order. For the processed sample 14 increasing the order by one requires $288\,MB$, $38\,MB$ and $5\,MB$ additional storage for this image size and three downsampling factors respectively. Downsampling the volume dimensions by $1/4$ decreases the needed computation time by about $95\,\%$ for all three TGV-DCT orders. Also the required memory is more than $90\,\%$ less.

Table 10 includes performance metric values for sample 14. All parameter sets show an improvement of the image quality except for TGV^2-DCT with $f_{downsampling}$ set to $1/4$. For this setting the TV increases in comparison to the masked input volume. The quantitatively best result is obtained by TGV^4-DCT without downsampling. In that case the CV is reduced by about $13\,\%$, the TV by $2.5\,\%$ and the entropy by $11\,\%$. These resulting volumes are calculated in about $15\,min$ and require $872\,MB$ GPU memory (Table 9). The fastest version is second order with $1/4$ downsampling factor which requires about $15\,s$.

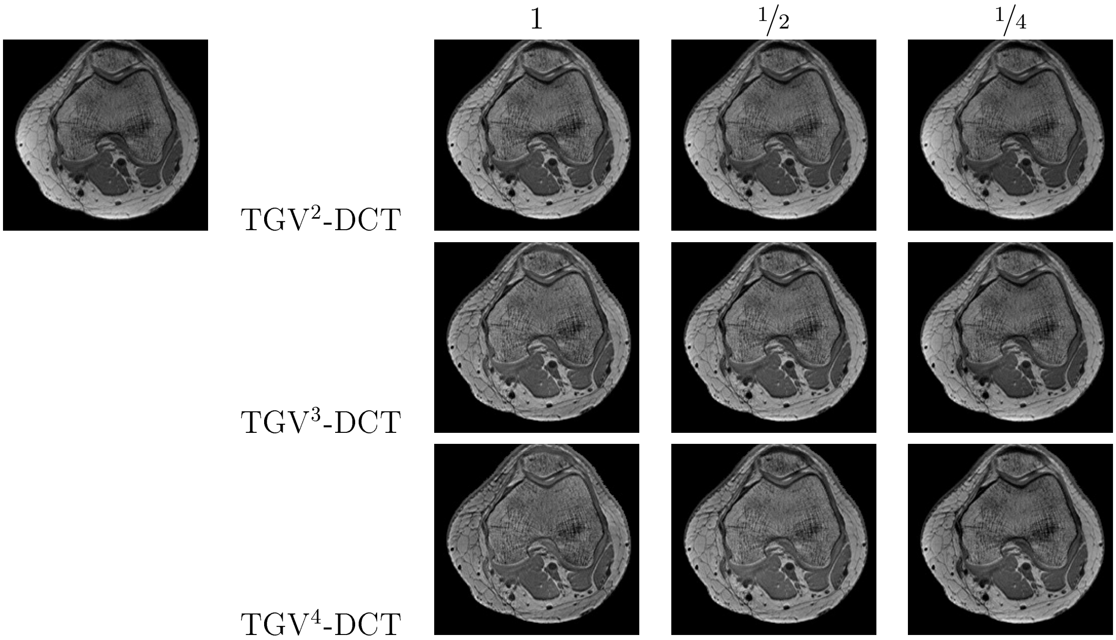


Figure 14: Slice 33 of the estimated bias-free volumes for sample 14. The top-left image shows the input. Rows include the estimated bias-free image for TGV orders of 2, 3 and 4. In the columns the $f_{downsampling}$ parameter is set to 1, 1/2 and 1/4.

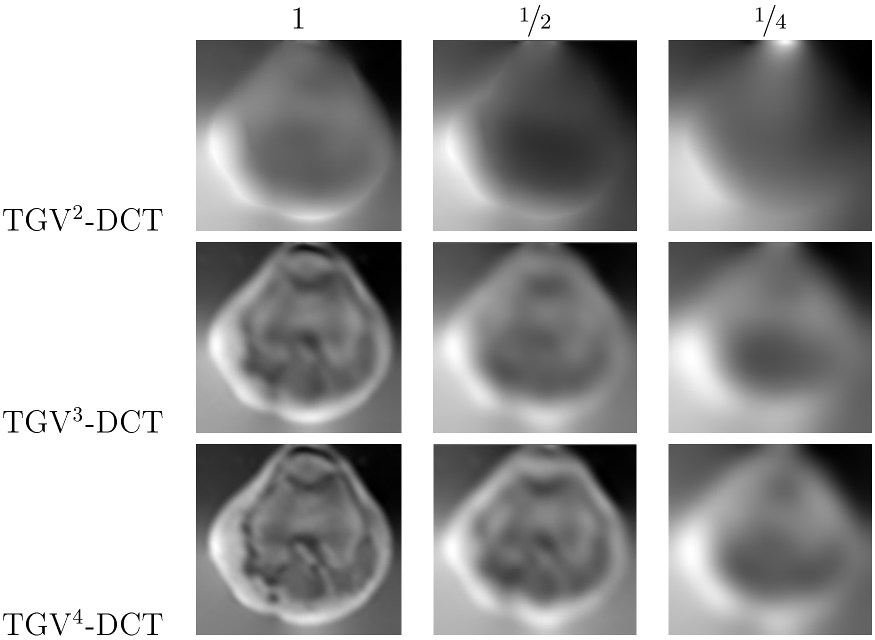


Figure 15: Slice 33 of the estimated bias volumes for sample 14. Rows include the estimated bias slice for TGV-DCT orders of 2, 3 and 4. In the columns the $f_{downsampling}$ parameter is set to 1, 1/2 and 1/4.

TGV-DCT order	$f_{downsampling}$	computation time		GPU memory usage	
[1]	[1]	[s]	[%]	[MB]	[%]
2	1	334.57	100	317	100
	$^1/_2$	43.297	87.059	61	80.757
	$^1/_4$	**14.898**	95.547	**27**	91.483
3	1	584.166	100	595	100
	$^1/_2$	75.1473	87.136	99	83.361
	$^1/_4$	25.2603	95.676	32	94.622
4	1	833.148	100	872	100
	$^1/_2$	106.413	87.228	136	84.404
	$^1/_4$	35.715	95.713	37	95.757

Table 9: Computation time in seconds and GPU memory usage in megabytes for the processed sample 14. The fourth columns shows the relative difference of the computation time of downsampling and the last column the relative difference of GPU memory usage. The bold values mark the optimum.

TGV-DCT order	$f_{downsampling}$	CV		TV		Entropy	
[1]	[1]	[1]	[%]	[1]	[%]	[1]	[%]
2	1	0.915	7.295	0.675e-1	0.735	6.683	4.021
	$^1/_2$	0.936	5.167	0.678e-1	0.294	6.819	2.068
	$^1/_4$	0.978	0.912	0.683e-1	- 0.441	6.942	0.302
3	1	0.869	11.955	0.667e-1	1.912	6.351	8.789
	$^1/_2$	0.889	9.929	0.673e-1	1.029	6.55	5.931
	$^1/_4$	0.924	6.383	0.677e-1	0.441	6.735	3.274
4	1	**0.858**	13.07	**0.663e-1**	2.5	**6.178**	11.274
	$^1/_2$	0.875	11.348	0.671e-1	1.324	6.429	7.669
	$^1/_4$	0.902	8.612	0.675e-1	0.735	6.624	4.869

Table 10: Metric values for the processed sample 14. The reference for the relative values is the masked input volume. The bold values mark the minimum.

3.3.2 Masked TGV-DCT Bias Correction

The masked version of the TGV-DCT bias correction algorithm – described in Section 2.5 – also highly reduces the computation time. It is compared to the previously processed downsampled algorithm by using the same sample volume. Table 11 lists the computation time and GPU memory required. In all processed cases the masked algorithm is much faster (72 to 80 % relative difference). It is also shown that the required GPU memory increases for the three TGV-DCT orders and downsampling factors. The relative difference of the required memory is between 5 and 15 %.

After multiplying the input image with the mask, the outcomes for the following three versions of the algorithm are identical. First the algorithm which calculates all operations on all voxels is used. Second the masked version is processed with a mask including all pixels. Third a mask – generated by region growing segmentation – is taken to efficiently perform the necessary operations on the foreground pixels only.

TGV-DCT order	$f_{downsampling}$	computation time		GPU memory usage	
[1]	[1]	[s]	[%]	[MB]	[%]
2	1	75.284	77.498	360	13.565
	1/2	11.789	72.771	70	**14.754**
	1/4	3.452	76.826	29	7.407
3	1	125.97	78.436	638	7.227
	1/2	19.453	74.114	108	9.091
	1/4	5.231	79.291	34	6.25
4	1	176.433	78.823	915	**4.931**
	1/2	27.117	74.517	145	6.618
	1/4	7.022	80.337	39	5.405

Table 11: Computation time in seconds and GPU memory usage in megabytes for the processed sample 14 using the masked version of the algorithm. The relative difference in comparison to the downsampled algorithm (Table 9) is shown in column four and six. The values in bold represent the optima.

3.3.3 Comparison with the reference method

In this section the method is compared to the N4ITK algorithm by Tustinson et al. 2010 [16]. Data processing is done by the ANTs [51] binary (version 2.1) using the default parameter values: number-of-histogram-bins 200, wiener-filter-noise 0.1e-1, bias-field-fwhm 0.15, maximum-number-of-iterations 50, spline-order 3, number-of-fitting-levels 4. For sample 15 the convergence measure (coefficient of variation) at the final iteration is 0.285e-3. Figure 16 shows the resulting corrected MR slice and bias field. The histogram of the corrected image contains a single narrow peak. For better comparison the additive bias field of the proposed method is converted into a multiplicative one. This is done by thresholding and conditional pixel-wise division (omitting zero division) of the input and the output image. In Figure 17 the results of the proposed method for the same input image and mask are shown. The first row contains a line profile plot of the bias field of both methods. On the one hand the histogram of the corrected image contains a less narrow peak but on the other the line profile plot shows a straighter intensity line compared to the one of the N4ITK result. Table 12 shows slightly lower entropy and CV values and a higher TV value for the reference method.

A second MR sample is used to compare the two methods. This one is not previously denoised and contains 7 equal phantom cylinders which are a hint for the bias field. Both methods are processed without a mask. N4ITK converges with a CV of 0.653e-3. Figure 18 contains the results for the N4ITK method. A second peak for the muscle is visible in the spectrum and is preserved in both methods. In Figure 19 the line profile plot of the two bias fields show that N4ITK can produce fast varying bias fields too. The metric values of Table 13 are similar to those of the previous sample in Table 12 .

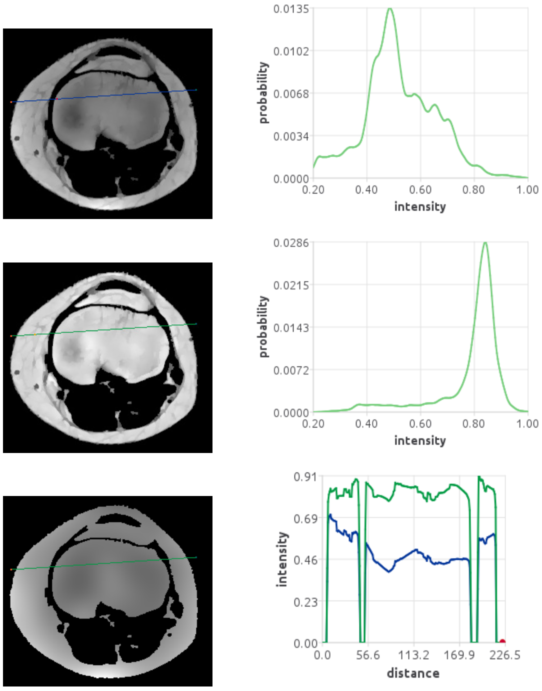


Figure 16: Results of the N4ITK-method for sample 15: First row: input image and the corresponding histogram. Second row: the shading corrected image and the histogram. Third row: the multiplicative bias field and a line profile plot of the input and the corrected image.

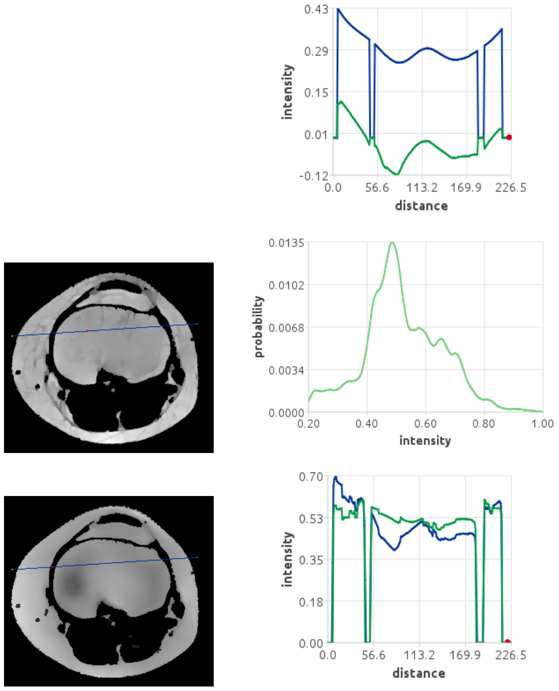


Figure 17: Results of the proposed TGV-DCT-method for sample 15: First row: line profile plot of the bias field of both methods. Second row: the shading corrected image and the histogram. Third row: the multiplicative bias field and a line profile plot of the input and the corrected image.

	CV	TV	Entropy
N4ITK	1.11	0.443e-1	5.96
TGV-DCT	1.14	0.37e-1	6.04

Table 12: CV, TV and entropy of the corrected sample 15 for the methods N4ITK and TGV-DCT.

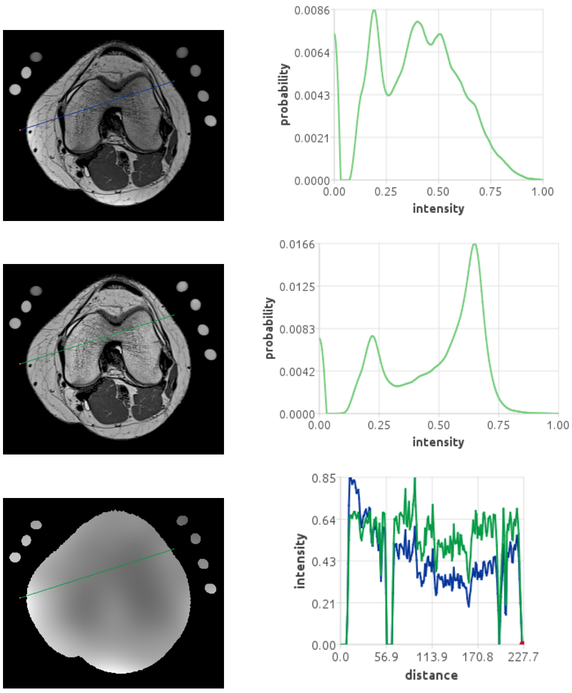


Figure 18: Results of the N4ITK-method for sample 16: First row: input image and the corresponding histogram. Second row: the shading corrected image and the histogram. Third row: the multiplicative bias field and a line profile plot of the input and the corrected image.

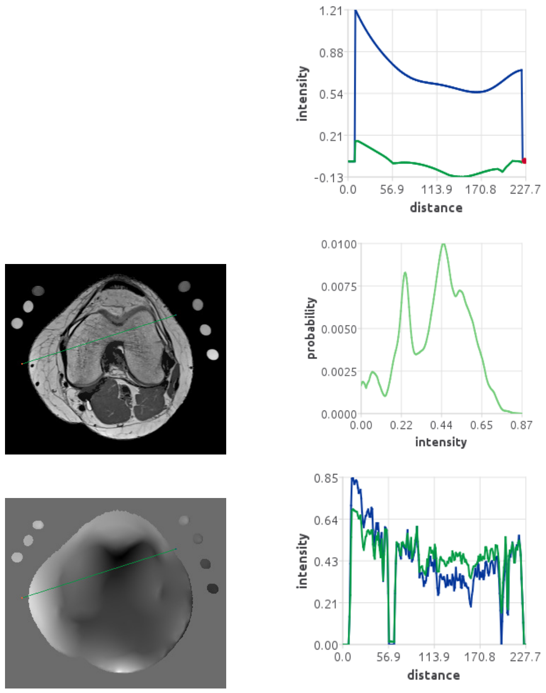


Figure 19: Results of the proposed TGV-DCT-method for sample 16: First row: line profile plot of the bias field of both methods. Second row: the shading corrected image and the histogram. Third row: the multiplicative bias field and a line profile plot of the input and the corrected image.

	CV	TV	Entropy
N4ITK	1.21	0.582e-1	6.33
TGV-DCT	1.25	0.509e-1	6.61

Table 13: CV, TV and entropy of the corrected sample 16 for the methods N4ITK and TGV-DCT.

The previous sample 16 is also processed by TGV-DCT of order 3 to 5. Figure 20 shows the resulting image, histogram, line profile plot and bias field. Table 14 includes the decreasing metric values for these results. The histogram peak of the muscle merges with the one for the other foreground pixels and the bias component includes more and more details. So contrast between the muscle and the surrounding tissue decreases.

The bias of the seven cylinders is partly removed by the processed methods. Because these regions are separated the PDF domain based method N4ITK performs much better than the others which corrected the image in the spatial domain. The line profile plot of Figure 20 shows that TGV-DCT removes the bias inside each cylinder, but it does not estimate a meaningful bias field in the background. Nevertheless it produces similar results than N4ITK for the biological tissue.

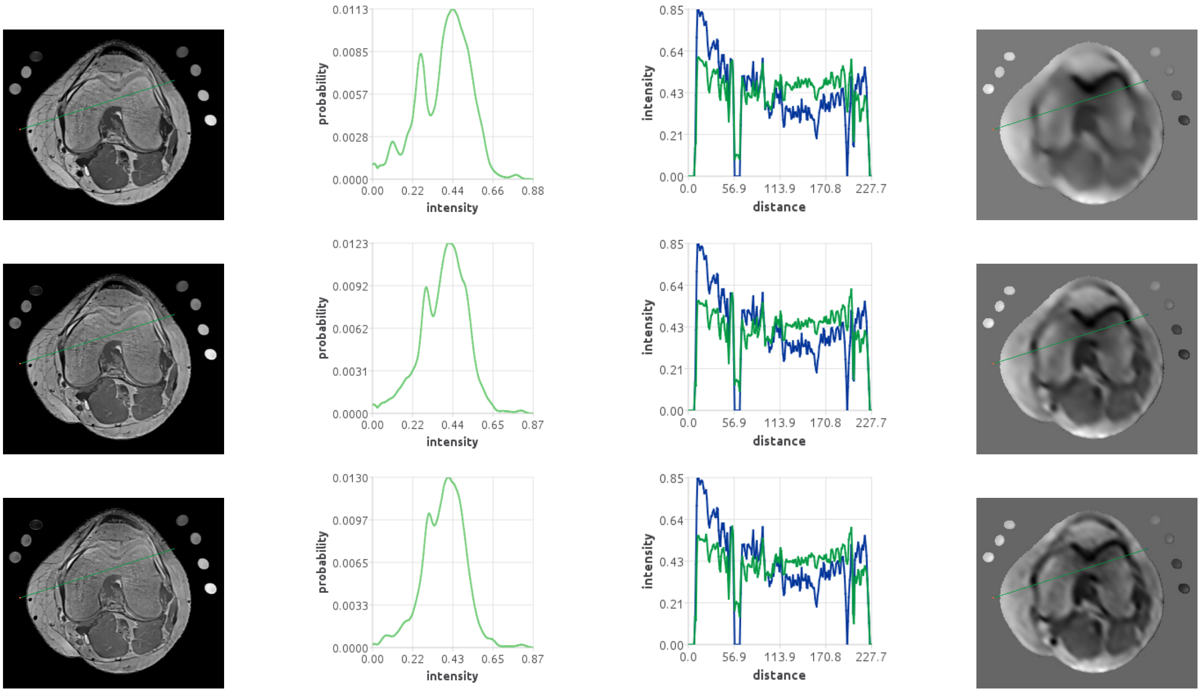


Figure 20: Results of the proposed TGV-DCT-method for sample 16: resulting image, histogram, line profile plot of input and output and bias image. The rows contain the results for TGV-DCT with order 3,4 and 5.

TGV-DCT Order	CV	TV	Entropy
3	1.157	0.399e-1	7.188
4	1.15	0.389e-1	7.025
5	1.14	0.383e-1	6.949

Table 14: CV, TV and entropy of the corrected sample 16 for the method TGV-DCT with increasing order.

3.3.4 Estimation with Denoising and Correction based on the Input Image

In the following experiment the downsampled version of the TGV-DCT bias correction algorithm is used to estimate the bias field with simultaneous denoising and correct the input image without denoising. This is achieved by choosing a low value for the denoising parameter $\lambda = 1$ and estimating a vector field v with this TGV-L^1 parameter. As described above in Section 2.4 the estimated bias is subtracted from the input image if downsampling is applied. Figure 21 shows the estimated bias-free image, probability function and a representative line profile plot. Again sample 16 is processed using N4ITK and several versions of the proposed algorithm. N4 generates two main peaks in the intensity spectrum and a straight line in the profile plot. The second order TGV-DCT algorithm generates a similar result. It differs in the width and position of the two intensity classes. The proposed method does not alter the position of the peaks. In other words the absolute intensity values are preserved. Third order TGV-DCT introduces additional inhomogeneities on the left fat, the bottom bone and top right fat tissue regions. Nevertheless the intensity peak is narrower than the one of the result of the second order algorithm. Downsampling the input by one half produces a qualitatively better result with two (symmetric) Gaussian distributions in the estimated PDF.

The last two columns of Figure 21 show the estimated bias fields for these calculations. N4 works in the logarithm domain and therefore produces a multiplicative bias field. For better comparison the input image is subtracted by the resulting estimate of the bias-free image. And the additive bias fields estimated by the proposed algorithm are transformed to multiplicative ones. This is done by dividing the input image with the resulting bias free image. To omit division by a value near zero the bias-free estimate is first thresholded and division is only done if the divisor is not equal to zero. Although the multiplicative bias field of N4 and TGV^2-DCT look similar (for example the gradient at the green line) they are not easy to compare. In general all output images are rescaled to contain black pixels for the minimum intensity and white ones for the maxima. The calculated multiplicative bias includes a few bright pixels. One can also see black pixels for regions where no bias field is estimated. N4 interpolates the bias in those regions using B-splines. The bias fields show more details for increasing the TGV-DCT order and less for a decreasing downsampling factor.

Table 15 contains metric values for this experiment. In fact N4 generates the worst CV and TV but a good entropy value. Quantitatively the third order method without downsampling produces the best result. In the following Section 3.3.5 an additional metric will be introduced to count for the local contrast between several regions. The estimated multiplicative bias field contains low values in the center of the image and the factors increase with the distance from the center. Maximum bias is visible at the top and right edge.

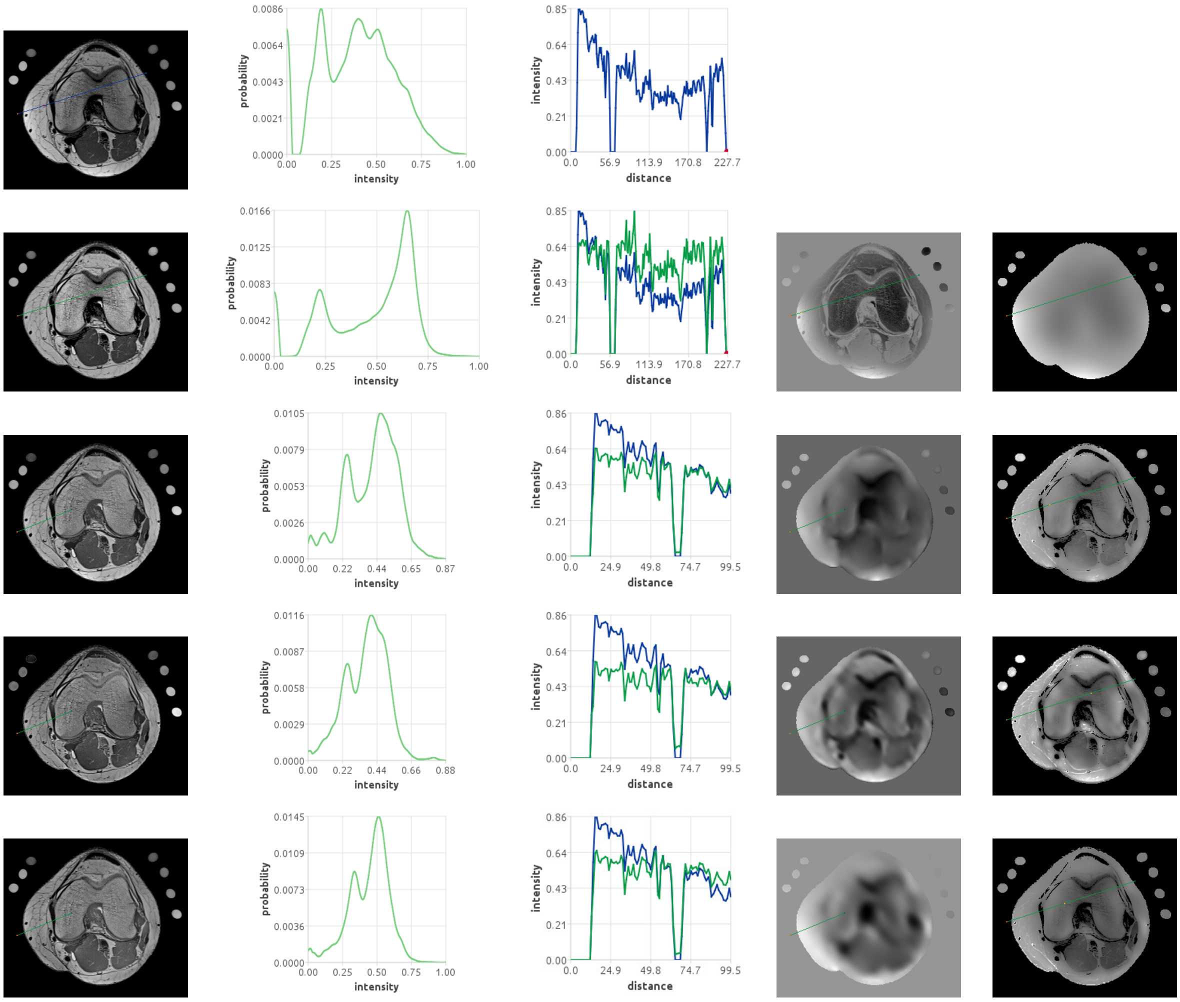


Figure 21: The resulting estimated bias-free image, the histogram and a representative line profile plot of the input (blue) and output(green) intensities, the additive and multiplicative bias field for the reference method N4ITK and several versions of the proposed TGV-DCT algorithm. The position of the line is additionally painted to the image. The rows contain the result of the algorithms and parameters listed in Table 15.

	CV	TV	Entropy
N4ITK	1.144	0.588e-1	7.183
TGV^2-DCT	1.139	0.524e-1	7.237
TGV^3-DCT	1.101	0.512e-1	7.03
TGV^3-DCT $f_{downsampling} = {}^1\!/_2$	1.093	0.534e-1	7.242

Table 15: CV, TV and entropy of the corrected sample 15 for the method TGV-DCT with increasing order.

3.3.5 Simultaneous Denoising and Bias Correction

First the authors of the N4ITK algorithm refer to the proposal of the original N3 by Sled et al. 1998 [15] about the intensity spectrum. Because these algorithms work in the logarithm domain, zero and very small intensity values are not recommended. Therefore the input image is not rescaled and the DICOM integer values are processed. Because no prior denoising by a Bilateral Filter is performed, the parameters is not influenced.

In this the TGV-DCT algorithm will perform simultaneous denoising. This means no downsampling and a value for the denoising parameter λ smaller than 2. Only the combined effect of denoising and deshading are evaluated by the inspection of the resulting estimated bias-free image.

Figure 22 shows sample 17 which is equal to sample 16 except the oil cylinders are removed. This figure includes the estimated bias-free image, the PDF, a representative line profile plot and the additive and multiplicative bias. The line profile plot spawns from the muscle through the bone to the fat tissue. PDF estimation is done by KDE with 256 intensity values, a bandwidth of 40 and a window from 150 (to suppress low background pixels) to the maximum intensity value of 2135.

The processed algorithms and parameters are shown in Table 16. It includes the chosen value for λ and the number of iterations processed. Furthermore the required computation time and metric values are listed. In addition to the three metric values used above the local contrast (described in Section 2.7.5) is evaluated. This is done by approximating the intensity difference from muscle to bone and from bone to fat based on the line profile plot. In the input image the difference from the last pixel of the muscle to the first pixel of the bone is 400 for example.

The input image contains a slow bias which increases the intensity values from top-left to bottom-left at the position of the profile line. This widens the PDF peak for the bone and fat tissue. Nevertheless, there are still two separate peaks visible. As mentioned above N4 produces two very narrow peaks and a smooth and slow varying multiplicative bias field. It improves the contrast between muscle and the other pixel values to 700 but there is no contrast between bone and fat tissue anymore. The maximum values of the bias field are visible at the top and right edge. This field is interpolated by B-splines, to allow values in the background to be estimated. As mentioned above in Section 3.3.3 N4ITK processes 200 iterations by default. Table 16 lists the processed algorithms, parameters, required computation time and metric values. N4 requires about $3.8\,s$ and generates the maximum contrast for the muscle tissue, but increases the TV of the image.

Next, second order TGV-DCT with $\lambda = 1.75$ is processed until convergence. It reduces the CV, TV and entropy value and preserves the original contrast between muscle and bone tissue. The additive estimated bias field in Figure 22 also shows maximum bias at the top and right edge. The PDF contains 2 main peaks, but there are two additional distributions very close to the second peak. These represent the intensities of the bone, fat and details in those two tissue classes. Simultaneous denoising additionally reduces the metric values, the straightness of the intensity profile line and allows to select a scale of interest. Therefore unwanted details which are assumed to be noise are removed from the image.

The next algorithm processed is third order TGV-DCT with the previous denoising parameter. Only 700 iterations and less than $0.2\,s$ are required for the calculation. The bias in the bone tissue is greater than the previous one. This generates brighter pixel values in the resulting bias-free estimate. In the PDF there are three main peaks visible.

Using the previous denoising parameter value another calculation performs denoising and simultaneous bias correction by only 1000 iterations of TGV^{15}-DCT. This requires less than $1.5\,s$ computation time and generates the best CV and entropy values of this experiment. There are three very narrow peaks visible in the PDF and much more details in the estimated bias field.

The next three calculations perform intensive denoising by setting λ to 1. Again the maximum step size parameter α contains the values calculated by the series defined in Section 2.2.5. These results contain very low TV values and require less iterations. Second order TGV-DCT runs 1000 iterations in $0.2\,s$. The effect of denoising is clearly visible in the line profile plot. The third order algorithm produces the best TV metric value and also best contrast between bone and fat. The PDF shows three very narrow peaks. The last computation demonstrates intensive denoising in just 600 iterations by TGV^{15}-DCT. All metric values are good in comparison to the another results. This parameters set runs faster than the reference method and produces better results. The resulting bias-free estimate shows three homogeneous regions and the PDF three very narrow and symmetric peaks. There are many details contained in the estimated bias which is removed from the denoised image.

These calculations show that the proposed algorithm can perform simultaneous bias correction and denoising. This additionally reduces the metric values and therefore improves the quality of the results. Furthermore the absolute intensity values and contrast between spatially separated regions is preserved.

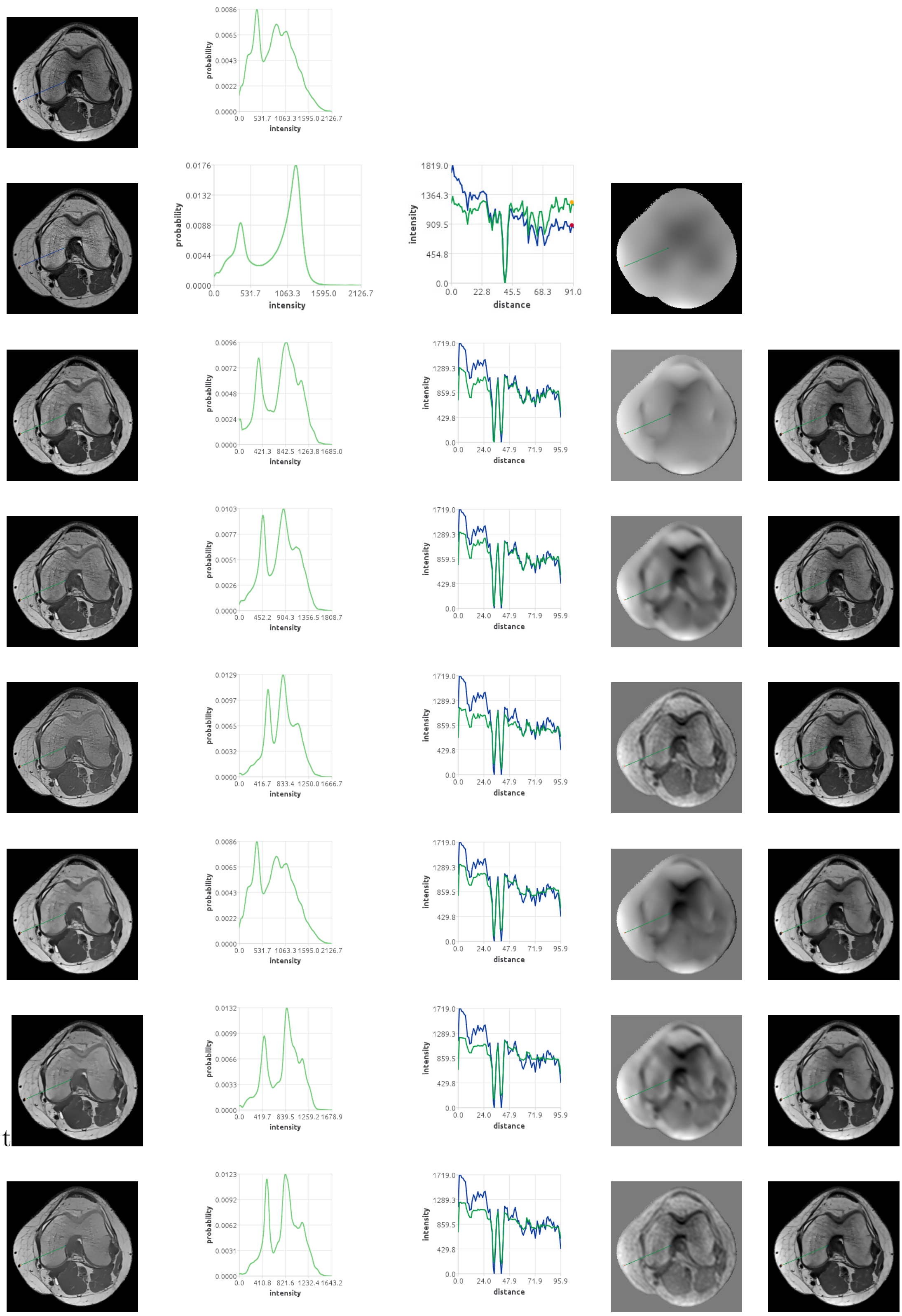


Figure 22: In the first row the input image and an estimate of the PDF is shown. The following rows contain the resulting estimated bias-free image, the corresponding PDF, a line profile plot of the input and corrected image, the bias and denoised image for sample 17. Table 16 lists the algorithms and parameter values used.

Algorithm	λ	#iter	Required Time [s]	CV	TV	Entropy	Contrast M-B	Contrast B-F
Input				1.046	129.587	7.612	400	200
N4ITK		2e2	3.873	0.992	136.725	7.111	**700**	0
TGV^2-DCT	1.75	5e4	9.65	0.961	90.158	7.254	400	100
TGV^3-DCT	1.75	7e2	0.229	0.96	100.342	7.23	300	100
TGV^{15}-DCT	1.75	1e3	1.463	**0.923**	86.139	**6.962**	300	120
TGV^2-DCT	1	1e3	0.2	0.962	70.187	7.2	200	220
TGV^3-DCT	1	1e3	0.303	0.942	**66.834**	7.074	300	**250**
TGV^{15}-DCT	1	6e2	0.91	0.934	67.033	7.033	200	150

Table 16: Metric values for sample 17. The first three columns contain input values: algorithm, parameter and number of iterations processed. Column four shows the required computation time. The following columns present the metric values: CV, TV, Entropy and Contrast between two different tissue edges. These are muscle-bone (M-B) and bone-fat (B-F).

3.3.6 Influence of the Parameters on the Inhomogeneity Shape

In principle the shape of the bias field depends on the input image and the chosen parameters. As mentioned above in Section 3.3.1 the downsampling factor $f_{downsampling}$ can also be used to focus on low frequency information. In this section only the influence of the maximum step size parameter for the second order gradient α_1 in Equation 1 is investigated.

In the following experiment the second order TGV-DCT inhomogeneity correction algorithm is processed with $\lambda = 1$, $\alpha_0 = 1$ and multiple values for the analyzed parameter α_1. No denoising is performed by subtracting the estimated bias from the input image. Figure 23 shows the resulting bias free estimate, the estimated bias and line profile plots for a set of parameter values.

A similar effect might be reached by choosing smaller values for λ and λ_0. But for parameter estimation it is much simpler to rescale the intensities and fix α_0 to 1. In Section 4.3 a recommendation for a strategy to chose a value for the parameter vector α is given.

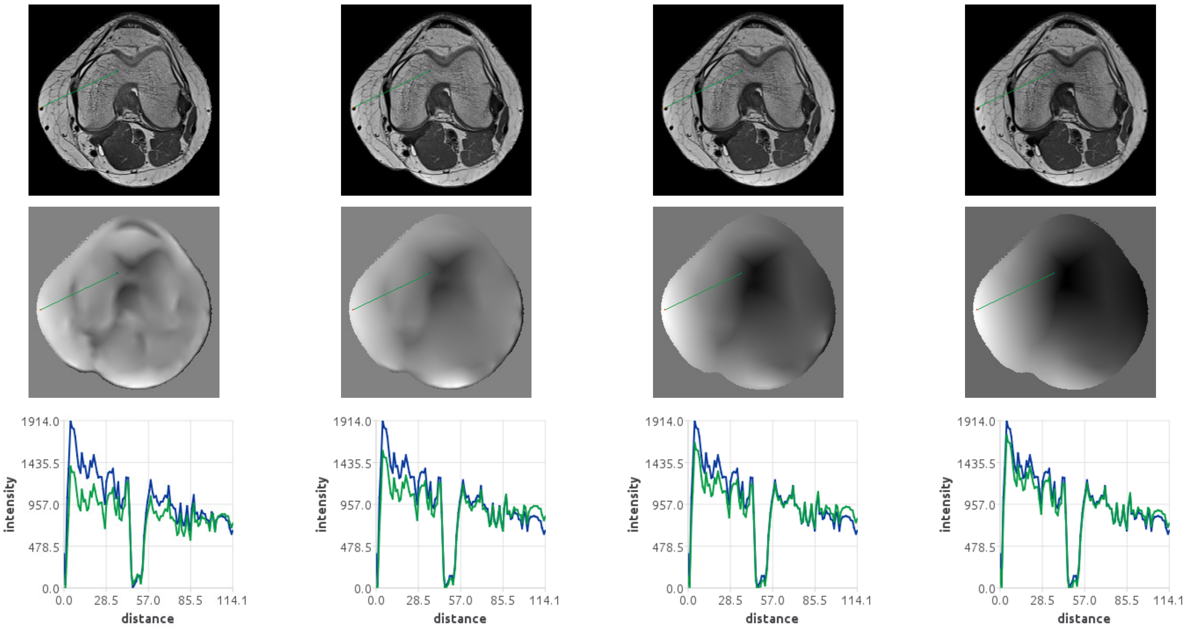


Figure 23: Results of the proposed TGV-DCT-method for sample 16: In each column resulting bias free estimate, the bias, and the line profile plots of the input image and bias free estimate are shown. The results of columns are calculated with $\alpha_1 \in \{2, 3, 4, 5\}$.

3.3.7 Inversion of the Laplace Operator

The two alternative methods for the inversion of the Laplace Operator are compared as follows. Sample 12 – which is shown in Figure 12 – is processed by TGV^2-L^1 with the default parameters. The first input image has the dimensions 251x251 and intensity range from 0 to 1. The TGV-L^1 algorithm generates a denoised image and additionally a vector field v. The divergence of this vector field is the input for the two extraction methods listed in Algorithm 2 and Algorithm 3. Both methods are executed 20 times. The CG-algorithm converges with an error threshold of $\epsilon = 10^{-3}$ after 207 iterations. Additionally sample 13 in Figure 13 is processed. This image has the dimensions 156x156. The parameters for CG are the same and convergence is reached after 342 iterations. Table 17 includes the required computation time and calculated metric values for the estimated bias-free images.

	Method	Duration	CV	TV	Entropy
Sample 12	DCT	$44.8\,ms \pm 4.238\,ms$	0.578	0.534e-1	7.284
	CG	$358.35\,ms \pm 41.473 ms$	0.59	0.54e-1	7.306
Sample 13	DCT	$160.25\,ms \pm 20.760\,ms$	0.632	0.554e-1	7.012
	CG	$1432\,ms \pm 151.344\,ms$	0.638	0.554e-1	7.018

Table 17: CV, TV and entropy of the two alternative bias field extraction methods for sample 12 and 13.

4 Discussion

4.1 Interpretation

4.1.1 Processed Samples

For the simulated ground truth data in Section 3.1.1 the TGV^2-DCT bias correction algorithm successfully restores the piecewise constant regions of sample 1 and 2. These samples really are piecewise constant compositions without any kind of noise. The calculated bias field contains only linear and quadratic terms with respect to the spatial dimension. The RMSE for the multiplicative field in sample 2 is higher than the value for sample 1. Nevertheless the algorithm improves the homogeneity of the image shown in Figure 2.

The next step is the simulation and processing of MR volumes in Section 3.1.2. For sample 3 the algorithm converges fast and strongly reduces inhomogeneity. Even for the noisy MR phantom (sample 4) bias correction is successful and additionally performs denoising. The huge reduction of the TV and entropy in Table 5 confirms this behavior. The CV metric is less influenced by this zero-mean noise.

The color images Adelson Checkerboard (sample 5) and Logvinenko Illusion (sample 6) are also used for evaluation in the work of Liang and Zhang 2015 [11]. The methods discussed in their work are qualitatively compared to the results in Figure 5 and Figure 6. The three methods do not estimate the shading on the cylinder as well as the proposed TGV-DCT method does. In sample 6 the piecewise constant information is better preserved in the result.

The standard test photograph sample 7 is compared to 4 different methods in Fu et al. 2015 [13] and sample 8 is included in Kimmel et al. 2003 [6]. Because no ground truth and metric values are available the results are again only qualitatively compared. The colors of the shaded regions of sample 7 in Figure 7 are better restored. For sample 8 it is difficult to define the desired goal. The statue basically consists of white stone and golden metal. An objective comparison is not done here. Increasing the TGV-DCT order does reduce inhomogeneity as can be seen in Figures 8 to 10.

Next the algorithm is processed using a measured MR volume. To analyze only the bias correction prior denoising is done. Figure 11 representatively shows that the resulting image contains piecewise constant regions and the shading component piecewise higher order intensity shapes. Also without prior denoising (sample 12) the bias field extracted by TGV^2-DCT improves the image quality.

Figure 13 shows the effect of increasing the TGV-DCT order for a measured MR slice. Up to order 9 the algorithm performs better which is visible in the histogram of the foreground pixels and line profile plot. Prior foreground masking and intensity rescaling drastically influences the results. Figure 20 includes the results of the TGV-DCT higher order algorithm which may not be near the desired goal. In the histogram the two intensity classes merge together.

In Section 3.3.6 the influence of the parameter α_1 is under investigation. With increasing value the bias becomes more smooth. The third term of Equation 1 is weighted stronger which minimizes the second derivative faster. There are less sharp edges in the bias field, and it is less similar to the gradient of the denoised image.

4.1.2 Downsampled TGV-DCT Bias Correction

Downsampling generally decreases the required computation time and GPU memory (Table 9). Therefore the algorithm can be processed much faster and higher order TGV-DCT becomes feasible for a specific hardware setting. Figure 15 shows that downsampling also highly influences the shape of the resulting bias field. As described above in Section 2.4 the low-pass character of the B-spline interpolation of smaller volumes produces slower and smoother fields. The metric values decrease with increasing TGV-DCT order from 2 to 4 (Table 10). On the one hand the results quantitatively get worse with increasing downsampling but on the other $f_{downsampling}$ is an additional parameter to control the smoothness. Therefore the results may qualitatively get closer to the desired goal by specifically adjusting this parameter.

4.1.3 Masked TGV-DCT Bias Correction

Foreground masking does not influence the resulting volumes as shown in Section 3.3.2. It requires additional GPU memory for the 13 index-vectors, but highly reduces the computation time. It differs in the way the boundary conditions of the finite differences are applied. The CUDA kernels are in general only launched for necessary voxels. Checks for the image dimension and boundary are done previously and not in each iteration.

Basically this version of the algorithm is recommended. It is also combined with the previously discussed downsampled algorithm. For cases where the GPU memory is not sufficient, the slower TGV-DCT bias correcton algorithm can be used. Note that the performance improvement depends on the count of foreground voxels and therefore on the mask volume.

4.1.4 Comparison to N4ITK

The N4ITK algorithm is designed for inhomogeneity correction of MR images. It assumes a slow varying bias field. This constrained is forced by using spline interpolation. Third order splines have equal intensity values, first and second order derivatives at the nodes. Intensity values between the nodes are interpolated by third order polynomials. The TGV-DCT method minimizes the TV of the resulting image. Noise is removed by iteratively subtracting the projected derivative and all other non-piecewise-constant contributions become part of the illumination component. Although these two goals are totally different, the results are comparable for the processed samples.

Figure 20 shows the results for increasing TGV-DCT order. In this case higher order algorithms do not improve the results qualitatively, if one is interested in the contrast of the muscle to its surrounding tissue. Nevertheless the metric values in Table 14 decrease.

Simultaneous denoising and bias correction – as described in Section 3.3.5 – greatly improves the metric values. Previous experiments have shown that reduction of the metric values CV, TV and entropy is not enough to quantize good image restoration. The extreme case of bias correction would generate a single intensity value which would end up in the theoretically best metric values. This would surely not be the intended result. Qualitative inspection of the image, PDF and line profile plot would clearly show that the result is getting worse, but the metric values improve (Figure 21 and Table 15). The local contrast between several separated tissue type regions is introduced to quantitatively count for the distance between multiple intensity distributions in the PDF. It is approximated based on the line profile plot. This measurement is assumed to be accurate enough for a meaningful comparison of the processed samples.

The results of Table 16 and Figure 22 are interpreted as follows. N4 very well performs bias correction. It generates maximum contrast between the muscle tissue and all other pixel intensities. The estimated multiplicative bias field is smoothly and slowly varying in the whole image. The downside of the result is that all other tissue classes than muscle get the same intensity value. This produces maximum homogeneity but totally removes the contrast.

Several results of the proposed TGV-DCT algorithm shown in Figure 22 outperform the reference method. Simultaneous denoising additionally produces more homogeneous regions in the image. In contrast to N4 the TGV-DCT algorithm works in the spatial and not the PDF domain. Therefore, it can use the information of spatial separations in the image and thus images with better contrast for multiple very close tissue intensities can be produced. On the other hand it does not estimate a meaningful bias field between isolated objects (oil cylinder in Figures 19 and 20). The algorithm can produce similar results as N4 with default parameter values. Furthermore, it allows to select the amount of denoising and shape of the bias field and hence the parameters can be adjusted to specific situations and desired outcomes.

4.1.5 Evaluation

The calculated metric values show an improvement in most of the tested samples. Although this is generally a hint for good performance, reduction of these values does not directly imply well performed shading correction. In case of very inhomogeneous illumination – more specific if some regions are strongly illuminated and others very less – the spectrum contains many intensity values at the beginning and another group at the end. This means the entropy is low in the overall image. Shading correction increases the entropy in such case. A more representative metric for these scenarios is the coefficient of variation.

Furthermore, the outcome of the experiments shows that if one is interested in the contrast of several tissue classes evaluation should include other metric values – as the local contrast – or some kind of segmentation procedure. In that way the CV, TV and entropy metric could be calculated for several separated regions and combined metrics like the joint coefficient of variation can be used. The problem of bias correction is closely connected to segmentation of homogeneous regions and classification of several intensity distributions in the image.

Note that the contrast metric is even improved by a bias field in some cases. The slowly and smoothly varying field may stretch the intensity difference between different regions. Therefore bias correction could also decrease the contrast by generating good estimates for the real bias-free data.

4.1.6 Convergence of the optimization algorithms

Convergence of the proposed bias correction method depends on the optimization algorithms used. Bias estimation and denoising is done by a TGV-L^1 Primal-Dual implementation. TGV is described in Bredies et al. 2010 [30]. The authors define convergence due to the L^2-norm of the first term in the TV-norm in Equation 8. A convergence estimate for TGV-L^2 is derived which is dependent on the image dimension, number of iterations and α_0. Unfortunately this estimate can not be used for the TGV-L^1 functional. But it is known that it decreases monotonically. In this work convergence is checked visually by the inspection of the image, histogram and line profile plots. Additionally statistical values like the minimum, maximum, mean, standard deviation and CV are examined. If the result changes just in a certain fraction ($1e$-3) of the intensity bandwidth, convergence is assumed.

The second step of the bias correction algorithm transforms the vector field v into the scalar bias l. This problem is stated as a minimization in Equation 2. Convergence of v in the previous step is therefore important. The CG-solver convergences for a positive definite matrix after $N_{voxel}-1$ iterations which is proofed by Hestenes and Stiefel 1952 [43].

The DCT-solver for the Poisson Equation – described in Section 2.3.3 – is a closed form solution. It directly calculates the optimum solution in the cosine domain.

4.1.7 Conditions for Successful Bias Correction by TGV-DCT

To state some conditions for successful bias estimation using TGV-DCT it is important to understand the role of each single term of the underlying TGV regularization functional. Section 2.2.2 describes the first order version. It is known that TV-L^1 generates piecewise constant images. Why does it extract such piecewise constant regions? The following discussion about TGV-DCT is based on the fact that every function can be decomposed by Taylor series expansion. For discretized images on a regular grid the distance from the parameter for which the function is approximated to the origin parameter is constant and usually defined to be simply one. This more specific version of the Taylor series is shown in Equation 18. It is based on the Taylor expansion method described by Dahmen and Reusken 2008 [52]. The scalar function value for the parameter $x + 1$ is equal to the function value left to it plus an infinite series of weighted derivatives.

$$f(x+1) = f(x) + \sum_{i=1}^{\infty} \frac{f^{(i)}(x)}{i!} \tag{18}$$

Putting the first term to the left side of the equation shows that the finite forward difference is equal to the weighted sum of all derivatives. Note that the gradient operator is usually approximated by the forward difference with Neumann zero flux boundary conditions. The implementation of TV-L^1 minimizes the L^1-norm of this gradient. Since L^1 is a special p-norm the minimization functional can be transformed by the Minkowsky Inequality to the separate minimization of all derivatives. The aim of TV-L^{1}is therefore interpreted to be the minimization of really all derivatives. In combination with the data term the optimization produces piecewise constant (all derivatives near zero) images which are similar to the input image.

For the investigation of second order TGV-DCT the previously stated Taylor Series Equation 18 is transformed to additionally contain the first derivative of the function on the left side. After this transformation the weighted sum of all higher order derivatives (starting at order two) is equal to the forward difference minus the first derivative. If one would minimize the terms on the left side this would minimize all derivatives except the first one. Such an optimization would therefore extract linear functions. The first term of the TGV^2-DCT energy functional in Equation 8 describes the minimization of the L^1-norm of the gradient of the denoised image u subtracted by a vector field v. The second term forces the minimization of the gradient of this vector field. According to the discussion of TV in the previous paragraph if v would contain the first order derivative the second term would minimize all derivatives of v.

Algorithm 4 iteratively minimizes all terms of Equation 8. The derivative of the Lagrange with respect to u and v is required to obtain the update terms for the iterative minimization. The dual variable for u is p. It contains the summed up projected forward difference of u minus the vector field v. The information of p is used for the primal update of u and v. Note that both terms of TGV^2-DCT in Equation 8 contain v. The primal update of v is therefore done by summing up p (the derivative of the first TGV^2-DCT term with respect to v) and a second term containing q. The role of q is the minimization of the L^1-norm of the gradient of v. At the beginning $\bar{v}$ is initialized with zero. The variable

p therefore contains the projected weighted sum of all derivatives in early iterations. And the vector field v stores the forward difference of u (as part of p) and the weighted sum of all higher order derivatives (part of p and q). According to Equation 18 this is equal to the first order derivative. The information of the first derivative slowly transfers from p to v with increasing iterations. TGV^2-L^1 therefore approximates piecewise linear regions in the input image. Note that the approximated bias field by TGV^2-DCT does not contain the residual of this linear approximation. In contrast the downsampled version of the algorithm does contain the residual in the bias field. The maximum step size for q has to be greater than the one for p. The vector field v quickly becomes the first order derivative and the primary minimization of all derivatives except the first one is done via the vector field p.

Third order TGV-DCT extracts piecewise quadratic functions. In contrast to TGV^2-DCT the second term does not force all derivatives of v to be minimized. It rather subtracts another vector field w which contains the quadratic information of u. This second term of TGV^3 is therefore similar to the first term of TGV^2.

In the following paragraph this argumentation is extended to the general case of any higher order TGV-DCT. Equation 10 describes the TGV functional as a minimization. For higher order terms it contains a general series which forces the minimization of higher order derivatives up to the order k. Since the maximum step sizes increase for higher order terms the algorithm extracts higher order information first. Additionally higher order derivation is done by the derivation of one order less as described above. Continuing with higher order terms successively describes the approximation of piecewise higher order polynomials in the image. Note that the vector field v still contains all non-constant information left in the scalar field u. If the objects under interest are assumed to be piecewise constant, this vector field v can be used to approximate the bias field of the image.

With increasing order the algorithm fits piecewise regions with intensity distributions of more complex shape. The downside of increasing order is that if the bias component does not consist of such higher order polynomials the algorithm starts to fit the piecewise constant objects with higher order terms. This behaviour is known under the name overfitting. Therefore parts of the object information contribute to the bias and are removed.

Increasing step size values from the first to higher derivatives (Section 2.2.5) mean that higher order information is minimized faster. Therefore it converges earlier and lower order information needs to be refined in later iterations. A general factor for α is introduced in the algorithm input subroutine. Too large step sizes blur the image in the early iterations and sharp edges return later. Lower step sizes do not blur the image and lead to faster convergence. But too low step sizes can result in a very slow correction process. This behavior is interpreted as follows. Too large step sizes let the algorithm make large updates which may pass over the desired result of the optimization. Such steps introduce error to the result which may remain in the data. On the one hand small steps are more accurate, but on the other they are slower.

The common assumption about the bias field (slow varying and smooth) is included in the TGV-DCT model by $\alpha_0 < \alpha_1$. These two weights are factors of energy terms in Equation 10. If the first step size is smaller, second order derivatives of the bias field are minimized faster. The effect is less edge-information and therefore smoothness of the bias. In this work the shape of the field is controlled by the TGV-DCT order and the downsampling factor only.

Each term in the energy functional of $\mathrm{TGV}^{\mathrm{k}}$ (Equation 10) is weighted. The factor for the data term of TGV-L^1 is λ and for the L^1-norm of the derivatives is α. The shape of the estimated bias field depends on the relative value of each integral compared to the others. The weights can therefore be adjusted due to the specific image dimensions, included noise and shape of the bias field.

The proposed TGV-DCT bias correction method is based on the fundamental assumption that the objects under interest are piecewise constant and higher order content corresponds to the bias component. Second order TGV-DCT fits a piecewise linear inhomogeneity field. If the order is increased and there is no higher order bias, the effect described in the previous paragraph occurs. Object information wrongly contributes to the higher order bias. This leads to decreasing contrast and a very narrow single histogram peak. It is therefore recommended to start with second order TGV-DCT and increase the order if it is necessary and helpful.

4.1.8 MR Coil Sensitivity Estimation

The MR signal model usually contains a multiplicative bias component as described in Section 1.2.2. In contrast to that, the proposed method approximates an additive bias field. Nevertheless it is possible to transform this estimate to a multiplicative one. This is shown in Figure 17 and further described in Section 3.3.3.

Anyway this estimate may contain several disturbing components. If the assumption of piecewise constant objects is not perfectly fulfilled, the sensitivity profile contains part of the information of the imaged objects.

The algorithm fits the optimal piecewise constant image to the data. Multiple regions of the same tissue type – which are not connected – may not get equal intensity values. Especially if the bias values in such regions are highly varying. Tuning the α parameter may help in such cases. It controls the shape of the bias field, and the proposed bias correction algorithm does include the assumption of smoothly and slowly varying inhomogeneities ($\alpha_0 < \alpha_1$).

A method based on a multiplicative bias model might result in better performance in such situations. But optimization becomes more difficult for increasing correlation of the estimated parameters.

4.2 Logarithm Transform

In section Section 1.2.2 the logarithm transform of the input image is mentioned. The multiplicative bias field is transformed into an additive one by many authors [1, 4–13]. Simply taking the logarithm could become problematic with very low values [15]. Usually the intensities are rescaled to values from 1 to 2. The inverse transform - which is necessary to estimate the bias-free image - is the exponential function and subtracting 1 from the resulting values.

Experiments have shown that the TGV-DCT outcome does not change if the data is processed in the logarithm domain. Note that the logarithm transform alters the image in the PDF domain. This independently alters each voxel of the image. Piecewise constant functions remain piecewise constant after taking the logarithm. In the range from 1 to 2 the logarithm is nearly linear. Nevertheless piecewise linear functions become nonlinear. And the value range of the bias free and the bias image is usually not known.

In this work this transformation is omitted. The proposed TGV-DCT algorithm approximates an additive bias field. As can be seen in Figure 1, this model successfully corrects an additive bias. The following samples in Figures 2 to 4 show that also multiplicative corruptions are partly restored. And finally the model also corrects measured MR volumes (Figures 11 to 14, 20 and 21).

4.3 Recommendation for Choosing the Parameters

The outcome of the algorithm is controlled by the choice of the parameters. For the second order TGV-DCT algorithm these are the denoising parameter λ (in Equation 4) and the weights for TGV terms α_0 and α_1 (in Equation 10).

The second term in Equation 10 includes the denoised image and the gradient of the shading component. It is therefore recommended to fix α_0 to 1, because both images depend on this parameter.

The first parameter to adjust is the denoising parameter. Run about $1e3$ iterations with the setting $\lambda = 1$, $\alpha_0 = 1$ and $\alpha_1 = 2$. The result may not converge totally, but the effect of the denoising parameter will be visible quite soon. For more intensive denoising (less details) choose a value in the range of $(0; 1]$. If the algorithm should perform less denoising, λ has to be increased. Values smaller than 3 are recommended.

Finally choose the shape of the bias field by setting the parameter α_1. This is shown in Figure 23. Bigger values weight the second term of Equation 10 stronger, which makes the bias more smooth. Experiments have shown that a value in the range of $(2; 5]$ is a feasible choice. This recommendation assumes that the intensity values are in the standard range of the DICOM spectrum.

4.4 Conclusion

Even with the default parameter set the algorithm produces meaningful results for the listed samples. Furthermore the parameters can be adjusted to specific situations. The fundamental assumption is that the objects under interest are piecewise constant. TGV^{k}-L^{1} combined with the DCT solver for the Poisson Equation approximates such piecewise constant images.

The DCT Poisson solver is preferred over the CG-solver. The results of both methods are nearly equal (Table 17). But the direct form solution of the DCT-solver is much faster. The proposed method corrects most of the bias without destroying interesting details and contrast between different regions.

Based on the idea of Liang and Zhang 2015 [11] the TGV-L^{1} algorithm of Bredies et al. 2010 [30] is combined to the DCT Poisson solver by Limare et al. 2011 [29]. Then the algorithm is extended to perform higher order TGV-DCT. Furthermore, a downsampled and a masked version is introduced. The overall method is implemented in CUDA for 3D images and successfully evaluated due to several data sets.

4.5 Outlook

The maximum step size vector α balances the regularization terms and therefore controls the shape of the extracted bias field. Analysis, tuning and optimization of this parameter is a topic for future research. Based on the insights of this work investigation of optimal TGV-DCT order is another open task. This may be done by a histogram based clustering algorithm, which evaluates the contrast of several intensity distributions.

For general color photographs the higher order TGV-DCT algorithm shows potential for further investigations. As a preprocessor step this method may improve the performance of other imaging tasks like segmentation, registration and classification. Bias information may also be removed due to lowering the entropy and thus compress image files. By removing slow biases, 3D volume rendering of MR data can be presented for a more intuitive view of tissues under interest.

The second order TGV regularization term is successfully used for radial MR data reconstruction by Knoll et al. 2011 [38]. Similar to the denoising task the TGV of the resulting image is minimized. The proposed method could therefore be included in the reconstruction step which would simultaneously perform transformation of the measurement data from the k-space, denoising and bias correction.

Inhomogeneity correction is an important step for quantitative imaging and also the generation of multimodal images. The algorithm may also be used for other imaging modalities like microscopy, computer tomography, ultrasound since these techniques contain similar non-uniform sensitivity profiles.

5 Literature

[1] U. Vovk, F. Pernus, and B. Likar. A review of methods for correction of intensity inhomogeneity in MRI. *IEEE Trans Med Imaging*, 26(3):405–421, March 2007.

[2] B. Belaroussi, M. Milles, S. Carme, Y. M. Zhu, and H. Benoit-Cattin. Intensity non-uniformity correction in MRI: Existing methods and their validation. *Med Image Anal*, 10(2):234 – 246, 2006.

[3] F. Knoll, C. Clason, K. Bredies, M. Uecker, and R. Stollberger. Parallel imaging with nonlinear reconstruction using variational penalties. *Magn Reson Med*, 67(1):34–41, 2012.

[4] M. Elad, R. Kimmel, D. Shaked, D. Shaked, R. Keshet, R. Keshet, and Hewlett packard Laboratories Israel. Reduced complexity retinex algorithm via the variational approach. *IEEE Trans Image Process*, 14:369–388, 2002.

[5] Q. Zhao, P. Tan, Q. Dai, L. Shen, E. Wu, and S. Lin. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Trans Pattern Anal Mach Intell*, 34(7):1437–1444, July 2012.

[6] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel. A variational framework for retinex. *Int J Comput Vision*, 52(1):7–23, April 2003.

[7] W. Ma and S. Osher. A TV bregman iterative model of retinex theory. SUCLA CAM Report 10-13, 2010.

[8] J. M. Morel, A. B. Petro, and C. Sbert. A PDE formalization of retinex theory. *IEEE Trans Image Process*, 19(11):2825–2837, November 2010.

[9] W. Ma, J. M. Morel, S. Osher, and A. Chien. An L1-based variational model for retinex theory and its application to medical images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 153–160, June 2011.

[10] Y. Zang, Z. Pan, J. Duan, G. Wang, and W. Wei. A double total variation regularized model of retinex theory based on nonlocal differential operators. In *Image and Signal Processing (CISP), 2013 6th International Congress on*, volume 01, pages 213–218, Dec 2013.

[11] J. Liang and X. Zhang. Retinex by higher order total variation L1 decomposition. *J Math Imaging Vis*, 52(3):345–355, July 2015.

[12] J. Chang and J. Bai. An image enhancement algorithm based on gaussian weighted bilateral filtering and retinex theory. In *2015 8th International Congress on Image and Signal Processing (CISP)*, pages 257–262, Oct 2015.

[13] X. Fu, Y. Liao, D. Zeng, Y. Huang, X. P. Zhang, and X. Ding. A probabilistic method for image enhancement with simultaneous illumination and reflectance estimation. *IEEE Trans Image Process*, 24(12):4965–4977, Dec 2015.

[14] L. Axel, J. Costantini, and J. Listerud. Intensity correction in surface-coil MR imaging. *AJR Am J Roentgenol*, 148(3):418–420, February 1987.

[15] J. G. Sled, A. P. Zijdenbos, and A. C. Evans. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Tran Med Imaging*, 17(1):87–97, Feb 1998.

[16] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee. N4ITK: Improved N3 bias correction. *IEEE Trans Med Imaging*, 29(6):1310–1320, June 2010.

[17] J. V. Manjón, J. J. Lull, J. Carbonell-Caballero, G. García-Martí, L. Martí-Bonmatí, and M. Robles. A nonparametric MRI inhomogeneity correction method. *Med Image Anal*, 11(4):336 – 345, 2007.

[18] E. H. Land, , and J. J. McCann. Lightness and retinex theory. *J Opt Soc Am*, pages 1–11, 1971.

[19] C.-T. Shen and W.-L. Hwang. Color image enhancement using retinex with robust envelope. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3141–3144, Nov 2009.

[20] W.-H. Chao, H.-Y. Lai, Y.-Y. I. Shih, Y.-Y. Chen, Y.-C. Lo, S.-H. Lin, S. Tsang, R. Wu, and F.-S. Jaw. Correction of inhomogeneous magnetic resonance images using multiscale retinex for segmentation accuracy improvement. *Biomed Signal Process Control*, 7(2):129 – 140, 2012.

[21] Y.-K. Wang and W.-B. Huang. A cuda-enabled parallel algorithm for accelerating retinex. *Journal of Real-Time Image Processing*, 9(3):407–425, 2014.

[22] J. M. Morel, A. B. Petro, and C. Sbert. What is the right center/surround for retinex? In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4552–4556, Oct 2014.

[23] X. Xie. Illumination preprocessing for face images based on empirical mode decomposition. *Signal Processing*, 103:250 – 257, 2014. Image Restoration and Enhancement: Recent Advances and Applications.

[24] C. Damerval, S. Meignen, and V. Perrier. A fast algorithm for bidimensional EMD. *IEEE Signal Process Lett*, 12(10):701–704, Oct 2005.

[25] L. Liang and Y. Si. Medical image enhancement using sliding weighted empirical mode decomposition. In *Information and Automation, 2015 IEEE International Conference on*, pages 3145–3148, Aug 2015.

[26] A. N. Tikhonov, A. S. Leonov, and A. G. Yagola. Nonlinear ill-posed problems. In *World Congress of Nonlinear Analysts '92*, volume I-IV, pages 505–511, 1992.

[27] L. Wang, L. Xiao, H. Liu, and Z. Wei. Variational bayesian method for retinex. *IEEE Trans Image Process*, 23(8):3381–3396, Aug 2014.

[28] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259 – 268, 1992.

[29] N. Limare, A. B. Petro, C. Sbert, and J. M. Morel. Retinex poisson equation: amodel for color perception. *Image Processing On Line*, 1, 2011.

[30] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM J Imaging Sci*, 3(3):492–526, 2010.

[31] N. Verma, M. C. Cowperthwaite, and M. K. Markey. Variational level set approach for automatic correction of multiplicative and additive intensity inhomogeneities in brain MR images. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 98–101, Aug 2012.

[32] Z. Shahvaran, K. Kazemi, M. S. Helfroush, N. Jafarian, and N. Noorizadeh. Variational level set combined with markov random field modeling for simultaneous intensity non-uniformity correction and segmentation of MR images. *J Neurosci Methods*, 209(2):280 – 289, 2012.

[33] T. Ivanovska, R. Laqua, L Wang, A. Schenk, J H Yoon, K Hegenscheid, H. Völzke, and V. Liebscher. An efficient level set method for simultaneous intensity inhomogeneity correction and segmentation of MR images. *Comput Med Imaging and Graph*, 48:9 – 20, 2016.

[34] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Model Simul*, 7(3):1005–1028, 2009.

[35] L. P. Yaroslavsky. Local criteria: a unified approach to local adaptive linear and rank filters for image restoration and enhancement. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, pages 517–521 vol.2, Nov 1994.

[36] D. Zosso, G. Tran, and S. J. Osher. Non-local retinex—a unifying framework and beyond. *SIAM J Imaging Sci*, 8(2):787–826, 2015.

[37] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J Math Imaging Vis*, 40(1):120–145, 2011.

[38] F. Knoll, K. Bredies, T. Pock, and R. Stollberger. Second order total generalized variation (TGV) for MRI. *Magn Reson Med*, 65, 2011.

[39] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, Jan 1998.

[40] M. Elad. *Retinex by two bilateral filters*, pages 217–229. Springer, Berlin, Heidelberg, 2005.

[41] M. Holler and K. Kunisch. On infimal convolution of TV-type functionals and applications to video and image reconstruction. *SIAM J Imaging Sci*, 7(4):2258–2300, 2014.

[42] M. Frigo and S.-G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[43] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J Res Nat Bur Stand*, 49(6):409–436, December 1952.

[44] fftw.org. 1d real-even DFTs (DCTs) - FFTW 3.3.4. `http://www.fftw.org/fftw3_doc/1d-Real_002deven-DFTs-_0028DCTs_0029.html#1d-Real_002deven-DFTs-_0028DCTs_0029`, Downloaded: July 2016.

[45] J. B. Arnold, J.-S. Liow, K. A. Schaper, J. J. Stern, J. G. Sled, D. W. Shattuck, A. J. Worth, M. S. Cohen, R. M. Leahy, J. C. Mazziotta, and D. A. Rottenberg. Qualitative and quantitative evaluation of six algorithms for correcting intensity nonuniformity effects. *NeuroImage*, 13(5):931 – 943, 2001.

[46] W. Huang and Z. Kong. Kernel density estimation and metropolis-hastings sampling in process capability analysis of unknown distributions. In *Proceedings of the ASME 2012 International Manufacturing Science and Engineering Conference MSEC2012*, June 2012.

[47] S. L. Keeling, M. Hintermüller, F. Knoll, D. Kraft, and A. Laurain. A total variation based approach to correcting surface coil magnetic resonance images. *Appl Math Comput*, 218(2):219 – 232, 2011.

[48] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, G. B. Pike, and A. C. Evans. Brainweb: Online interface to a 3D MRI simulated brain database. *NeuroImage*, 5:425, 1997.

[49] E. H. Adelson. Checkershadow illusion. http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html, Downloaded: July 2016.

[50] opencv.org. Open source computer vision library. https://github.com/opencv/opencv/, Downloaded: July 2016.

[51] B. B. Avants, N. J. Tustison, and H. J. Johnson. Advanced normalization tools (ants). https://github.com/stnava/ANTs, Downloaded: July 2016.

[52] M. Dahmen and A. Reusken. *Numerik für Ingenieure und Naturwissenschaftler.* Springer Berlin Heidelberg, 2008.