Fast segmentation from blurred data in 3D fluorescence microscopy

Martin Storath, Dennis Rickert, Michael Unser, Andreas Weinmann

***Abstract*—We develop a fast algorithm for segmenting 3D images from linear measurements based on the Potts model (or piecewise constant Mumford-Shah model). To that end, we first derive suitable space discretizations of the 3D Potts model which are capable of dealing with 3D images defined on non-cubic grids. Our discretization allows us to utilize a specific splitting approach which results in decoupled subproblems of moderate size. The crucial point in the 3D setup is that the number of independent subproblems is so large that we can reasonably exploit the parallel processing capabilities of the graphics processing units (GPU). Our GPU implementation is up to 18 times faster than the sequential CPU version. This allows to process even large volumes in acceptable runtimes. As a further contribution, we extend the algorithm in order to deal with non-negativity constraints. We demonstrate the efficiency of our method for combined image deconvolution and segmentation on simulated data and on real 3D widefield fluorescence microscopy data.**

***Index Terms*—Image segmentation, 3D images, Potts model, piecewise constant Mumford-Shah model, parallelization, GPU, non-negativity constraints**

I. Introduction

SEGMENTATION is an important component of many image processing pipelines. For instance, a segmentation of the image can be used to extract objects of interest such as, e.g., nuclei of cells in microscopic images. Furthermore, it can serve for dimensionality reduction by aggregating related pixels into larger regions ("superpixels"). In many imaging modalities however, the image of interest cannot be recorded directly, but it is reconstructed from indirect linear measurements given by the device. The relation between the imaged object and the data is given by some imaging operator; for example, the data acquisition in computed tomography is described by the Radon transform, and the operators in microscopy are often assumed to be shift-invariant which results in operators of convolution type.

Segmentation of indirectly measured images is typically performed in two steps: first the image is reconstructed from the measured data by (stabilized) inversion of the imaging operator; then, the resulting image is partitioned or segmented. The advantage of this two stage procedure is that there is a large number of methods available for the separate steps. For instance in 3D fluorescent microscopy, the deblurring stage can be accomplished by the Tikhonov-Miller method, the (regularized) Richardson-Lucy method [1–3], or wavelet methods [4], and the segmentation stage by k-means, active contours [5], or graph cuts [6], to mention only a few. On the flipside, the results of such two stage approaches are suboptimal since the distortions of the reconstruction step propagate to the segmentation step. In particular when data is poor, combined approaches, i.e. joint reconstruction and segmentation, exhibit better performance [7–12]. We here focus on a classical, yet challenging, variational segmentation approach called the *Potts model* or the *piecewise-constant Mumford-Shah model* [13–15]. For indirectly measured images, it is given by

$$u^* = \arg\min_u \gamma \, \|\nabla u\|_0 + \|Au - f\|_2^2. \qquad (\mathscr{P}_\gamma)$$

Here, A denotes the mathematical model for the imaging operator, and f denotes the measured data. The inverse Potts model is an optimization problem of two concurring objectives. The term $\|Au - f\|_2^2$ ensures fidelity to the measured data whereas the Potts prior $\|\nabla u\|_0$ enforces a small jump set (or discontinuity set) of a minimizer. The relative weight of the two target functionals is regulated by the parameter γ; the higher γ, the smaller the jump set of a solution. The value of the jump-penalty $\|\nabla u\|_0$ can only be small if u is a piecewise constant function. Thus, a solution u^* can be regarded as reconstruction from data f with a piecewise constant prior; see also [7, 12, 15]. The desired segmentation is encoded in the piecewise constant function u^*: the constant regions of u^* correspond to the segments, and the jump set of u^* corresponds to the segment boundaries. Moreover, the function value of u^* on a segment gives an estimate on the mean intensity on that segment.

A. Prior and related work

The Potts model is named after R. Potts who considered a related regularizer in his seminal work in statistical mechanics [13]. The classical Potts model, i.e., $(\mathscr{P}_\gamma)$ with $A = \mathrm{id}$, was introduced to image processing by Geman and Geman [14] in a statistical framework. From a calculus of variation point of view, the problem was first considered by Mumford and Shah [15]; there it is also known as piecewise constant Mumford-Shah model. In recent work, $(\mathscr{P}_\gamma)$ is also referred to as the ℓ_0 gradient model [16, 17]. The Potts problem is – even for $A = \mathrm{id}$ – computationally challenging: it is known to be NP hard in dimension greater than one [18, 19]. Nevertheless, the Potts model is a very popular model for image segmentation; see [14, 19–21] to mention only some examples. For the case $A = \mathrm{id}$, classical algorithms are based on simulated annealing

M. Storath is with the Image Analysis and Learning Group, Universität Heidelberg, Germany.

D. Rickert is with the Institute of Computational Biology, Helmholtz Zentrum München, Germany.

M. Unser is with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Switzerland.

A. Weinmann is with the Institute of Computational Biology, Helmholtz Zentrum München and with the Department of Mathematics, University of Applied Sciences Darmstadt.

[14] and on approximation by elliptic functionals [22]. A non-exhaustive selection of popular approaches for $A = \mathrm{id}$ are active contours [5, 23–25], graph cuts [6, 19], convex relaxations [26–29], semi-global matching [30, 31], iterative thresholding [16, 32], fused coordinate descent [33], region fusion [17], and alternating direction method of multipliers (ADMM) [34]. Although the latter approach is computationally more demanding than other recent approaches [16, 17, 33], it gives the best quality in practice; see [17]. In [6, 35], graph cuts are used for partitioning 3D images with discretized label spaces. 3D active meshes are used in [36] for cell segmentation and tracking in microscopic images. Since the Potts problem is computationally rather expensive, parallelization for graphics processing units (GPU) has recently received a lot of attention; see for example [26, 29, 37].

The setup we focus on, the Potts problem with $A \neq \mathrm{id}$, has been studied to a much lower extent. On the theoretical side, the regularizing properties of the inverse Potts model in the sense of inverse problems have been investigated in [9, 38–40]. We stress that the inverse Potts problem is computationally even more demanding than the classical Potts problem. For instance, the inverse Potts problem is NP hard even in one dimension [41, 42]. Early algorithmic approaches are based on Ambrosio-Tortorelli approximations [7, 43]. Rondi and Santosa [7] use this approach in electrical impedance tomography; Bar et al. [43] apply it for semi-blind image deblurring. Fornasier and Ward [32] propose iterative thresholding algorithms and present applications to inpainting problems; see also [44, 45]. A related iterative thresholding type algorithm has been proposed for deconvolution problems [46]. An iterative Potts minimization scheme has been developed in [47]. Kim et al. [48] propose a level-set active contour approach for deconvolution problems. Ramlau and Ring [9] use a level-set approach for the joint reconstruction and segmentation of x-ray tomographic images; related methods are applied to electron tomography [49] and SPECT [38]. Active contours are also used for deconvolving fluorescence microscopy images in [50]. An ADMM strategy has been proposed in [12]. We note that the existing algorithmic approaches focus mostly on the 2D setup, which is mainly due to the high computational costs.

B. Challenges in 3D and Contribution

Here, we focus on the inverse Potts problem for 3D images. Besides the problems already encountered in 2D, there are additional challenges in 3D: *(i)* In many imaging modalities, the resolution in the x_3-direction is typically much lower than in the other directions. This has to be taken into account when discretizing the Potts prior $\|\nabla u\|_0$. *(ii)* The main challenge for 3D data is the high computational effort due to two reasons: the number of pixels/voxels is typically much larger in 3D volumes than in planar images, and the 3D neighborhood system is bigger than the 2D system (e.g., 8-connected neighborhood in 2D vs. 26-connected neighborhood in 3D). Hence there is need for efficient algorithms that are able to process even large 3D images in reasonable time. In particular, algorithms that can be parallelized well are important to obtain feasible runtimes without loss of segmentation quality.

In this work, we propose an efficient algorithmic framework for joint 3D-image reconstruction and multilabel segmentation based on the Potts model. We show its applicability for joint reconstruction and segmentation in 3D fluorescence microscopy. In particular, we contribute by addressing the challenges *(i)* and *(ii)* as follows. Regarding *(i)*, we consider the discretization of the Potts prior in 3D for non-cubic regular grids reflecting the different resolutions w.r.t. different directions. Our discretization is based on weighted directional difference scheme which has been introduced in the two-dimensional case by Chambolle [51] and which has been refined in a previous work of the authors [34]. For non-cubic regular grids, we here propose a new discretization that takes the different resolutions w.r.t. the different directions into account. We here show the practical benefits in numerical experiments on synthetic and real data.

Regarding *(ii)*, we extend a splitting strategy which was originally developed by the authors for 2D image partitioning [12]. It relies on the alternating directions method of multipliers (ADMM) which leads to subproblems that can be efficiently solved by dynamic programming. As mentioned before, the ADMM has shown the largest potential with respect to quality but is computationally demanding in 3D. The crucial point of the proposed design is that the algorithm is parallelizable in a way that it leads to a significant speedup on the GPU. We demonstrate the computational advantage of the subproblem parallelization by implementing a tailored parallel GPU version of the algorithm which is, on a single GPU, up to around 18 times faster than a sequential implementation. This is important to be able to process larger volumes in reasonable time.

Besides the specifics of the 3D setup, we further address the following general issue: *(iii)* The gray-values (or color-values) of a reconstructed image often correspond to physical quantities that cannot take on negative values; for example the intensities in fluorescence microscopy. To take this into account, we impose a non-negativity constraint which leads to the non-negative Potts model

$$\arg\min_u \gamma \, \|\nabla u\|_0 + \|Au - f\|_2^2, \quad \text{subject to } u \geq 0. \qquad (\mathscr{P}_\gamma^+)$$

For this model, we derive a new algorithm which is virtually as efficient as the unconstrained version.

Finally, we stress that our method neither requires a priori knowledge on the number of segments and labels nor on the gray-values of a solution. Further, the method does not require any initial guess of the solution.

In our numerical experiments, we focus on 3D image segmentation from blurred measurements. We first evaluate the capabilities of our method on synthetic datasets with ground truth. Then we apply it to real data of a wide field fluorescence microscope. Using our parallel GPU implementation, we achieve a joint deconvolution and segmentation of a $400 \times 400 \times 50$ image in less than three minutes. We note that the experiments are based on depth-invariant point spread functions. More sophisticated models for 3D fluorescence microscopy use depth-variant point spread functions which can give more accurate restorations [52, 53]. Further, we have restricted ourselves to imaging modalities which are modeled by linear measurements, and in particular convolution operators. We note that the present algorithm can be adapted to nonlinear imaging operators, e.g. those appearing in electrical impedance tomography [54]. The relevant condition is that we can solve the corresponding Tikhonov-regularized problem within reasonable time.

C. Organization of the paper

In Section II, we derive a suitable discretization of the Potts prior for 3D images defined on cubic and non-cubic regular grids. In Section III, we present the proposed computational approach to ($\mathscr{P}_\gamma^+$), and we elaborate on the solution of the arising subproblems. Section IV deals with numerical experiments on synthetic data and on real data.

II. Discretization of the Potts prior for 3D images

As a first step, we derive a suitable discretization of the jump penalty $\|\nabla u\|_0$. In the following, we assume that u is a vector-valued N-dimensional image, i.e., $u \in X = \mathbb{R}^{d_1 \times d_2 \ldots \times d_N \times C}$. Here, C denotes the number of channels of the image; for example, $C = 3$ for color images. We use a finite difference discretization of the form

$$\|\nabla u\|_0 = \sum_{s=1}^{S} \omega_s \|\nabla_{a_s} u\|_0. \tag{1}$$

The vectors $a_s \in \mathbb{Z}^N \setminus \{0\}$ belong to a finite difference system $\mathcal{N}$ with $S \geq N$ elements. For $a \in \mathbb{Z}^N$, we let

$$\|\nabla_a u\|_0 = |\{i = (i_1, \ldots, i_N) : u_{i,:} \neq u_{i+a,:}\}|,$$

where we use the notation $u_{i,:} = (u_{i,1}, ..., u_{i,C}) \in \mathbb{R}^C$ to denote the data sitting in the voxel with coordinates i. $\nabla_a u$ can be seen as a finite difference approximation of the directional derivative of u with respect to the direction a. Optimized discretizations of this form were initiated for planar images ($N = 2$) by Chambolle [51]. In the simplest case, $\mathcal{N}$ consists of the coordinate unit vectors $e_1, \ldots, e_N$ and the weights $\omega_1, \ldots, \omega_N$ are all equal to one. Unfortunately, this simple discretization corresponds to measuring lengths in the Manhattan metric which may lead to undesired block building in the results; see [34, 51].

A. Near-isotropic discretization for 3D images defined on cubic grids

To get more isotropic discretizations, we utilize larger finite difference systems. Such a system can be chosen rather arbitrarily in principle; the only hard constraint is that the system spans $\mathbb{R}^N$. In practice however, it is reasonable to start with the coordinate unit vectors $e_1, \ldots, e_N$ and to successively enlarge the neighborhood system by adding additional directions. These extra directions can be planar diagonal finite difference vectors such as $e_1 + e_2, e_1 + e_3, \ldots,$ and space diagonals such as $e_1 + \ldots + e_N$.

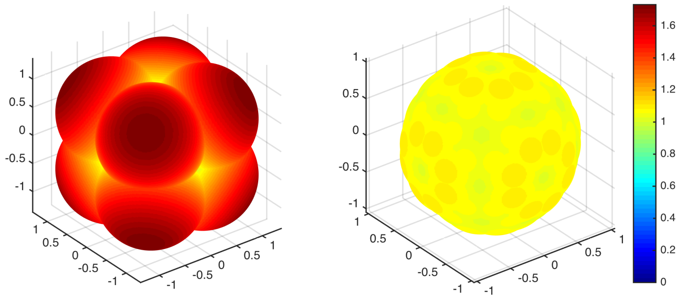


Fig. 1: Near-isotropic discretization on a cubic grid. Penalties induced by the anisotropic discretization (left) and the quasi-isotropic discretization (right). A point x on the surface encodes a direction $x/|x|$ with its jump penalty $|x|$. (The color code corresponds to $|x|$.) The diagonals are penalized to much by the anisotropic discretization. The proposed discretization comes close to the desired unit ball.

In the remainder, we will focus on 3D. There, 26-connected neighborhood systems are frequently used. Identifying antipodal points, this leads to the finite difference system

$$\begin{aligned}\mathcal{N} = \{&(1,0,0),(0,1,0),(0,0,1),\\ &(1,1,0),(1,-1,0),(1,0,1),\\ &(1,0,-1),(0,1,1),(0,1,-1),\\ &(1,1,1),(1,1,-1),(1,-1,-1),(-1,1,-1)\}.\end{aligned} \tag{2}$$

The system contains three coordinate directions, six planar diagonal directions, and four volumetric diagonals, which amounts to 13 orientations in total.

We now derive appropriate weights ω. To this end, we note that a finite difference system gives rise to a metric on $\mathbb{R}^N$ induced by the norm

$$\|x\|_{\mathcal{N}} = \sum_{s=1}^{S} \omega_s |\langle x, a_s \rangle|, \qquad x \in \mathbb{R}^N. \tag{3}$$

A reasonable condition for the weights is that the length of the vectors in the neighborhood system coincides with their Euclidean length [12]; that is,

$$\|a_s\|_{\mathcal{N}} = \|a_s\|_2 \quad \text{for all } s = 1, ..., S. \tag{4}$$

Plugging these S conditions into (3), we get the linear system of equations

$$T\omega = r, \tag{5}$$

where T is an $S \times S$ matrix given by $T_{st} = |\langle a_s, a_t \rangle|$ and the righthand side by $r_s = \|a_s\|_2$.

Proposition 1. *The weights associated with the 26 connected neighborhood system in* (2) *are given by*

$$\begin{aligned}\omega_1 = \ldots = \omega_3 &= \frac{2}{\sqrt{3}} - 1,\\ \omega_4 = \ldots = \omega_9 &= \frac{1}{6}\left(3\sqrt{2} - 2\sqrt{3}\right),\\ \omega_{10} = \ldots = \omega_{13} &= \frac{1}{6}\left(3 - 3\sqrt{2} + \sqrt{3}\right).\end{aligned} \tag{6}$$

A proof is provided in the appendix. Next, we quantify the gain of this discretization. A reasonable measure of isotropy is the ratio of the shortest to the longest vector (measured in the Euclidean length) on the unit ball with respect to $\|\cdot\|_{\mathcal{N}}$,

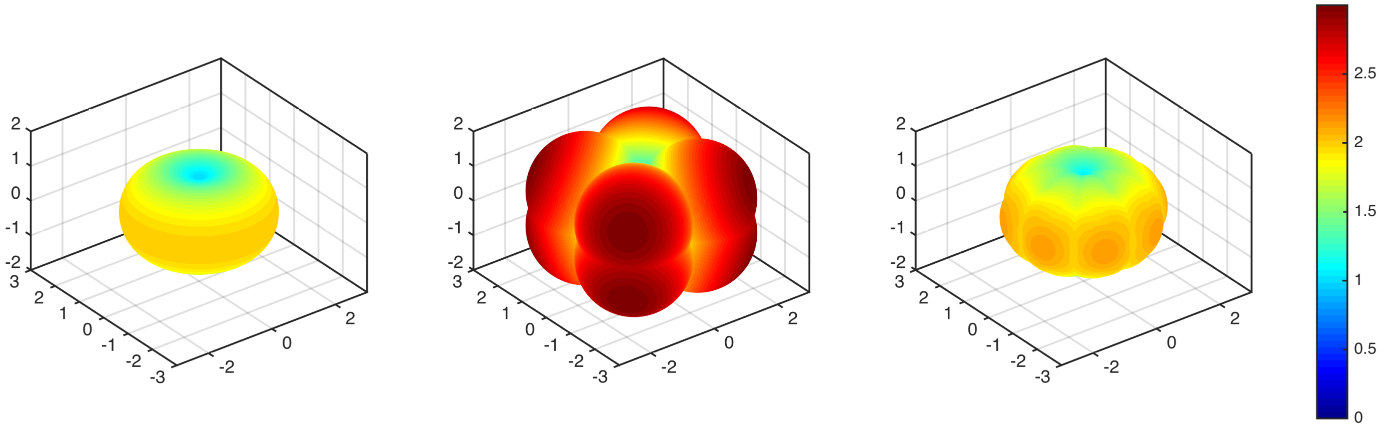


Fig. 2: Near-isotropic discretization on the non-cubic regular grid $\Delta = (1, 1, 2)$. A point x on the surface corresponds to the direction $x/|x|$, and its penalty into that direction is $|x|$. The desired penalties define an ellipsoid (left) with the semi-axis lengths $\delta = (2, 2, 1)$. The anisotropic discretization (middle) reproduces the Euclidean length in the coordinate directions but puts too much weight on the diagonals. The proposed weight design (right) yields a satisfactory approximation to the desired Euclidean length.

see [51]. The value 1 corresponds to perfect isotropy. For the simple 6-connected neighborhood we obtain the unfavorable ratio $1/\sqrt{3} \approx 0.58$. For the proposed neighborhood system, the ratio is approximately 0.91, which is much closer to the desired value 1. This is illustrated in Figure 1.

B. Discretization for 3D images defined on non-cubic regular grids

The sampling rates of 3D images are often not equal with respect to all directions. For example in microscopic images, the sampling density in the x_1-x_2-plane is typically much higher than in the x_3 direction. Then the above discretization can lead to distortions of the length measure in x_3 direction. Therefore, we here design weights that account for images with direction-dependent sampling densities. Let us assume that the distance between two pixels in x_n direction is given by Δ_n; i.e., the voxels are cubes with the side lengths $\Delta_1, \Delta_2, \Delta_3$, and the size of these cubes do not change w.r.t. location (homogeneity). Thus, the area δ_1 of the interface between two adjacent pixels in x_1 direction is given by $\delta_1 = \Delta_2\Delta_3$. Likewise, the interface areas of the other directions are given by $\delta_2 = \Delta_1\Delta_3$ and $\delta_3 = \Delta_1\Delta_2$, respectively. In order to account for the voxel size properly, a jump with respect to the coordinate direction x_n should get the cost δ_n. This motivates to use $\|x\|_\delta = \sqrt{\sum_n (\delta_n x_n)^2}$ as jump penalty for the direction $x \in \mathbb{R}^N$. Considering the directions $x = a_s$ for $s = 1, \ldots, S$ leads to the new right hand side r' in (5) given by

$$r'_s = \|a_s\|_\delta, \quad \text{for } s = 1, \ldots, S. \tag{7}$$

Note that the left hand side of (5) does not change because it encodes the pixel neighborhood system.

When we solve the linear system (5) using the adapted righthand side r' from (7), it can happen that the solution ω has negative components. This happens for example for the sampling steps $\Delta = (1, 1, 2)$, and its corresponding unit jump penalty vector $\delta = (2, 2, 1)$. To avoid this undesired effect, we impose a non-negativity constraint, and we solve the linear system in a least squares sense; that is, we construct the weights by

$$\min_\omega \ \|T\omega - r\|_2^2, \quad \text{s.t.} \quad \omega \geq 0. \tag{8}$$

This problem can be solved using for example an active set method [55, Chapter 23]. If one of the weights gets equal to zero, it does not contribute to the overall penalty. Therefore, we remove all zero weights and the corresponding finite difference vector. In the above example, $\Delta = (1, 1, 2)$, the weight that corresponds to the orientation $(0, 0, 1)$ is equal to zero. Thus, we remove that from our system $\mathcal{N}$ and end up with 12 finite difference vectors. In Figure 2 we visualize the jump penalties that correspond to the weights computed via (8). The practical advantage of incorporating the non-equidistant sampling steps into the weight design is illustrated in Figure 3 for synthetic data and in Figure 6 for real data. At any rate the computational cost for solving the constrained least squares problem (8) is negligible because its dimension is low and because it has to be performed only once per imaging setup.

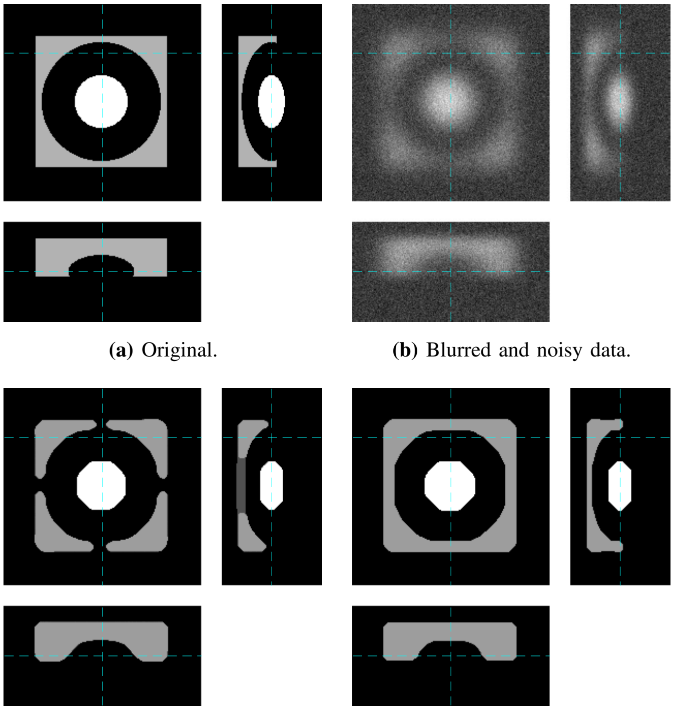
III. A parallel algorithm for segmenting 3D images from linear measurements

Based on the above discretization (1) of the Potts prior, we now develop our strategy for the Potts problem with non-negativity constraints ($\mathscr{P}_\gamma^+$). We first present a splitting based on ADMM, and then provide the solvers for the arising subproblems. We then discuss the computational complexity and the potential for parallelization. The adaption to the unconstrained version ($\mathscr{P}_\gamma$) is straightforward.

A. A splitting approach based on ADMM

Our first step is to reformulate ($\mathscr{P}_\gamma^+$) with the discretization (1) as the constrained optimization problem

$$\begin{aligned} \text{minimize} \quad & \sum_{s=1}^{S} \gamma\omega_s \|\nabla_{a_s} u_s\|_0 + \|Av - f\|_2^2, \\ \text{subject to} \quad & v - u_s = 0, \ \text{for all } 1 \leq s \leq S \\ & u_s \geq 0, \ \text{for all } 1 \leq s \leq S. \end{aligned} \tag{9}$$



(a) Original.

(b) Blurred and noisy data.

(c) Partitioning assuming equidistant sampling steps.

(d) Partitioning with incorporation of non-equidistant sampling.

Fig. 3: Practical effects of the non-equidistant weight design. The image was blurred by a Gaussian kernel of standard deviation 10 and corrupted by Gaussian white noise with standard deviation $\sigma = 0.1$. The image has size $192 \times 192 \times 96$, and we assume half of the x_1, x_2 resolution in x_3 direction, i.e., the sampling steps $\Delta = (1, 1, 2)$. We observe that incorporating the non-equidistant sampling produces a better result than the equidistant one. (See Figure 6 for a comparison on real data.)

Here, u_s denotes the split variable corresponding to the direction a_s, where $1 \leq s \leq S$. We form the augmented Lagrangian $\mathcal{L}_\mu$ of the target functional including the equality constraints which reads as

$$\begin{aligned} &\mathcal{L}_\mu(\{u_s\}_{s=1}^S, v, \{\lambda_s\}_{s=1}^S) \\ &= \sum_{s=1}^S \gamma\omega_s \|\nabla_{a_s} u_s\|_0 + \langle \lambda_s, v - u_s \rangle + \frac{\mu}{2}\|v - u_s\|_2^2 + \|Av - f\|_2^2. \end{aligned} \tag{10}$$

The non-negativity persists as constraints for $u_s \geq 0$, for all $1 \leq s \leq S$. Here, the parameter $\mu > 0$ control how strongly the splitting variables are coupled. The $\lambda_s \in X$, $1 \leq s \leq S$, take the role of Lagrange multipliers. This specific splitting allows us to use ADMM for (10). (See, e.g., [56] for an extensive overview on ADMM.) In each iteration we minimize $\mathcal{L}$ sequentially with respect to $u_1, ..., u_S$, and v and subject to the inequality constraints. Then, we perform gradient ascent steps in the Lagrange multipliers $\lambda_1, \cdots, \lambda_S$.

Then, we obtain by some algebraic manipulation (see appendix) that

$$\arg\min_{u_s \geq 0} \mathcal{L}_\mu = \arg\min_{u_s \geq 0} \frac{2\gamma\omega_s}{\mu} \|\nabla_{a_s} u_s\|_0 + \|u_s - w_s\|_2^2 \tag{11}$$

where $w_s = v + \frac{\lambda_s}{\mu}$. For the minimization with respect to v, we obtain

$$\begin{aligned} &\arg\min_v \mathcal{L}_\mu = \\ &\qquad \arg\min_v \|Av - f\|_2^2 + \frac{\mu S}{2}\left\| v - \frac{1}{S}\sum_{s=1}^S (u_s - \frac{\lambda_s}{\mu}) \right\|_2^2. \end{aligned} \tag{12}$$

Iterating these minimization steps and the gradient ascent steps in the Lagrange multipliers, we obtain the following procedure:

$$\begin{cases} u_1^{k+1} &= \arg\min_{u_1 \geq 0} \frac{2\gamma\omega_1}{\mu_k} \|\nabla_{a_1} u_1\|_0 + \|u_1 - (v^k + \frac{\lambda_1^k}{\mu_k})\|_2^2, \\ &\vdots \\ u_S^{k+1} &= \arg\min_{u_S \geq 0} \frac{2\gamma\omega_S}{\mu_k} \|\nabla_{a_S} u_S\|_0 + \|u_S - (v^k + \frac{\lambda_S^k}{\mu_k})\|_2^2, \\ v^{k+1} &= \arg\min_v \|Av - f\|_2^2 + \frac{\mu S}{2}\|v - \frac{1}{S}\sum_{s=1}^S (u_s^k - \frac{\lambda_s^k}{\mu})\|_2^2 \\ \lambda_s^{k+1} &= \lambda_s^k + \mu_k(v^{k+1} - u_s^{k+1}), \quad \text{for all} \quad 1 \leq s \leq S. \end{cases} \tag{13}$$

We use as coupling parameters an increasing sequence $(\mu_k)_{k\in\mathbb{N}}$ which we will specify later on.

A pseudocode of the procedure is given in Algorithm 1. First we initialize the variable v with a solution of the corresponding minimization problem in (13). In the main loop, we calculate the solutions of the subproblems in the first $S+1$ lines of (13), and then perform a gradient ascent step in the Lagrange multiplier. This loop is iterated until we reach a maximum number of iterations or another stopping criterion such as the maximum squared ℓ^2 distance of the split variables $\|u_s - u_t\|_2^2$ for $s \neq t$. As final result, we return the average of the split variables $u_1, \ldots, u_S$. Next we explain how to solve the subproblems involved in Algorithm 1.

B. Solution of the Tikhonov step

Let us first consider the subproblem in the $(S+1)$-th line of (13). This is a Tikhonov problem of the form

$$v^* = \arg\min_v \|Av - f\|_2^2 + \tau\|v - z\|_2^2.$$

Differentiating the functional with respect to v reveals that a minimizer v^* satisfies

$$2A^*(Av^* - f) + 2\tau(v^* - z) = 0$$

where A^* denotes the adjoint of A (or the transpose if A is given as real-valued system matrix). Resolving this equation gives us the linear system

$$(A^*A + \tau I)\, v^* = A^* f + \tau z, \tag{14}$$

where I denotes the identity. Thus, the computation of v^* reduces to solving a linear system of equations. As the linear system is typically very large we use iterative solvers such as the conjugate gradients methods. To ensure fast convergence we utilize a "warmstart". That is, we use the solution of the previous iteration v^k as initial guess for the computation of v^{k+1}.

Algorithm 1: Proposed algorithm for the inverse Potts model in 3D with non-negativity constraints ($\mathscr{P}_\gamma^+$)

```
Input: Measured data $f \in \mathbb{R}^L$, model parameter $\gamma > 0$
Output: 3D image $u \in \mathbb{R}^{d_1 \times d_2 \times d_3}$
begin
    Initialize $u_s, \lambda_s$ with 0 and $v = \arg\min_z \|Az - f\|_2^2 + \frac{\mu_0 S}{2}\|z\|_2^2$;
    $k \leftarrow 1$;
    repeat
        /* Potts steps                                                    */
        for $s = 1, \dots, S$ do
            $u_s \leftarrow$ Solve Potts problems with data $g = v + \lambda_s/\mu_k$
            line-wise w.r.t. the direction $a_s$, using the
            parameter $\gamma' = \frac{2\omega_s\gamma}{\mu_k}$, see Algorithm 2
            (All univariate solvers can be executed in parallel)
        end
        /* Tikhonov step                                                  */
        $v \leftarrow \arg\min_z \|Az - f\|_2^2 + \frac{\mu_k S}{2}\|z - \frac{1}{S}\sum_{s=1}^S (u_s - \frac{\lambda_s}{\mu_k})\|_2^2$;
        /* Lagrange multiplier update step                                */
        for $s = 1, \dots, S$ do
            $\lambda_s \leftarrow \lambda_s + \mu_k(v - u_s)$;
        end
        $k \leftarrow k + 1$;
    until stopping criterion reached;
    return $u = \frac{1}{S}\sum_{s=1}^S u_s$
end
```

In various setups, the structure of the imaging operator can be exploited for a significant acceleration. In the simplest case of A being the identity we have that

$$v^* = \frac{f + \tau z}{1 + \tau},$$

which is simply a weighted average of f and z. If A describes the (circular) convolution with some convolution kernel ϕ, i.e., $Au = \phi * u$, then we can use Fourier methods. Applying the Fourier transform on both sides of (14) implies that the Fourier transform of v^*, denoted by $\widehat{v^*}$, is given by

$$\widehat{v^*}_j = \frac{\overline{\hat{\phi}_j}\hat{f}_j + \tau\hat{z}_j}{|\hat{\phi}_j|^2 + \tau},$$

where j runs over all indices in the domain of $\widehat{v^*}$.

For vector-valued images, we take advantage of the fact that the Tikhonov problem is separable with respect to the components. That means, we can solve it componentwise for each channel.

C. Solving the Potts subproblems with non-negativity constraints

Now we look at the first S lines of the proposed algorithm (13). There, we solve problems of the form

$$\arg\min_{u \geq 0} \ \gamma' \, \|\nabla_a u\|_0 + \|u - g\|_2^2,$$

with some $\gamma' > 0$, an image $g \in X = \mathbb{R}^{d_1 \times d_2 \ldots \times d_N \times C}$, and a direction $a \in \mathbb{Z}^N \setminus \{0\}$. For notational simplicity, we first consider the case of a single channel image, i.e., $C = 1$. We observe that the problem decomposes into univariate Potts problems along paths indicated by the finite difference vector a. That is, it remains to solve univariate Potts problems with non-negativity constraints of the form

$$\arg\min_{u \in \mathbb{R}^n, u \geq 0} \ \gamma' \, \|\nabla u\|_0 + \|u - h\|_2^2. \tag{15}$$

Here, h is a vector of length n where n is the length of a one dimensional line in direction $a \in \mathbb{R}^3$ with starting pixel $b \in \mathbb{R}^3$. To fix the ideas let us look at the example $a_1 = (1, 0, 0)$. All 1D lines in this directions are parametrized by the offset pixels $b = (1, i_2, i_3)$ for $i_2 = 1, 2 \ldots, d_2$ and $i_3 = 1, 2 \ldots, d_3$. (Recall that the image g is of dimension $d_1 \times d_2 \times d_3$.) For such an offset pixel $b = (1, i_2, i_3)$, the data h is given by $h_j = g_{j, i_2, i_3} = g_{b + j a_1}$. Thus we solve $d_2 \cdot d_3$ times a univariate Potts problem (15) with data h specified by the offset pixels $b = (1, i_2, i_3)$ for $i_2 = 1, 2 \ldots, d_2$ and $i_3 = 1, 2 \ldots, d_3$. For a general direction a, we proceed analogously with respect to the data

$$h_j = g_{b + ja}$$

and all suitable offset pixels b. Note that for the diagonal directions, offset pixels from more than one face of the cube are involved to parametrize all 1D lines through the cube, and the lengths n of the 1D lines depend on the offset pixel and on the dimensions.

It is well known that the unconstrained version of the univariate Potts problem, i.e., $\arg\min_{u \in \mathbb{R}^n} \ \gamma' \, \|\nabla u\|_0 + \|u - h\|_2^2$, can be solved exactly by dynamic programming [15, 57–60]. A quadratic time and linear space algorithm for the univariate Potts problem was proposed by Friedrich et al. [61] which we describe next. To this end, it is convenient to use the Matlab-like notation $h_{l:r} = (h_l, \ldots, h_r)$. We assume that we have already computed minimizers u^l of the Potts functionals $P_\gamma(u) = \gamma' \|\nabla u\|_0 + \|u - h_{1:l}\|_2^2$ associated with the partial data $h_{1:l}$ for each $l = 1, ..., r-1$ and some r smaller than the number n of elements of h. Using this information, the Potts functional associated to data $h_{1:r}$ can be efficiently minimized via

$$\min_{u \in \mathbb{R}^r} P_\gamma(u) = \min_{l = 1, \ldots, r} P_\gamma(u^{l-1}) + \gamma + \epsilon_{l,r}, \tag{16}$$

where $\epsilon_{l,r}$ is the quadratic deviation of the data $h_{l:r}$ from its mean. Here we use the convention that u^0 is an empty vector and $P_\gamma(u^0) = -\gamma$. A corresponding minimizer reads $u^r = (u^{l^*-1}, \mu_{l^*}, \ldots, \mu_{l^*})$ where l^* is a minimizing argument of the righthand side of (16) and μ_{l^*} denotes the mean value of $(h_{l^*}, ..., h_r)$. We obtain a minimizer for full data d by successively computing u^r for each $r = 1, ..., n$. The described method can be implemented in $O(n^2)$ time and $O(n)$ memory using pre-computation of the moments and storing only jump locations [61]. It has been shown that a significant amount of configurations are unreachable and thus can be skipped [34, 62] which improves the practical runtime by a fourfold to fivefold factor.

We next adapt the above algorithm such that it incorporates the non-negativity constraints. The basic dynamic programming principle carries over. To account for the non-negativity constraints we need to compute the quadratic deviation from the best constant approximation $\epsilon^+_{l,r}$ on all intervals. To this end, we observe that the best non-negative constant approximation $\mu^+_{l,r}$ for $h_{l:r}$ is given by

$$\mu^+_{l,r} = \arg\min_{\mu \geq 0} \sum_{i=l}^{r} (\mu - h_i)^2 = \frac{\max(0, \sum_{i=l}^{r} h_i)}{r - l + 1}.$$

This leads to the expression for the approximation error

$$\begin{aligned}
\epsilon^+_{l,r} &= \min_{\mu \geq 0} \sum_{i=l}^{r} (\mu - h_i)^2 \\
&= \sum_{i=l}^{r} \mu^2_{l,r} - 2\mu_{l,r} h_i + h_i^2 \\
&= (r - l + 1)\, \mu^2_{l,r} - 2\mu_{l,r} \sum_{i=l}^{r} h_i + \sum_{i=l}^{r} h_i^2 \\
&= -(r - l + 1)\, \mu^2_{l,r} + \sum_{i=l}^{r} h_i^2 \\
&= -\frac{\max(0, \sum_{i=l}^{r} h_i)^2}{r - l + 1} + \sum_{i=l}^{r} h_i^2
\end{aligned}$$

The value $\epsilon'_{l,r}$ can be evaluated in $O(1)$ when we precompute the cumulative sums $M_r = \sum_{i=1}^{r} h_i$ and $S_r = \sum_{i=1}^{r} h_i^2$. Hence, we obtain

$$\epsilon^+_{l,r} = -\frac{\max(0, M_r - M_{l-1})^2}{r - l + 1} + S_r - S_{l-1}.$$

We provide the complete pseudocode of the solver in Algorithm 2.

The solver can be easily adapted to deal with vector valued data, that is, when $h_i \in \mathbb{R}^C$ for all i. To this end we consider the squared approximation errors $\epsilon^+_{l,r,k}$ on each channel $k = 1, \dots, C$, which are given by $\epsilon^+_{l,r,k} = \min_{\mu \geq 0} \sum_{i=l}^{r} (\mu - h_{i,k})^2$. The necessary value of $\epsilon^+_{l,r}$ then consists of the sum of the channelwise errors, thus $\epsilon^+_{l,r} = \sum_{k=1}^{C} \epsilon^+_{l,r,k}$. We stress that this procedure is different to channelwise solution of the Potts problem. In particular, the presented procedure enforces a common jump set of all channels.

Algorithm 2: Fast solver for the univariate Potts problems with non-negativity constraint (subproblems of type (15))

```
Input: Data vector $h \in \mathbb{R}^n$, model parameter $\gamma' > 0$;
Output: Global minimizer $u^*$ of the univariate Potts problem with
        non-negativity constraint (15)
begin
    /* Find the optimal jump locations */
    $M_0 \leftarrow 0$;
    $S_0 \leftarrow 0$;                                  /* init cumulative moments */
    for $r \leftarrow 1, \dots, n$ do
        $M_r \leftarrow M_{r-1} + h_r$;
        $S_r \leftarrow S_{r-1} + h_r^2$;    /* cumulative first and second moments */
        $P_r \leftarrow S_r - \max(M_r^2/r, 0)$;                  /* init with $\epsilon^+_{1,r}$ */
        $J_r \leftarrow 0$;                         /* init right-most jump location */
        for $l \leftarrow r, r-1 \dots 2$ do
            /* non-neg. mean squared deviation of $h_{l:r}$ */
            $\epsilon^+ \leftarrow S_r - S_{l-1} - \max(M_r - M_{l-1}, 0)^2/(r - l + 1)$;
            if $d + \gamma' > P_r$ then
                Break;                  /* Skip unreachable configurations */
            end
            $p \leftarrow P_{l-1} + \gamma' + \epsilon^+$ ;       /* compute candidate value $p$ */
            if $p \leq P_r$ then
                $P_r \leftarrow p$ ;                  /* store new best Potts value */
                $J_r \leftarrow l - 1$ ;      /* update right-most jump location */
            end
        end
    end
    /* Reconstruct the minimizer from the optimal jump locations */
    $r \leftarrow n$; $l \leftarrow J_r$;
    while $r > 0$ do
        /* set $\mu^+_{l+1,r}$ on partition $l+1, \dots, r$ */
        $u^*_{l+1:r} \leftarrow \max(0, (M_r - M_l)/(r - l))$;
        $r \leftarrow l$; $l \leftarrow J_r$;                           /* go to next jump */
    end
    return $u^*$;
end
```

D. Computational complexity and parallelization

Assume for simplicity that the number of pixels is equal in each direction, i.e., that $d = d_1 = \ldots = d_N$. In each step of (13), we have to solve d^{N-1} Potts problems of size d. With the quadratic complexity of the Potts solver we get the complexity $O(d^{N+1}) = O(M^{(N+1)/N})$ where M denotes the number of pixels. For vector valued data with C channels we get $O(M^{(N+1)/N}C)$. It remains to treat the Tikhonov step which is typically the most time consuming part. Here, a linear system of equations has to be solved which amounts to cubic complexity w.r.t. M in general. Frequently, the special structure of A can be exploited for a significant speedup. If we are in the above described case that we can utilize fast Fourier transforms, then the evaluation of v^* is $O(CM \log M)$ where M is the number of pixels and C the number of channels. The complexity grows linearly in the number of channels for vector-valued data.

The time complexity can be reduced significantly by parallelization. In fact, since the univariate Potts solver acts in each step on nonintersecting lines on the data, we can execute all d^{N-1} Potts solvers in parallel. For example, for $N = 3$ and $d = 100$, $d^2 = 10000$ univariate problems can be solved in parallel in each step. Currently available graphics processing units have typically around thousand processing units. Thus, GPU parallelization can even be almost fully exploited in relatively small problem sizes.

IV. Experimental results

We illustrate the potential of our method for joint deconvolution and segmentation of 3D wide field fluorescence microscopy images. Before that, we discuss the choice of the parameters and our implementation.

A. Algorithmic setup

In our numerical experiments, we used the sampling-step adapted weights of Section II-B with the ordering of the p_s given by (2). We initialized all v by $\arg\min_v \|Av - f\|_2^2 + \frac{\mu_0 S}{2}\|v\|_2^2$ and all other variables u_s, λ_s by 0. We did not observe significant changes in the results for other orderings of a_s or other reasonable initializations. By contrast, the choice of the coupling sequence $(\mu_k)_{k \in \mathbb{N}}$ can have a visible effect on the result and the runtime: the lower the increments of μ_k, the better the results but also the higher the runtime. The condition

$\sum_k \mu_k^{-1/2} < \infty$ ensures that the split variables converge to each other. We omit the proof, as it is a straightforward adaption of the proof for the two-dimensional setup presented in [12]. In practice, slow progressions such as $\mu_k = \mu_0 k^{2.01}$, have the highest potential with respect to reconstruction quality. Thus, if computation time is not critical, we suggest to use slow progressions. Otherwise, one may try a faster progression such as $\mu_k = \mu_0 2^k$. Here we use the progression $\mu_k = \mu_0 1.05^k$ which we found to be a fair tradeoff between quality of the results and computational costs in the present setup; see also [42].

Image dimension	Sequential (CPU)	Parallel (GPU)	Ratio
$50\times\ 50\times 50$	32.0 sec	7.4 sec	4.3
$100\times 100\times 50$	161.3 sec	15.8 sec	10.2
$200\times 200\times 50$	705.5 sec	49.9 sec	14.1
$300\times 300\times 50$	1656.8 sec	97.5 sec	17.0
$400\times 400\times 50$	3080.6 sec	169.2 sec	18.2

TABLE I: Comparison of CPU and GPU runtimes. For the largest image, the parallelized GPU implementation yields a speedup of around 18 times. This way, even large volumes can be processed in reasonable time.

B. Implementation and runtimes

We have created two implementations of our method: a sequential version for the CPU and a parallel version for the GPU.

For the sequential version, we utilize Matlab with C++/MEX files for the time-critical operations; i.e., the univariate Potts solvers. The main advantage of this version is its flexible integration with the Matlab computing framework. It is, for example, possible to supply arbitrary proximal mapping operators as Matlab function handles without changes to the implementation of the algorithm. Furthermore, this implementation improves upon the computation time of our previously proposed Java-based implementation for the two-dimensional Potts problem (with $A = id$) [34][1] by a factor two.

The parallel version has been implemented for the GPU, based on C++ and the CUDA framework. The GPU significantly decreases the computation time provided that the proximal mapping can be evaluated efficiently on the GPU. In the present work, the imaging operator possess an efficient implementation based on the CUDA FFT library cuFFT. In our experiment (Table I), we processed 3D images of different sizes with our sequential CPU and our parallel GPU version. Benchmarking was performed using a workstation with 32GB RAM, a core i7-4770k processor with 3.5GHz and a GTX Titan Black GPU. The software used was Matlab 2015b, CUDA 7.0 and gcc 4.7.3. From Table I, we observe that the parallel version leads to an eighteen-fold speedup. This potential is almost fully exploited even for moderate image dimensions of $200 \times 200 \times 50$. Large volumes of dimension $400\times400\times50$ can be processed in less than three minutes. The tradeoff is that the GPU implementation requires more effort to be adapted to alternate imaging setups; e.g. changing the proximal mapping requires implementation of the operator in CUDA and recompilation of the entire algorithm.

The core algorithm has two parts which benefit massively from GPU acceleration, i.e. solving the 1D Potts problems in parallel, and, to a lower degree, the calculation of the weighted mean solution from all 1D results. We implemented the solver for the 1D Potts problem as a custom kernel. The calculation of the weighted mean solution is implemented using cuBLAS, a GPU accelerated variant of the basic linear algebra library. This part does not require any custom kernels. The RAM requirements on the GPU scale with $O(MS)$, where M is the total number of pixels of the image, and where S denotes the number of directions used for the discretization of the Potts prior. The factor S is due to the fact that the implementation solves the Potts problems for multiple directions at once in an asynchronous fashion; this allows us to avoid latency from kernel calls and results in higher speedup for small-to-medium sized problems at the cost of increased RAM requirement. In practice, we observed that images with a size of up to GPURAM divided by 20 could be processes without issues.

We compared the results produced by the GPU and the CPU implementation of our algorithm. For the 1D solvers, we did not observe any differences in the resulting 1D segmentations. Further, we did not observe any difference when segmenting images for the case A = id. When segmenting from blurred data, we observed that about 0.1 percent of all voxels ended up in different segments when comparing the GPU and the CPU implementation. Hence, there is a minor difference between the results obtained on the GPU and on the CPU. We believe that this is due to different rounding effects.

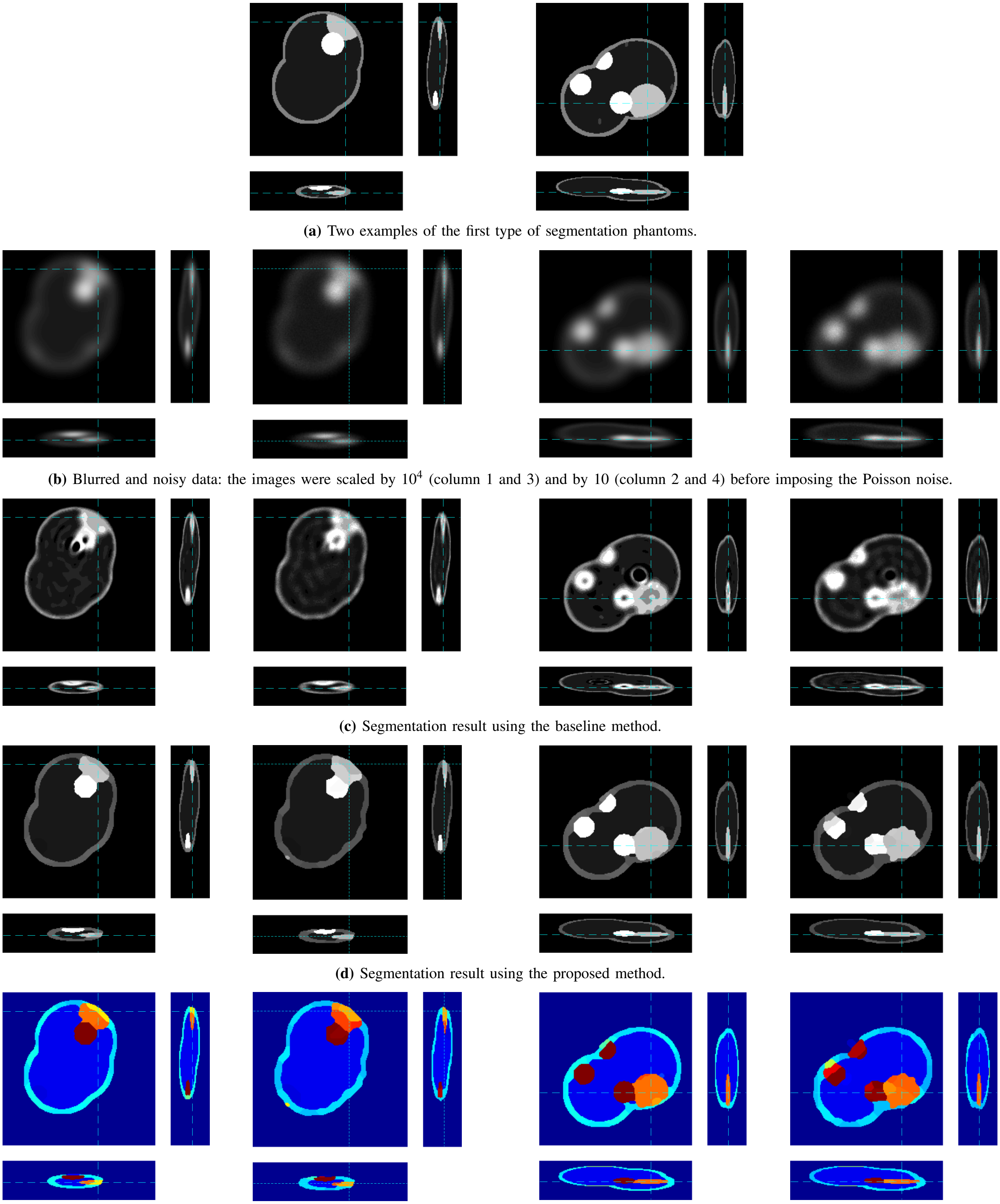
C. Simulation study

We performed a simulation study for joint reconstruction and segmentation from blurred data. The experimental setup is as follows. As test data, we created two series of structurally different segmentation phantoms of size $200 \times 200 \times 50$. The first series consists of 50 cell-type phantoms. We used the function “phantom3d”[2] to create three overlapping ellipsoids to represent a cell body, a boundary of 11 pixels width to represent a cell boundary and complemented it with a few ellipsoids representing organelles. Cell body, boundary and organelles were each given distinct gray values. Two typical examples of the generated phantoms are shown in Figure 4. Each of the 50 phantoms has different randomly shifted ellipsoid centers. The second series of phantoms consists of 50 images of size $200\times200\times50$ containing randomly rotated and shifted versions of geometric primitives. Two typical examples are depicted in Figure 5. All phantoms have a dynamic range in $[0, 100]$. We blurred the phantoms using a Gaussian point spread function. Then, we corrupted the resulting image by Poisson noise. We produced different image qualities (or noise levels) by simulating different photon counts. To this end, we scaled the blurred images by the factor 10^p, for $p = 0, ..., 4$, before imposing the Poisson noise. Note that a higher scaling factor means higher data quality.

To measure the segmentation quality of the computed multilabel segmentations, we use the Rand index (RI) [63, 64],

[1] Implementation of the method in [34] is available at http://pottslab.de.

[2] Available from the Matlab File Exchange.



(a) Two examples of the first type of segmentation phantoms.

(b) Blurred and noisy data: the images were scaled by 10^4 (column 1 and 3) and by 10 (column 2 and 4) before imposing the Poisson noise.

(c) Segmentation result using the baseline method.

(d) Segmentation result using the proposed method.

(e) Results of (d) visualized in the jet colormap.

Fig. 4: Qualitative results of the simulation study for the first type of phantoms: The first two and the last two columns show the segmentation results for the two example phantoms in (a). The first and the third column have a higher data quality (scaling factor 10^4), the second and fourth column have a lower quality (scaling factor 10). The segmentations in (c) and (d) are encoded as piecewise constant gray-level images where the gray-values correspond to the estimated mean value on each segment. In (e), the results of (d) are shown in pseudocolors. We observe that the baseline method captures the cell boundary well but also produces fuzzy results in the body and the organelles. The proposed method better recovers the organelles; it has a moderate oversegmentation at the boundary. (See Table II for quantitative results.)

Image scaling (higher number	Rand index	
means higher data quality)	Baseline method	Proposed method
10^4	0.9301 ± 0.0052	0.9645 ± 0.0031
10^3	0.9286 ± 0.0052	0.9642 ± 0.0036
10^2	0.9221 ± 0.0063	0.9610 ± 0.0036
10^1	0.9152 ± 0.0058	0.9500 ± 0.0040
10^0	0.8872 ± 0.0077	0.9265 ± 0.0049

TABLE II: Quantitative results of the simulation study for the first class of phantoms: The table shows the average Rand index over all simulated 50 phantoms along with the standard deviation. The rows are indexed by the scaling of the data where a higher scaling means higher data quality (corresponding to a lower noise level). The proposed method consistently yields a higher Rand index. (See Figure 4 for qualitative results.)

Image scaling (higher number	Rand index	
means higher data quality)	Baseline method	Proposed method
10^4	0.9197 ± 0.0105	0.9896 ± 0.0049
10^3	0.9193 ± 0.0105	0.9882 ± 0.0051
10^2	0.9196 ± 0.0105	0.9812 ± 0.0050
10^1	0.9182 ± 0.0111	0.9813 ± 0.0051
10^0	0.9097 ± 0.0124	0.9589 ± 0.0115

TABLE III: Quantitative results of the simulation study for the second class of phantoms. (See Table II for the description and Figure 5 for qualitative results.)

which we briefly explain in the following. (Note that the frequently used Dice index is limited to two-label segmentations.) Let $X = \{x_1, ..., x_N\}$ be a given set of points and let Y and Y' be two partitionings of this set. (In our case, X is the set of pixels, and Y and Y' are the partitionings given by the ground truth and by the result of the employed method, respectively.) The Rand index (RI) is defined by the expression $\mathrm{RI}(Y, Y') = \binom{N}{2} \sum_{i<j}^{N} t_{ij}$. Here t_{ij} is equal to one if there exist k and k' such that both x_i and x_j are in both Y_k and $Y'_{k'}$, or if x_i is in both Y_k and $Y'_{k'}$ while x_j is in neither Y_k or $Y'_{k'}$. Otherwise, t_{ij} is set to zero. The Rand index is bounded from above by 1; a higher value means a better match. For the evaluation of the Rand index we used the implementation of K. Wang[3].

As baseline approach, we use a two stage method. We first performed a Richardson-Lucy deconvolution [1, 2] which is suitable in the presence of Poisson noise. We optimized the number of Richardson-Lucy iterations such that the best result with respect to the mean squared error to the original image is achieved. On this result, we applied a k-means clustering where we optimized the k-means parameter K with respect to Rand index to the ground truth segmentation. Here, the scan range for the parameter K was set from half to twice the true number of labels of the ground truth. For the baseline approach, the parameters were optimized for each phantom and each noise realization individually to get as good results as possible. For both stages, we used the reference implementations of Matlab. For the proposed method, we optimized the model parameter γ and the initial coupling parameter μ_0 with respect to the Rand index. For the proposed method, for each of the five noise levels, a single image of the first class of phantoms was used to adjust the hyperparameter. Then, depending on the noise level, these parameters were applied to each of the phantoms. (They may be seen as the validation set; the "training" image was not included in this set.) For the baseline method, hyperparameters were scanned for each of the 50 phantom image individually, i.e. each image was used both for test and validation, creating a bias advantageous for the performance of the baseline method.

The outcome of the first series of phantoms is reported in Figure 4 and in Table II. Figure 4 exemplarily shows the outcome of the experiment for two different phantoms and two different noise levels. In Table II, we report the quantitative results averaged over all 50 cell-type phantoms. The analogous results for the geometric primitive phantoms are given in Figure 5 and in Table III. Qualitatively, we observe that the proposed method produces more contiguous segments than the baseline method. Quantitatively, we observe that the proposed method provides a higher Rand index than the baseline method consistently for all noise levels. This indicates an improved segmentation quality. Both methods tend to oversegmentation, i.e., the number of produced segments is larger than the number of true segments, which can be addressed by post-processing.

D. *Results on 3D widefield fluorescence microscopy data*

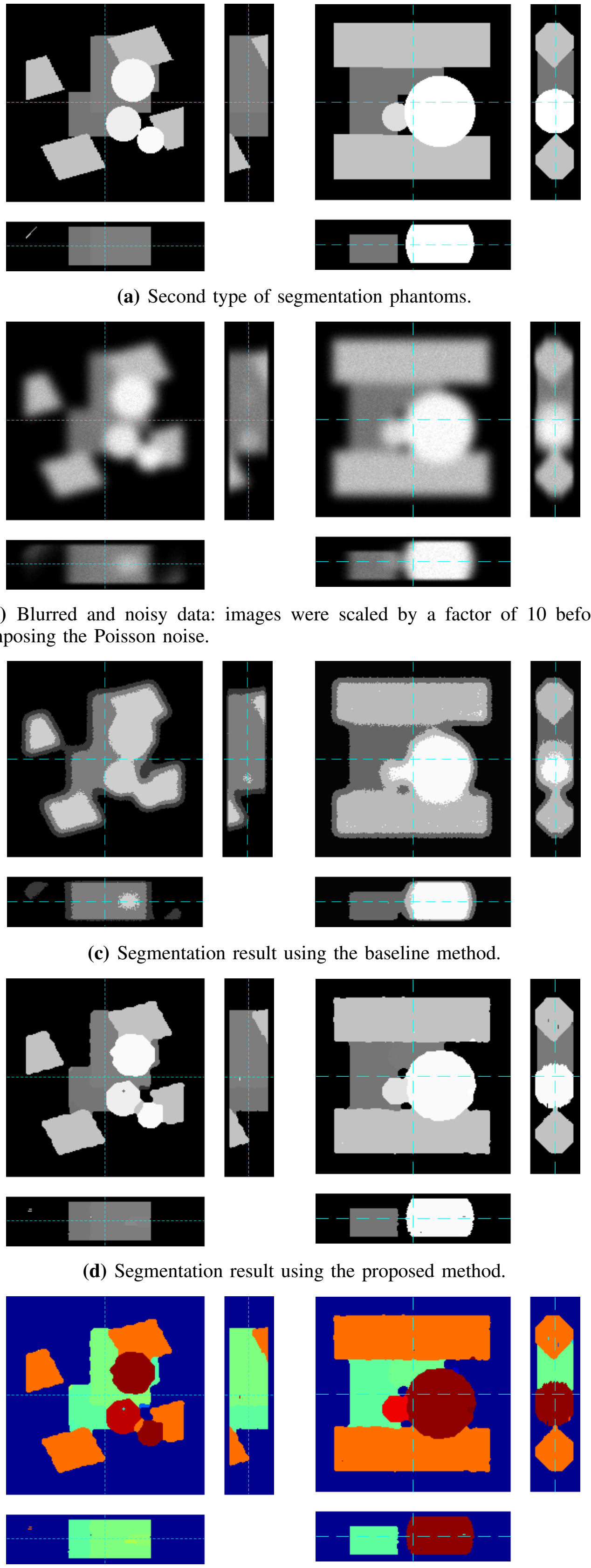
Next we show experimental results on real 3D widefield fluorescence microscopy data. The data was acquired using an Olympus CellR widefield microscope [65].[4] The point spread function (PSF) of the image acquisition setup is shown in Figure 6. It exhibits good localization in the x_1-x_2 plane but has a wide spread in the x_3 direction. The sampling steps are given by $\Delta_1 = \Delta_2 = 64.5$ nm and $\Delta_3 = 160$ nm. For the baseline approach we used 30 Richardson–Lucy iterations, and performed a k-means clustering where we chose the number of labels K to match the expected number of labels in the image. (For further comparison, results for other deconvolution methods on the present data set can be found in [65–67].) For the proposed method, we used the initial coupling parameter $\mu_0 = 10^{-6}$ as proposed in [12]. The model parameter γ was chosen depending on the experiment as indicated below.

1) Fluorescence bead: The first real data set of size 256^3 is the image of a fluorescence bead with known diameter $2.5\,\mu$m. We have manually chosen the γ-parameter to capture the topological structure of the hollow sphere. Figure 6 shows the segmentation of the bead from the linear measurements. We observe that the baseline approach overestimates the size of the bead with respect to the x_3 direction. The output of the proposed method matches significantly better with the true dimensions of the bead. We observe that it slightly oversegments the bead (two segments instead of one). However, such oversegmentation can typically be resolved relatively easy by region merging.

2) C-elegans embryo: The second real data set is the microscopic image of a C-elegans embryo using a DAPI

[3] Available from the Matlab File Exchange.

[4] The data set is freely available at http://bigwww.epfl.ch/deconvolution/.



(a) Second type of segmentation phantoms.

(b) Blurred and noisy data: images were scaled by a factor of 10 before imposing the Poisson noise.

(c) Segmentation result using the baseline method.

(d) Segmentation result using the proposed method.

(e) Results of (d) visualized in the jet colormap.

Fig. 5: Qualitative results of the simulation study for the second class of phantoms (compare Figure 4 and Table III). For the baseline method, the parameters have been optimized on an image-by-image basis, whereas for the proposed method, the parameters already used for the first type of phantoms have been employed without adaption.

fluorescent stain. We downsampled the original image size by the factor two in each dimension to obtain an image of size 337×357×56. We have manually chosen the γ-parameter such that the white organelles in the cells are captured as individual segments and that the main cell body is only moderately oversegmented. Figure 7 shows the result of this experiment. The baseline approach captures most nuclei sharply but it has some incomplete segments and has partially fuzzy boundaries in the x_3 direction. The proposed method has better localized boundaries. Again, we observe a slight oversegmentation in the cell body. The runtime amounts to around 43 seconds for the baseline method, to around 29 minutes for the sequential CPU implementation, and to around 291 seconds using the parallel GPU implementation.

V. Conclusion and future research

We have developed a fast algorithm for segmenting 3D images from linear measurements based on the Potts model; in particular, we addressed the specific challenges of the 3D setup. First, we proposed a new design for discretizations of the jump penalty to address the issue of non-cubic grids frequently appearing in 3D. Secondly, we presented a highly parallel implementation on the GPU which allowed us to process even large volumes in reasonable computation time. Thirdly, to take the non-negativity of many physical quantities into account, we incorporated corresponding constraints into the model and into the proposed algorithmic scheme. We have illustrated the potential of our method for joint image deconvolution and segmentation of 3D widefield microscopy images. We note that the method can be adapted to other imaging setups such as x-ray computed tomography and photoacoustic tomography. The feasibility condition is that we can compute the proximal mapping of the data term in reasonable time.

Future directions of research include the adaption to further imaging setups such as magnetic particle imaging and the adaption to the (piecewise smooth) Mumford-Shah functional as in [68].

Acknowledgement

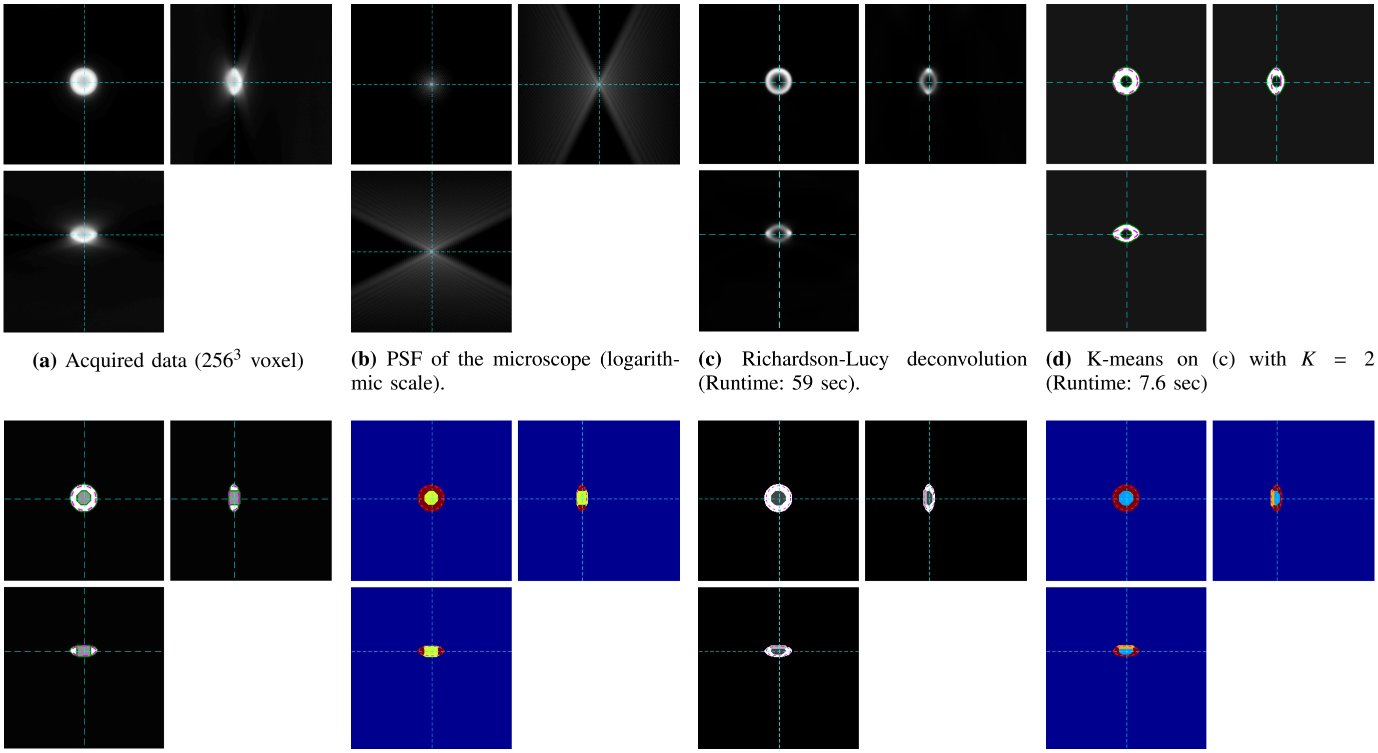
M. Storath and A. Weinmann acknowledge the support by the German Research Foundation (DFG grant STO1126/2-1, WE5886/4-1). M. Unser was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no. 267439. The authors would like to thank Daniel Sage and Ferréol Soulez for valuable discussions on deconvolution of widefield microscopy images.

Appendix

A. Proof of Proposition 1

First let $x \in \mathcal{N}$ be one of the coordinate directions, say $x = a_1 = (1, 0, 0)$. Then, the righthand side of (4) is given by $\|x\|_2 = 1$. For the lefthand side, we obtain

$$\begin{aligned}\|x\|_{\mathcal{N}} &= \sum_{s=1}^{S} \omega_s |\langle x, a_s\rangle| = \sum_{s=1}^{S} \omega_s |(a_s)_1| \\ &= \omega_1 + \omega_4 + \omega_5 + \omega_6 + \omega_7 + \omega_{10} + \omega_{11} + \omega_{12} + \omega_{13}\end{aligned}$$



(a) Acquired data (256^3 voxel)

(b) PSF of the microscope (logarithmic scale).

(c) Richardson-Lucy deconvolution (Runtime: 59 sec).

(d) K-means on (c) with $K = 2$ (Runtime: 7.6 sec)

(e) Proposed method using equidistant weight design ($\gamma = 0.06$, Runtime: 142 sec)

(f) Result of (e) visualized in the jet colormap.

(g) Proposed method using weights adapted to non-equidistant sampling ($\gamma = 0.06$, Runtime: 293 sec)

(h) Result of (g) visualized in the jet colormap.

Fig. 6: First real data experiment: Joint reconstruction and segmentation of a ball-shaped fluorescence bead with diameter $2.5\,\mu$m. The sampling steps are 64.5 nm with respect to the x_1 and x_2 axes and 160 nm with respect to the x_3 axis. The true size of the bead is indicated by dashed lines. (Note that the ball-shaped bead becomes ellipsoidal in the displayed pixel coordinates due to the non-cubic grid.) We observe that the baseline approach overestimates the diameter in the axial direction. The proposed approach recovers the true dimensions of the bead significantly better. The proposed weight design adapted to the non-equidistant sampling gives an improved resolution in x_3-direction compared to the equidistant design.

By symmetry, we have $\omega_1 = \ldots = \omega_3$, $\omega_4 = \ldots = \omega_9$, and $\omega_{10} = \ldots = \omega_{13}$. Plugging this into (4) leads to the condition

$$\omega_1 + 4\omega_4 + 4\omega_{10} = 1.$$

Note that the other finite difference vectors in coordinate directions, a_2 and a_3, lead to the same equation. Similarly, the planar diagonals, such as $x = (1, 1, 0)$, lead to the equation

$$2\omega_1 + 6\omega_4 + 4\omega_{10} = \sqrt{2},$$

and the volume diagonals (e.g., $x = (1, 1, 1)$) yield

$$3\omega_1 + 6\omega_4 + 6\omega_{10} = \sqrt{3}.$$

Solving this linear system with respect to ω_1, ω_4 and ω_{10} gives us the weights claimed in (6).

B. Derivation of the ADMM iteration

The first step is to rewrite the augmented Lagrangian (10) in its scaled form; see [56]. To this end, we rewrite

$$\langle \lambda_s, v - u_s \rangle + \frac{\mu}{2}\|v - u_s\|_2^2 = \frac{\mu}{2}\|v - u_s + \tfrac{\lambda_s}{\mu}\|_2^2 - \frac{1}{2\mu}\|\lambda_s\|_2^2, \quad (17)$$

so that

$$\begin{aligned}&\mathcal{L}_\mu(\{u_s\}_{s=1}^S, v, \{\lambda_s\}_{s=1}^S)\\ &= \sum_{s=1}^S \gamma\omega_s \|\nabla_{a_s} u_s\|_0 + \frac{\mu}{2}\|v - u_s + \tfrac{\lambda_s}{\mu}\|_2^2 - \frac{1}{2\mu}\|\lambda_s\|_2^2 + \|Av - f\|_2^2.\end{aligned} \quad (18)$$

Then, as the last two terms do not depend on u_s minimization with respect to u_s gives (11).

To bring $\arg\min_v \mathcal{L}_\mu(\{u_s\}_{s=1}^S, v, \{\lambda_s\}_{s=1}^S)$ into a convenient form we combine the second S terms as

$$\sum_{s=1}^S \frac{\mu}{2}\|v - u_s + \tfrac{\lambda_s}{\mu}\|_2^2 = \frac{\mu S}{2}\left\| v - \frac{1}{S}\sum_{s=1}^S (u_s - \frac{\lambda_s}{\mu})\right\|_2^2 + C$$

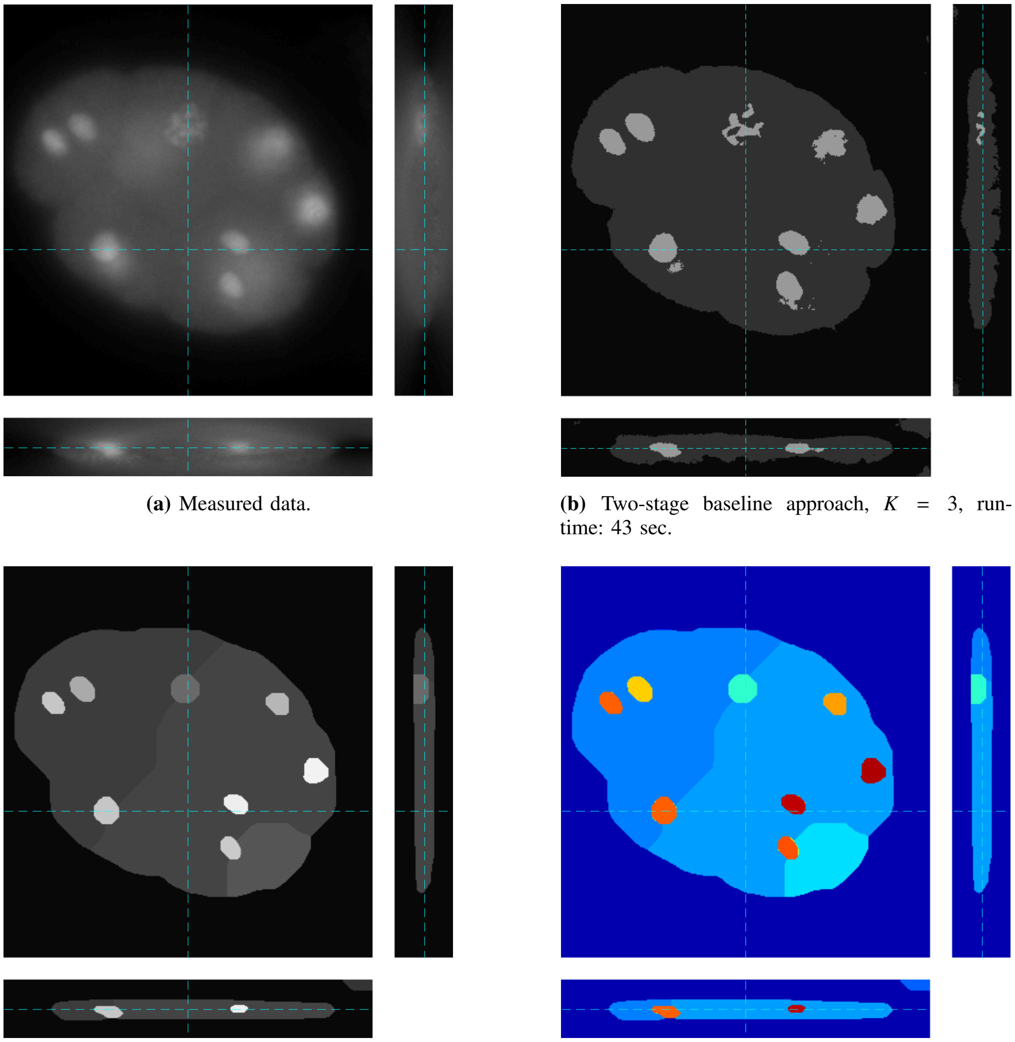
with a constant $C \in \mathbb{R}$ that does not depend on a. Here, we used the fact that

$$\textstyle\sum_i x_i(a - b_i)^2 = (\sum_i x_i)(a - \frac{\sum_i b_i x_i}{\sum_i x_i})^2 + C'$$

for $a, b_1, ..., b_N \in \mathbb{R}$ and $x_1, ..., x_N > 0$ and a constant $C' \in \mathbb{R}$ that does not depend on a, see e.g. [12]. Minimizing with respect to v yields (12).

References

[1] W. Richardson, “Bayesian-based iterative method of image restoration,” *Journal of the Optical Society of America*, vol. 62, no. 1, pp. 55–59, 1972.



(a) Measured data.

(b) Two-stage baseline approach, $K = 3$, runtime: 43 sec.

(c) Proposed approach, $\gamma = 0.1$, runtime: 291 sec.

(d) Result of (c) visualized in the jet colormap.

Fig. 7: Second real data experiment: Deconvolution and segmentation of the widefield microscopy image of a C-Elegans embryo. The data (a) was acquired using the same setup as in Figure 6. (b) The baseline approach captures parts of the nuclei well, but it has also incomplete segments and fuzzy boundaries in x_3 direction. (c,d) The proposed method produces more regular and complete segments. We observe a moderate over-segmentation in the cell body.

[2] L. Lucy, "An iterative technique for the rectification of observed distributions," *The Astronomical Journal*, vol. 79, p. 745, 1974.

[3] N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, "Richardson-Lucy algorithm with total variation regularization for 3D confocal microscope deconvolution," *Microscopy research and technique*, vol. 69, no. 4, pp. 260–266, 2006.

[4] C. Vonesch and M. Unser, "A fast thresholded Landweber algorithm for wavelet-regularized multidimensional deconvolution," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 539–549, 2008.

[5] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J.-C. Olivo-Marin, and C. Zimmer, "Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces," *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1396–1410, 2005.

[6] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[7] L. Rondi and F. Santosa, "Enhanced electrical impedance tomography via the Mumford–Shah functional," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 6, pp. 517–538, 2001.

[8] L. Rondi, "On the regularization of the inverse conductivity problem with discontinuous conductivities," *Inverse Problems and Imaging*, vol. 2, no. 3, pp. 397–409, 2008.

[9] R. Ramlau and W. Ring, "A Mumford–Shah level-set approach for the inversion and segmentation of X-ray tomography data," *Journal of Computational Physics*, vol. 221, no. 2, pp. 539–557, 2007.

[10] ——, "Regularization of ill-posed Mumford–Shah models with perimeter penalization," *Inverse Problems*, vol. 26, no. 11, p. 115001, 2010.

[11] G. Paul, J. Cardinale, and I. Sbalzarini, "Coupling image restoration and segmentation: a generalized linear model/Bregman perspective," *International Journal of Computer Vision*, vol. 104, no. 1, pp. 69–93, 2013.

[12] M. Storath, A. Weinmann, J. Frikel, and M. Unser, "Joint image reconstruction and segmentation using the Potts model," *Inverse Problems*, vol. 31, no. 2, p. 025003, 2015.

[13] R. Potts, "Some generalized order-disorder transformations," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, no. 1, pp. 106–109, 1952.

[14] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[15] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[16] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l_0 gradient minimization," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, p. 174, 2011.

[17] R. Nguyen and M. Brown, "Fast and effective L_0 gradient minimization by region fusion," in *IEEE International Conference on Computer Vision*

(ICCV), 2015, pp. 208–216.

[18] O. Veksler, "Efficient graph-based energy minimization methods in computer vision," Ph.D. dissertation, Cornell University, 1999.

[19] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[20] Z. Kato and T.-C. Pong, "A Markov random field image segmentation model for color textured images," *Image and Vision Computing*, vol. 24, no. 10, pp. 1103–1114, 2006.

[21] J. Yuan, E. Bae, X.-C. Tai, and Y. Boykov, "A continuous max-flow approach to Potts model," in *ECCV*, 2010, pp. 379–392.

[22] L. Ambrosio and V. Tortorelli, "Approximation of functional depending on jumps by elliptic functional via Γ-convergence," *Communications on Pure and Applied Mathematics*, vol. 43, no. 8, pp. 999–1036, 1990.

[23] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[24] C. Zimmer and J.-C. Olivo-Marin, "Coupled parametric active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1838–1842, 2005.

[25] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *International Journal of Computer Vision*, vol. 72, no. 2, pp. 195–215, 2007.

[26] T. Pock, A. Chambolle, D. Cremers, and H. Bischof, "A convex relaxation approach for computing minimal partitions," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 810–817.

[27] A. Chambolle, D. Cremers, and T. Pock, "A convex approach to minimal partitions," *SIAM Journal on Imaging Sciences*, vol. 5, no. 4, pp. 1113–1158, 2012.

[28] J. Lellmann and C. Schnörr, "Continuous multiclass labeling approaches and algorithms," *SIAM Journal on Imaging Sciences*, vol. 4, no. 4, pp. 1049–1096, 2011.

[29] E. Strekalovskiy, A. Chambolle, and D. Cremers, "A convex representation for the vectorial Mumford-Shah functional," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1712–1719.

[30] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[31] A. Drory, C. Haubold, S. Avidan, and F. A. Hamprecht, "Semi-global matching: a principled derivation in terms of message passing," in *Pattern Recognition*. Springer, 2014, pp. 43–53.

[32] M. Fornasier and R. Ward, "Iterative thresholding meets free-discontinuity problems," *Foundations of Computational Mathematics*, vol. 10, no. 5, pp. 527–567, 2010.

[33] X. Cheng, M. Zeng, and X. Liu, "Feature-preserving filtering with L_0 gradient minimization," *Computers & Graphics*, vol. 38, pp. 150–157, 2014.

[34] M. Storath and A. Weinmann, "Fast partitioning of vector-valued images," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1826–1852, 2014.

[35] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, 2001, pp. 105–112.

[36] A. Dufour, R. Thibeaux, E. Labruyere, N. Guillen, and J.-C. Olivo-Marin, "3-D active meshes: fast discrete deformable models for cell tracking in 3-D time-lapse microscopy," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1925–1937, 2011.

[37] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "An algorithm for minimizing the Mumford-Shah functional," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 1133–1140.

[38] E. Klann, R. Ramlau, and W. Ring, "A Mumford-Shah level-set approach for the inversion and segmentation of SPECT/CT data," *Inverse Problems and Imaging*, vol. 5, no. 1, pp. 137–166, 2011.

[39] E. Klann and R. Ramlau, "Regularization properties of Mumford-Shah-type functionals with perimeter and norm constraints for linear ill-posed problems," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 413–436, 2013.

[40] M. Jiang, P. Maass, and T. Page, "Regularizing properties of the Mumford-Shah functional for imaging applications," *Inverse Problems*, vol. 30, no. 3, p. 035007, 2014.

[41] B. Alexeev and R. Ward, "On the complexity of Mumford–Shah-type regularization, viewed as a relaxed sparsity constraint," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2787–2789, 2010.

[42] M. Storath, A. Weinmann, and L. Demaret, "Jump-sparse and sparse recovery using Potts functionals," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3654–3666, 2014.

[43] L. Bar, N. Sochen, and N. Kiryati, "Variational pairing of image segmentation and blind restoration," in *ECCV 2004*. Springer, 2004, pp. 166–177.

[44] M. Fornasier, R. March, and F. Solombrino, "Existence of minimizers of the Mumford-Shah functional with singular operators and unbounded data," *Annali di Matematica Pura ed Applicata*, vol. 192, no. 3, pp. 361–391, 2013.

[45] M. Artina, M. Fornasier, and F. Solombrino, "Linearly constrained nonsmooth and nonconvex minimization," *SIAM Journal on Optimization*, vol. 23, no. 3, pp. 1904–1937, 2013.

[46] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1107–1114.

[47] A. Weinmann and M. Storath, "Iterative Potts and Blake-Zisserman minimization for the recovery of functions with discontinuities from indirect measurements," *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2176, p. 20140638, 2015.

[48] J. Kim, A. Tsai, M. Cetin, and A. Willsky, "A curve evolution-based variational approach to simultaneous image restoration and segmentation," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, 2002, pp. I–109.

[49] E. Klann, "A Mumford-Shah-like method for limited data tomography with an application to electron tomography," *SIAM Journal on Imaging Sciences*, vol. 4, no. 4, pp. 1029–1048, 2011.

[50] J. Helmuth and I. Sbalzarini, "Deconvolving active contours for fluorescence microscopy images," in *Advances in Visual Computing*. Springer, 2009, pp. 544–553.

[51] A. Chambolle, "Finite-differences discretizations of the Mumford-Shah functional," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 33, no. 02, pp. 261–288, 1999.

[52] C. Preza and J.-A. Conchello, "Depth-variant maximum-likelihood restoration for three-dimensional fluorescence microscopy," *Journal of the Optical Society of America A*, vol. 21, no. 9, pp. 1593–1601, 2004.

[53] J. Kim, S. An, S. Ahn, and B. Kim, "Depth-variant deconvolution of 3D widefield fluorescence microscopy using the penalized maximum likelihood estimation method," *Optics Express*, vol. 21, no. 23, pp. 27 668–27 681, 2013.

[54] M. Cheney, D. Isaacson, and J. C. Newell, "Electrical impedance tomography," *SIAM review*, vol. 41, no. 1, pp. 85–101, 1999.

[55] C. Lawson and R. Hanson, *Solving least squares problems*. SIAM, 1974, vol. 161.

[56] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[57] A. Chambolle, "Image segmentation by variational methods: Mumford and Shah functional and the discrete approximations," *SIAM Journal on Applied Mathematics*, vol. 55, no. 3, pp. 827–863, 1995.

[58] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 17, 1985, pp. 137–154.

[59] G. Winkler and V. Liebscher, "Smoothers for discontinuous signals," *Journal of Nonparametric Statistics*, vol. 14, no. 1-2, pp. 203–222, 2002.

[60] J. Kleinberg and É. Tardos, *Algorithm design*. Pearson Education India, 2006.

[61] F. Friedrich, A. Kempe, V. Liebscher, and G. Winkler, "Complexity penalized M-estimation," *Journal of Computational and Graphical Statistics*, vol. 17, no. 1, pp. 201–224, 2008.

[62] R. Killick, P. Fearnhead, and I. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[63] W. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[64] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[65] A. Griffa, N. Garin, and D. Sage, “Comparison of deconvolution software in 3D microscopy: A user point of view—-Part 1,” *GIT Imaging & Microscopy*, vol. 12, pp. 43–45, 2010.

[66] F. Soulez, L. Denis, Y. Tourneur, and E. Thiébaut, “Blind deconvolution of 3D data in wide field fluorescence microscopy,” in *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2012, pp. 1735–1738.

[67] D. Sage, L. Donati, F. Soulez, D. Fortun, G. Schmit, A. Seitz, R. Guiet, C. Vonesch, and M. Unser, “Deconvolutionlab2: An open-source software for deconvolution microscopy,” *Methods*, 2017.

[68] K. Hohm, M. Storath, and A. Weinmann, “An algorithmic framework for Mumford-Shah regularization of inverse problems in imaging,” *Inverse Problems*, vol. 31, no. 11, p. 115011, 2015.