Simultaneous tomographic reconstruction and segmentation with class priors

Romanov, Mikhail; Dahl, Anders Bjorholm; Dong, Yiqiu; Hansen, Per Christian

Published in:
Inverse Problems in Science and Engineering

Link to article, DOI:
10.1080/17415977.2015.1124428

Publication date:
2015

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Romanov, M., Dahl, A. B., Dong, Y., & Hansen, P. C. (2015). Simultaneous tomographic reconstruction and segmentation with class priors. *Inverse Problems in Science and Engineering*, *24*(8), 1432-1453. https://doi.org/10.1080/17415977.2015.1124428

To appear in *Inverse Problems in Science and Engineering*
Vol. 00, No. 00, Month 20XX, 1–22

Simultaneous Tomographic Reconstruction and Segmentation with Class Priors

Mikhail Romanov, Anders Bjorholm Dahl, Yiqiu Dong, and Per Christian Hansen*

Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Kgs. Lyngby, Denmark

(*Received 00 Month 20XX; final version received 00 Month 20XX*)

We consider tomographic imaging problems where the goal is to obtain both a reconstructed image and a corresponding segmentation. A classical approach is to first reconstruct and then segment the image; more recent approaches use a discrete tomography approach where reconstruction and segmentation are combined to produce a reconstruction that is identical to the segmentation. We consider instead a hybrid approach that simultaneously produces both a reconstructed image and a segmentation. We incorporate priors about the desired classes of the segmentation through a Hidden Markov Measure Field Model, and we impose a regularization term for the spatial variation of the classes across neighboring pixels. We also present an efficient implementation of our algorithm based on state-of-the-art numerical optimization algorithms. Simulation experiments with artificial and real data demonstrate that our combined approach can produce better results than the classical two-step approach.

Keywords: tomographic reconstruction, segmentation, regularization, numerical optimization, Hidden Markov Measure Field Models

AMS Subject Classifications: 65F22, 65K10

1. Introduction

In computed tomography (CT) it is often the case that the reconstructed 2D or 3D image is also used as an intermediate result in order to arrive at a segmentation of the reconstructed object. The purpose of the segmentation is, e.g., to separate an object from the background [1] or to identify specific objects or regions [2], and this has many applications in medical imaging and in non-destructive testing in materials science. The classical approach is first to produce a 2D or 3D reconstruction, followed by segmentation of this image. Both steps are likely to introduce errors and artifacts, and the errors in the reconstructed image usually propagate to the segmentation.

In order to avoid this one can try to combine the image reconstruction and segmentation in a single reconstruction model (leading to a more complex problem). One idea is to allow only a small number of (known or unknown) pixel values, leading to the problem of discrete tomography – see [2], [3], [4] for examples. Another idea is to use segmentation principles to identify objects with known pixel values in order to obtain sharper edges around these objects [1]. In both cases, the segmented image is identical to the reconstructed image.

In this work we take a different approach, which we refer to as *Simultaneous Reconstruction and Segmentation* (SRS), where we simultaneously produce both a reconstructed

*Corresponding author. Email: pcha@dtu.dk image and a segmentation. This idea was originally proposed by Ramlau and Ring [5] who demonstrate that the combined approach is able to produce good reconstructions with sharp boundaries of the objects. Their CT algorithm, which was later generalized to SPECT in [6], simultaneously reconstructs and segments directly from the data. The same approach was applied to CT problems with limited data in [7]. Van de Sompel and Brady [8] proposed a related idea using a Hidden Markov Measure Field Model (HMMFM) [9] that "learns" information about the different types of objects or phases from the given data during the SRS process. Other papers that describe the use of a hidden Markov field are [10] and [11].

Instead of "learning" or estimating information about the segmentation classes during the reconstruction phase, we want to use this information as a prior. We therefore propose a variant of the above method where we explicitly specify prior information about the different classes (i.e., about different types of phases), to be used in the segmentation via the HMMFM.

There is a variety of computational algorithms for tomographic reconstruction that use regularization in order to deal with the difficulties of the underlying inverse problem [12]. Filtered back projection and its extension to 3D geometries are very popular algorithms [13], [14] that work well when enough data/projections are available and the noise is not too large. Algebraic iterative reconstruction methods [15] are also used frequently, and they can give better reconstructions in case of limited data. Yet other algorithms are based on variational formulations where one minimizes a combination of a data-fitting term and a regularization term (such as [16], [17]) that penalizes unwanted features; these methods can be more flexible and compensate for large noise and limited data. While TV regularization has been very popular in recent years, it still has some limitations such as smearing of texture-rich regions and staircasing [18].

Concerning segmentation techniques, there are many approaches to the problem of extracting regions of the image, e.g., by modeling the boundary between regions either using implicit contour models like the level-set methods [19], [20], [21] or including an explicit boundary model as in snakes [22]. Another group of methods are based on pixel labeling, e.g., modeled by a Markov Random Field where solutions can be obtained efficiently using graph-cut methods [23], [24]. In order to segment texture-rich images methods based on texture-features like the structure tensor [25] are popular, but also sparse methods have shown to be effective for texture segmentation [26], [27]. Some methods however require manual input [22], [27], [28] in order to work correctly. In CT all of the methods may be used, but it is preferable to use methods that do not need initialization by the user.

The classical approach with two separate steps (first reconstruct the image and then segment it) has proven to be quite effective. However, although prior knowledge about the expected types of objects or phases can be used during the segmentation, this knowledge is rarely used in the reconstruction step. Discrete tomography provides one way to incorporate such information about the desired pixel intensities.

In this paper we consider a general SRS framework and we show how various assumptions about the problem lead to a specific computational problem that can be handled by means of state-of-the-art numerical optimization methods. We then use numerical simulations to show that our algorithm can give very good segmentation result for problems where the parameters of the reconstructed phases are known, because our specific use of this prior information shrinks the range of possible solutions and hence increases the quality of the reconstructed image.

Our algorithm employs a user-specified HMMFM to assign to each pixel the probabilities for belonging to the given classes. In this way we give a quantitative measure of the similarity between each reconstructed pixel and each given class. This approach to the segmentation problem allows us to naturally use the connection between the HMMFM and the posterior probability density distribution. Our computational algorithm uses efficient numerical optimization algorithms, thus avoiding the slowly converging sampling methods.

Recall that X-ray CT is based on the principle that if we send X-rays through an object and measure the damping of each ray then, with infinitely many rays, we can perfectly reconstruct the object. The attenuation of an X-ray is proportional to the object's attenuation coefficient, as described by Lambert-Beer's law [29, §2.3.1]. We divide the domain onto pixels whose unknown nonnegative attenuation coefficients are organized in the vector $\boldsymbol{x} \in \mathrm{R}^N$. Similarly we organize the measured damping of the rays into the vector $\boldsymbol{b} \in \mathrm{R}^M$. Then we obtain a linear system of equations $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ with a large sparse *system matrix* governed solely by the geometry of the measurements: element a_{ij} is the length of the ith ray passing through pixel j, and the matrix is sparse because each ray only hits a small number of pixels [30].

Our paper is organized as follows. In Section 2 we present our general formulation of the reconstruction model, and in Section 3 we make our model more specific by means of assumptions related to the CT problem. Section 4 presents our considerations related to an efficient computational algorithm for the SRS problem. Our numerical experiments are presented in Section 5, and we finish the paper with a short conclusion in Section 6. Throughout the paper we use the following notation:

- i is the data index.
- j is the pixel index.
- k is the class index.
- $\boldsymbol{b}$ is a vector with the measured data $b_1, b_2, \ldots, b_M$.
- $\boldsymbol{x}$ is a vector with the pixels $x_1, x_2, \ldots, x_N$ of the image (the columns of the image are stacked into one long vector); the pixel values are the attenuation coefficients of the object.
- $\boldsymbol{A}$ is the $M \times N$ system matrix for the CT problem which describes the linear relationship $\boldsymbol{b} = \boldsymbol{A}\,\boldsymbol{x}$ between the attenuation coefficients and the data.
- $\delta = \{\delta_{jk}\}$ is the set of probabilities in the HMMFM for each class k of the object and for each pixel j, satisfying $\sum_{k=1}^{K} \delta_{jk} = 1$ for all j.
- $p(\boldsymbol{x}, \delta \,|\, \boldsymbol{b})$ is the posterior probability density function for the image and the HMMFM, given the data.

In each pixel, the probabilities for all the classes of the HMMFM must sum to one. We note that the above notation trivially generalizes to 3D problems where x_j denotes a voxel instead of a pixel.

2. Problem Formulation

To solve inverse problems it is necessary to incorporate prior information about the solution, in order to obtain a well-defined problem and a stable solution. In this work, our prior takes the form of explicit information about the different types of phases in the object, which provides a rigorous prior for the segmentation. Specifically, we assume that the object's attenuation coefficients (the pixels of the reconstruction) belong to K classes, where K is much smaller than the number N of pixels. We assume that K is known and we provide prior information about these classes in the form of probability densities associated with the classes. The parameters of the different classes (the mean attenuation coefficients μ_k and their variances σ_k for $k = 1, \ldots, K$) are assumed to be known from previous studies; see §3.2. Moreover, following [9] we use an HMMFM to incorporate a spatial prior inspired by the framework introduced in [8].

Our reconstruction problem originates from a classical constrained Bayesian formulation:

$$\begin{aligned} &\underset{\boldsymbol{x},\delta}{\operatorname{argmax}} \quad p(\boldsymbol{x},\delta\,|\,\boldsymbol{b}) \\ &\text{s.t.} \quad \sum_{k=1}^{K}\delta_{jk}=1,\ \delta_{jk}\geq 0, \quad j=1,\dots,N,\ k=1,\dots,K. \end{aligned} \tag{1}$$

In accordance with Bayes' rule, the probability $p(\boldsymbol{x},\delta\,|\,\boldsymbol{b})$ in (1) can be formulated as:

$$p(\boldsymbol{x},\delta\,|\,\boldsymbol{b})=\frac{p(\boldsymbol{b}\,|\,\boldsymbol{x},\delta)\,p(\boldsymbol{x}\,|\,\delta)\,p(\delta)}{p(\boldsymbol{b})}, \tag{2}$$

where

- $p(\boldsymbol{b}\,|\,\boldsymbol{x},\delta)=p(\boldsymbol{b}\,|\,\boldsymbol{x})$ is the probability of obtaining the data $\boldsymbol{b}$ given the image $\boldsymbol{x}$; the data does not depend on the segmentation of the image.
- $p(\boldsymbol{x}\,|\,\delta)$ is the probability of $\boldsymbol{x}$ given the probabilities of each class in each pixel.
- $p(\delta)$ expresses our belief in the HMMFM; usually, the more complex or chaotic the δ, the less we trust in it.
- $p(\boldsymbol{b})$ is a normalization constant.

The function given in (2) is a product of several probabilities, and to make the optimization process easier (as is common) we consider the logarithm of the objective function instead of the function itself. Thus, ignoring the normalization constant $p(\boldsymbol{b})$, we arrive at the general formulation of the SRS problem:

$$\begin{aligned} (\boldsymbol{x}^*,\delta^*) &= \underset{\boldsymbol{x},\delta}{\operatorname{argmax}}\log p(\boldsymbol{x},\delta\,|\,\boldsymbol{b}) \\ &= \underset{\boldsymbol{x},\delta}{\operatorname{argmax}}\Big(\log p(\boldsymbol{b}\,|\,\boldsymbol{x})+\log p(\boldsymbol{x}\,|\,\delta)+\log p(\delta)\Big) \\ \text{s.t.} \quad &\sum_{k=1}^{K}\delta_{jk}=1,\ \delta_{jk}\geq 0, \quad j=1,\dots,N,\ k=1,\dots,K. \end{aligned} \tag{3}$$

This model can be used for simple classification situations (e.g., graylevel/intensity classification) as well as for more complex classification algorithms. In this paper we will use and analyze this framework for a simple graylevel problem.

Having solved the problem for $\boldsymbol{x}$ and δ, it is easy to obtain the segmentation knowing the HMMFM. For each pixel x_j we compute the corresponding labels s_j as:

$$s_j\equiv\underset{k}{\operatorname{argmax}}\ \delta_{jk}, \qquad j=1,\dots,N \tag{4}$$

which is simply the most probable class in each pixel. Hereby we both obtain a reconstructed image and a segmentation based on specific model assumptions.

3. Assumptions for the CT Reconstruction Problem

To make the general SRS framework (3) more specific we make assumptions about the probability density functions that we use. Here we consider them as known, and indeed in many cases we will know the attenuation coefficients and their uncertainties for the phases under study (e.g., muscle, fat, bone, and air in medical imaging). How to obtain these parameters is not part of this work.

3.1. *Data Fitting Term*

Our first assumption regards the noise model. According to [31], the measured data is usually a sum of several terms:

- Data received from the X-ray illumination of the object, with Poisson noise.
- Poisson noise of the measuring equipment and from external sources.
- Gaussian noise caused by the electronics and the conversion from an analog signal to digital data.

Following [31] we assume that the signal-to-noise level is high enough that we can approximate the total noise by additive unbiased Gaussian noise $\mathcal{N}(0, \sigma_{\text{noise}}^2)$ where σ_{noise} is the standard deviation of the noise. We also assume that the noise is uncorrelated with the data and independent such that $p(\boldsymbol{b} \,|\, \boldsymbol{x}) = p(\boldsymbol{\varepsilon})$, where the vector $\boldsymbol{\varepsilon}$ represents the Gaussian noise, i.e., $\varepsilon_i \sim \mathcal{N}(0, \sigma_{\text{noise}}^2)$. Consequently the first term in our reconstruction model, which we refer to as the *data fidelity term*, is given by:

$$p(\boldsymbol{b} \,|\, \boldsymbol{x}) = \prod_{i=1}^{M} p(b_i \,|\, \boldsymbol{x}) = \frac{1}{\sqrt{2^M \pi^M} \sigma_{\text{noise}}^M} \exp\left(-\sum_{i=1}^{M} \frac{(\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b})_i^2}{2\sigma_{\text{noise}}^2} \right)$$
$$= \frac{1}{\sqrt{2^M \pi^M} \sigma_{\text{noise}}^M} \exp\left(-\frac{\|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}\|_2^2}{2\sigma_{\text{noise}}^2} \right),$$

where $(\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b})_i$ denotes the ith element of the residual vector. The first term in (3) thus becomes:

$$\log p(\boldsymbol{b} \,|\, \boldsymbol{x}) = -\frac{\|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}\|_2^2}{2\sigma_{\text{noise}}^2} - \frac{1}{2} \log(2^M \pi^M \sigma_{\text{noise}}^{2M}), \tag{5}$$

and we note that $\frac{1}{2} \log(2^n \pi^n \sigma_{\text{noise}}^{2n})$ is a constant that is unaffected by the optimization. The data fidelity term in this problem formulation is identical to a classical least squares fitting term.

3.2. *Class Fitting Term*

To specify the second term of (3) related to the classes of the image, we assume that the object is composed of a set of K different phases, and that each phase has the same attenuation coefficient everywhere. Specifically, we assume a Gaussian mixture where we specify the distribution of the attenuation coefficients within a class as a normal distribution with mean value μ_k equal to the expected attenuation coefficient and a (small) standard deviation σ_k. Here, all μ_k and σ_k for $k = 1, \ldots, K$ are assumed to be known, and we have

$$p(x_j \,|\, \text{class} = k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x_j - \mu_k)^2}{2\sigma_k^2} \right).$$

Now let $\delta_j = \{\delta_{j1}, \dots, \delta_{jK}\}$ denote the set of the K class probabilities associated with pixel j. Then we can write the probability $p(\boldsymbol{x} \,|\, \delta)$ as

$$p(\boldsymbol{x} \,|\, \delta) = \prod_{j=1}^{N} p(x_j \,|\, \delta_j) = \prod_{j=1}^{N} \sum_{k=1}^{K} p(x_j \,|\, \text{class} = k)\, p(\text{class} = k \,|\, \delta_j).$$

Since we assume that $p(\text{class} = k \,|\, \delta_j) = \delta_{jk}$ is the probability of the pixel j belonging to class k, the above expression takes the form:

$$p(\boldsymbol{x} \,|\, \delta) = \prod_{j=1}^{N} \sum_{k=1}^{K} \delta_{jk} \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x_j - \mu_k)^2}{2\sigma_k^2} \right).$$

Thus, the second term of (3) can be written as

$$\log p(\boldsymbol{x} \,|\, \delta) = \sum_{j=1}^{N} \log \left[\sum_{k=1}^{K} \frac{\delta_{jk}}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x_j - \mu_k)^2}{2\sigma_k^2} \right) \right]. \tag{6}$$

We refer to this as the *class-fitting term.*

3.3. *Regularization Term*

The third term of (3) is our confidence in the segmentation and it is thus a *regularization term* where we specify our prior knowledge about the behavior of the segmentation. Contrary to classical regularization methods, our prior is concerned with the classes (and not the pixel values). We use the following generic formulation

$$\log p(\delta) = -\sum_{k=1}^{K} R(\delta_k), \tag{7}$$

where $\delta_k = \{\delta_{1k}, \dots, \delta_{Nk}\}$ is the set of probabilities for class k and R is a function that expresses our prior, such as our knowledge of the expected spatial correlations of the classes among the pixels. In this paper, we use two different functions inspired by common choices in image reconstruction, cf. [12].

For ease of presentation, consider the case where $\boldsymbol{x}$ represents a 2D image (the extension to 3D is obvious) and let j' and j'' denote indices to the two elements of $\boldsymbol{x}$, in its 2D representation, that are neighbors horizontally and vertically. Let $\mathcal{J}$ denote the set of indices j for which both j' and j'' refer to pixels inside the image.

- The total variation (TV), i.e., the 1-norm of the gradient magnitude for the kth class associated with all pixels of $\boldsymbol{x}$:

$$R_{\text{TV}}(\delta_k) = \sum_{j \in \mathcal{J}} \left((\delta_{jk} - \delta_{j'k})^2 + (\delta_{jk} - \delta_{j''k})^2 \right)^{1/2}. \tag{8}$$

- A Tikhonov-type regularizer equal to the squared 2-norm of the gradient magnitude for the kth class associated with all pixels of $\boldsymbol{x}$:

$$R_{\text{Tik}}(\delta_k) = \sum_{j\in\mathcal{J}} \big((\delta_{jk} - \delta_{j'k})^2 + (\delta_{jk} - \delta_{j''k})^2 \big). \tag{9}$$

The use of the TV function R_{TV} (8) allows discontinuities in the probabilities for the classes associated with neighboring pixels – and one expects this to be well suited for the segmentation process. The use of the Tikhonov function R_{Tik} (9) enforces some spatial smoothness of the probabilities among classes associated with neighboring pixels.

3.4. *The Reconstruction Model*

At this time we assemble the three terms specified above into the complete reconstruction model. In doing so, we introduce two *regularization parameters* λ_{noise} and λ_{class} that let us balance the weight given to the noise term and the regularization term, respectively, and as is common we absorb the factor $1/(2\sigma^2_{\text{noise}})$ into λ_{noise}. The complete reconstruction model thus takes the following form (swapping "min" for "max" and changing signs of the terms):

$$\begin{aligned}(\boldsymbol{x}^*, \delta^*) = \operatorname*{argmin}_{\boldsymbol{x},\delta} \Bigg(& \lambda_{\text{noise}} \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda_{\text{class}} \sum_{k=1}^{K} R(\delta_k) \\ & - \sum_{j=1}^{N} \log \left[\sum_{k=1}^{K} \frac{\delta_{jk}}{\sqrt{2\pi}\sigma_k} \exp\left(- \frac{(x_j - \mu_k)^2}{2\sigma_k^2} \right) \right] \Bigg) \qquad (10) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \delta_{jk} = 1, \ \delta_{jk} \geq 0, \quad j = 1, \ldots, N, \ k = 1, \ldots, K.\end{aligned}$$

To summarize, our reconstruction problem (10) takes as input our data $\boldsymbol{b}$ and the class parameters μ_k, σ_k, $k = 1, \ldots, K$, together with the regularization function $R(\cdot)$ and the regularization parameters λ_{noise}, λ_{class}, and produces the solution pair $\boldsymbol{x}^*$ and δ^*. We note that the reconstruction problem is non-convex in $\boldsymbol{x}$ and our computational algorithm (described in the next section) must take this into account.

3.5. *Simplifications*

To deal with the non-convexity of the problem, we introduce two different simple convex approximations to the class-fitting term $p(\boldsymbol{x}\,|\,\delta)$, and both of them are used in the algorithm described in the next section. Note that the objective function in (10) is a multi-modal function for each pixel, consisting of a sum of Gaussian functions, that causes the non-convexity of the problem. The underlying idea is then to approximate this function with a uni-modal approximation – a single Gaussian function – such that we can use standard methods from convex optimization.

The first type of simplification was introduced in [8], and for each pixel x_j it "lumps" the parameters δ_{jk}, μ_k and σ_k into the parameters $\hat{\mu}_j$, $\hat{\sigma}_j$ and $\hat{\delta}_j = 1$, where

$$\hat{\mu}_j = \sum_{k=1}^{K} \delta_{jk}\mu_k, \quad \hat{\sigma}_j^2 = \sum_{k=1}^{K} \delta_{jk}(\sigma_k^2 + \mu_k^2) - \hat{\mu}_j^2, \qquad j = 1, \ldots, N. \tag{11}$$

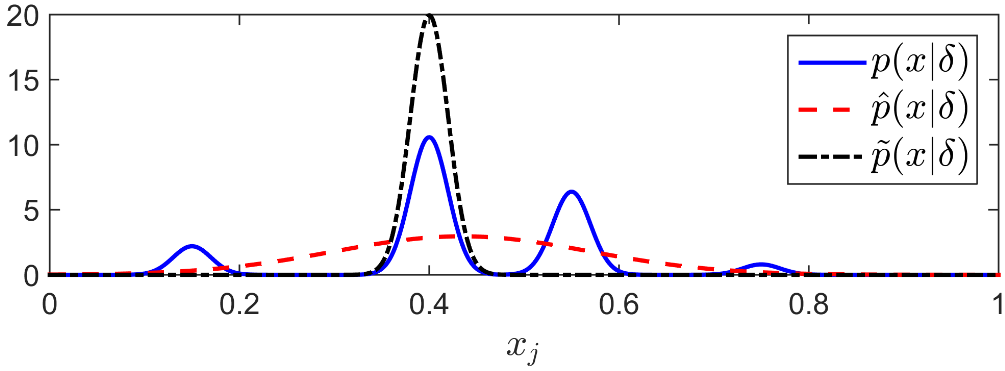

Figure 1. Illustration of the simplified uni-modal class-fitting functions $\hat{p}(\boldsymbol{x} \,|\, \delta)$ (12) and $\tilde{p}(\boldsymbol{x} \,|\, \delta)$ (13) together with the multi-modal function $p(\boldsymbol{x} \,|\, \delta)$ (6). We use the parameters $K = 4$, $\delta_j = \{0.11, 0.53, 0.32, 0.04\}$, $\mu_1 = 0.15$, $\mu_2 = 0.4$, $\mu_3 = 0.55$, $\mu_4 = 0.75$, and $\sigma_1 = \sigma_2 = \sigma_3 = \sigma = 4 = 0.02$.

The simplified function then takes the form (omitting an additive constant):

$$\log \hat{p}(\boldsymbol{x} \,|\, \delta) = -\sum_{j=1}^{N} \frac{(x_j - \hat{\mu}_j)^2}{2\hat{\sigma}_j^2}. \tag{12}$$

The second type of simplification takes a different approach. When we are close to the solution we expect that for most of the pixels x_j the corresponding δ_{jk} will approach a situation where one of them, say, δ_{jk_j}, is close to 1 while the remaining ones are small – corresponding to a high probability that this pixel belongs to class k_j. In this situation it is natural to set

$$\hat{\delta}_{jk} = \begin{cases} 1, & \text{if } k = k_j, \\ 0, & \text{otherwise}, \end{cases} \qquad j = 1, \ldots, N$$

and then (11) simplifies to $\hat{\mu}_j = \mu_{k_j}$, $\hat{\sigma}_j = \sigma_{k_j}$, and we obtain the simplified function

$$\log \tilde{p}(\boldsymbol{x} \,|\, \delta) = -\sum_{j=1}^{N} \frac{(x_j - \mu_{k_j})^2}{2\sigma_{k_j}^2}. \tag{13}$$

Figure 1 illustrates these approximations.

4. Algorithm

We compute a solution to the reconstruction problem (10) using an iterative, alternating optimization algorithm as is common practice for non-convex problems. Our algorithm is based on the following idea. First we use the simplification in (12) to roughly approximate the solution. When we are close to the solution, we instead use the simplification (13) in order to improve the reconstruction result. Hence the algorithm has two overall stages, and in both stages the iterations alternate between two steps in which we update either the image or the classes. In the first stage we compute an approximate solution pair ($\hat{\boldsymbol{x}}^*$, $\hat{\delta}^*$) using the simplified expression (12). This approximation is then used as initial guess for the second stage based on the alternative simplified function in (13). The algorithm is summarized in Fig. 2 at the end of this section.

4.1. *Stage 1*

In the first stage we apply our iterative scheme to problem (10) with the simplifications (11) and (12). Our algorithm alternates between two steps in which we update either $\hat{\boldsymbol{x}}^n$ or $\hat{\delta}^n$, where the superscript n denotes the iteration. In the *image-update step* we update the pixel values:

$$\hat{\boldsymbol{x}}^{n+1} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left(\lambda_{\text{noise}} \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{N} \frac{(x_j - \hat{\mu}_j^n)^2}{2(\hat{\sigma}_j^n)^2} \right), \tag{14}$$

and we note that $\hat{\mu}_j^n$ and $\hat{\sigma}_j^n$ depend on the iteration – they are functions of the current HMMFM and computed using (11)). The function in (14) is convex in $\boldsymbol{x}$ and we compute its minimum by means of the Conjugate Gradient Least Squares (CGLS) algorithm [32] which is well suited for large-scale problems. As the initial guess for CGLS we use the image from the previous iterate.

In the *class-update step* we update the classes:

$$\hat{\delta}^{n+1} = \underset{\delta}{\operatorname{argmin}} \left(\lambda_{\text{class}} \sum_{k=1}^{K} R(\delta_k) - \sum_{j=1}^{N} \log \left[\sum_{k=1}^{K} \frac{\delta_{kj}}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(\hat{x}_j^n - \mu_k)^2}{2\sigma_k^2} \right) \right] \right) \tag{15}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \delta_{jk} = 1, \quad \delta_{jk} \geq 0, \qquad j = 1, \ldots, N, \quad k = 1, \ldots, K.$$

Different algorithms can be used to solve this problem; we use the iterative Frank-Wolfe algorithm [33], also known as the conditional gradient method, which enforces the conditions (for $j = 1, \ldots, N$ and $k = 1, \ldots, K$)

$$\sum_{k=1}^{K} \hat{\delta}_{jk}^{n+1} = 1 \qquad \text{and} \qquad \hat{\delta}_{jk}^{n+1} > 0.$$

We choose the initial guess $\hat{\delta}$ for this algorithm as $\hat{\delta}_{jk}^0 = 1/K$, for all j and k, which by numerical experiments was found to be the most robust initialization.

4.2. *Stage 2*

When the changes in the image $\hat{\boldsymbol{x}}^n$ and the classes $\hat{\delta}^n$ become small during Stage 1, this signals that we are making little progress with the approach based on the simplifications (11) and (12). We stop these iterations when

$$\|\hat{\boldsymbol{x}}^{n+1} - \hat{\boldsymbol{x}}^n\|_2 / \|\hat{\boldsymbol{x}}^n\|_2 \leq 10^{-6}. \tag{16}$$

At this stage, most of the pixels j in the image have been assigned to a single class, meaning that the corresponding δ_{jk} are close to 0 or 1 and $\hat{\sigma}_j \ll \|\hat{\boldsymbol{x}}^n\|_2$.

It is now natural to switch to Stage 2 in which we use the second type of simplification in (13) where, for each pixel, we only use the class with the highest probability. The main goal of this stage is to consider those remaining pixels for which δ_{jk} have not yet approached 0 or 1, which typically are those at the edges inside the image. Equation (13) ensures that these pixels are also assigned to a single class, forcing $\hat{\sigma}_j$ to be small. We emphasize that, according to our experiments, Stage 2 will give good results only when most pixels have been assigned to a single class in Stage 1.

First we perform the image-updating step, which now takes the form:

$$\boldsymbol{x}^{n+1} = \operatorname*{argmin}_{\boldsymbol{x}} \left(\lambda_{\text{noise}} \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{N} \frac{(x_j - \mu_{k_j})^2}{2\sigma_{k_j}^2} \right), \tag{17}$$

where k_j is the index to the largest δ_{jk} for the jth pixel. We still use CGLS to solve (17). The class-updating step is identical to that of Stage 1, and again we solve it with Frank-Wolfe algorithm. Usually it is enough to perform 1–3 iterations of the Stage 2. Our observation is that the fine-tuning of the classification results in an improved reconstruction.

```
Initialization: $\hat{\delta}^0_{jk} = 1/K$, $\forall j, k$.
Stage 1
    For $n = 1, 2, \ldots, n_1$
        Compute $\hat{\mu}_j$ and $\hat{\sigma}_j$ using (11).
        Compute $\hat{\boldsymbol{x}}^n$ using (14) by means of CGLS with initial guess $x_j = \hat{\mu}_j^{n-1}$.
        Compute $\hat{\delta}^n$ using (15) by means of FWA with initial guess $\hat{\delta}^{n-1}$.
Stage 2
    Set $\boldsymbol{x}^{n_1} = \hat{\boldsymbol{x}}^{n_1}$ and $\delta^{n_1} = \hat{\delta}^{n_1}$.
    For $n = n_1 + 1, n_1 + 2, \ldots, n_1 + n_2$
        Compute $k_j$ such that $\delta_{jk_j} = \max_k \delta_{jk}$, $\forall j$.
        Compute $\boldsymbol{x}^n$ using (17) by means of CGLS with initial guess $x_j = \mu_{k_j}^{n-1}$.
        Compute $\delta^n$ using (15) by means of FWA with initial guess $\delta^{n-1}$.
Finish: Set $\boldsymbol{x}^* = \boldsymbol{x}^{n_1+n_2}$ and $\delta^* = \delta^{n_1+n_2}$.
```

Figure 2. Summary of our SRS algorithm; "CGLS" is a robust implementation of the Conjugate Gradient algorithm for Least Squares problems, and "FWA" denotes the Frank-Wolfe algorithm. Note that an initial image is not needed for the SRS algorithm.

5. Numerical Results

In this section we present a series of numerical experiments where we compare our two-stage algorithm SRS with the classical approach where the reconstruction and segmentation are performed in two consecutive independent steps; the segmentation step knowing the parameters μ_k and σ_k for each class. We consider two versions of our algorithm, SRS-TV and SRS-Tik, corresponding to the two different regularization terms (8) and (9), respectively. We use quite small test images such that the artifacts of the different algorithms are clearly visible.

For the reconstruction step of the classical approach we use two different algorithms: the Filtered Back Projection (FBP) method as implemented in MATLAB's `iradon` function, and Total Variation (TV) regularization as implemented in the software package[1]

[1] The code for TVReg is available from `http://www2.compute.dtu.dk/~pcha/TVReg/`.

Table 1. Regularization parameters used in the SRS algorithm for the different test problems in this section and the real-data example in the next section.

	Shepp-Logan	Binary	4-class	Gray-scale	Real data
SRS-Tik λ_{noise}	$4.2 \cdot 10^{-3}$	$4.5 \cdot 10^{-4}$	$6.5 \cdot 10^{-4}$	$4.0 \cdot 10^{-4}$	0.2
λ_{class}	1.0	0.5	0.5	0.4	0.3
SRS-TV λ_{noise}	$2.0 \cdot 10^{-2}$	$9.5 \cdot 10^{-4}$	$3.0 \cdot 10^{-2}$	$8.5 \cdot 10^{-4}$	–
λ_{class}	0.4	0.3	0.32	0.5	–

TVReg [34] which solves the problem

$$\min_{\boldsymbol{x}} {}^1\!/_2 \|\boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \alpha\,\mathrm{TV}(\boldsymbol{x}), \qquad 0 \le x_i \le 1, \quad i = 1, \ldots, N, \tag{18}$$

where $\mathrm{TV}(\boldsymbol{x})$ uses standard finite differences to compute the gradient magnitude. The segmentation in the classical approach is done by means of a Markov random field algorithm via graph cuts [35] using the `graph-tool` software [36], assuming that the number K of classes is known.

5.1. *Test Using Artificial Data*

The test problem used throughout these experiments is a standard 2D CT problem with 58 projections at angles $(i/58) \cdot 180°$, $i = 1, 2, \ldots, 58$, and with 181 parallel rays for each angle, and the image is 128×128 pixels. The corresponding sparse matrix A is thus of size $M \times N = 10,499 \times 16,384$ corresponding to an underdetermined system. Each nonzero element a_{ij} is the length of ray i through pixel j, and A was generated by means of the function `paralleltomo` from the MATLAB package[2] AIR Tools [15]. Given the exact test image $\boldsymbol{x}^{\text{exact}}$ we generate data with additive Gaussian noise as $\boldsymbol{b} = A\,\boldsymbol{x}^{\text{exact}} + \boldsymbol{e}$, where each element of the noise vector $\boldsymbol{e}$ is from the same normal distribution with zero mean and standard deviation chosen such that $\|\boldsymbol{e}\|_2 / \|A\,\boldsymbol{x}^{\text{exact}}\|_2 = 0.01$.

To study the performance of the algorithms, we use four different test images $\boldsymbol{x}^{\text{exact}}$ (they are shown in Figures 5–6):

(1) The Shepp-Logan phantom from MATLAB.
(2) A binary (2-class) phantom consisting of an image with random fine structures that are mostly horizontal.
(3) A 4-class phantom with random regions separated by thin structures.
(4) A gray-scale image with sharp contours and regions with smoothly varying intensities.

We include the Shepp-Logan phantom because it is so widely used (although it is an easy problem for TV regularization). The last other phantoms are generated by the function `phantomgallery` in AIR Tools.

In the FBP reconstructions we used the Hann filter and linear interpolation. In the TV reconstruction algorithm and in our method we manually chose the regularization parameters that minimize the image reconstruction error. In the SRS algorithm we choose the regularization parameters to give a good balance between reconstruction errors and segmentation errors; the parameters we use are listed in Table 1. The development of an automatic procedure for choosing these parameters is outside the scope of this paper. The number of iterations n_1 in Stage 1 is based on the stopping rule (16); in Stage 2 we always used $n_2 = 5$ iterations.

[2] Version 1.2 of the AIR Tools package is available from `http://www2.compute.dtu.dk/~pcha/AIRtools/`.

Table 2. Summary of errors. In each entry, the top and bottom numbers are the relative image reconstruction error ε_{rec} and the relative segmentation error ε_{seg}, respectively.

Test problem	FBP→Seg	TV→Seg	SRS-Tik	SRS-TV
1: Shepp-Logan	0.34 0.056	0.038 0.0038	0.021 0.0026	0.023 0.0031
2: Binary	0.46 0.096	0.33 0.035	0.18 0.015	0.26 0.029
3: 4-class	0.39 0.38	0.16 0.077	0.047 0.0057	0.055 0.0064
4: Gray-scale	0.24 0.095	0.082 0.0040	0.060 0.0047	0.087 0.0051

5.1.1. *Study of Convergence*

Given the reconstructed image $\boldsymbol{x}^*$, the *relative image reconstruction error* is defined as

$$\varepsilon_{\text{rec}} = \|\boldsymbol{x}^{\text{exact}} - \boldsymbol{x}^*\|_2 / \|\boldsymbol{x}^{\text{exact}}\|_2. \tag{19}$$

Given the vector $\boldsymbol{s}^*$ of labels computed by means of (4), the *relative segmentation error* is defined as the fraction of mislabeled pixels,

$$\varepsilon_{\text{seg}} = \frac{1}{N} \sum_{j=1}^{N} I(s_j^{\text{exact}} \neq s_j^*), \tag{20}$$

where $\boldsymbol{s}^{\text{exact}}$ is the vector of true labels, N is the number of pixels in the image, and I is a logical indicator function.

The error histories for algorithm SRS-Tik are shown in Fig. 3, and images and segmentations for selected iterations are shown in Fig. 4 (the error histories, reconstructions, and segmentations for SRS-TV are very similar and not shown here). The error decreases monotonically; the first few iterations give the biggest reduction and then the error reaches a plateau. During Stage 1 both the reconstruction and the segmentation improve – the image intensities improve, noise is removed, and the edges become increasingly sharper. A few steps of Stage 2 improves the quality of the reconstruction, while the segmentation is hardly changed.

5.1.2. *Study of Reconstruction and Segmentation Errors*

The reconstruction and segmentation errors ε_{rec} and ε_{seg} for all algorithms and phantoms are summarized in Table 2, and the computed images and segmentations are shown in Figures 5 and 6. "FBP→Seg" and "TV→Seg" denote reconstruction by means of FBP and TV, respectively, followed by segmentation, while "SRS-Tik" and "SRS-TV" denote our algorithm with the regularization terms (9) and (8), respectively.

We see from both the table and the figures that the worst results are produced by FBP → segmentation; both the reconstructed image and the segmented image contain a lot of noise and artifacts.

The results for TV → segmentation are closer to the ground truth, but the reconstructions and the segmentations have the characteristic TV artifacts: the reconstructions capture small details but they also exhibit the standard TV staircasing effect, they include a number of small unwanted "clusters" of pixels, and the pixel intensities tend to be incorrect. Indeed, the images of the pixel-wise absolute errors $|\boldsymbol{x}^{\text{exact}} - \boldsymbol{x}^*|$ clearly show the incorrect image intensities with errors in the range 0.1–0.4. The underlying problem

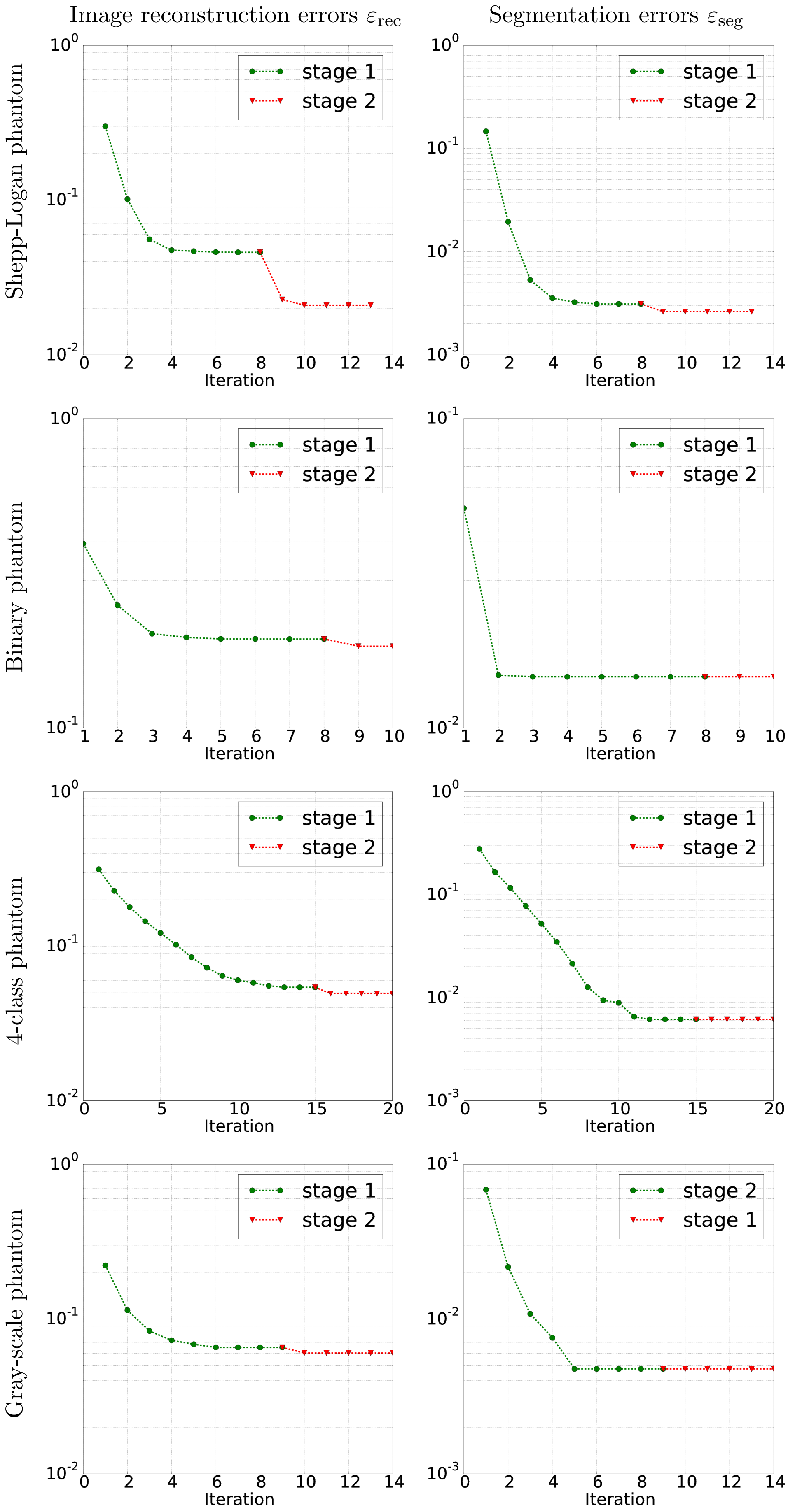

Figure 3. Error histories: the evolution of the image and segmentation errors $\varepsilon_{\mathrm{rec}}$ and $\varepsilon_{\mathrm{seg}}$ during the iterations of the SRS-Tik algorithm (the error histories for SRS-TV are very similar).

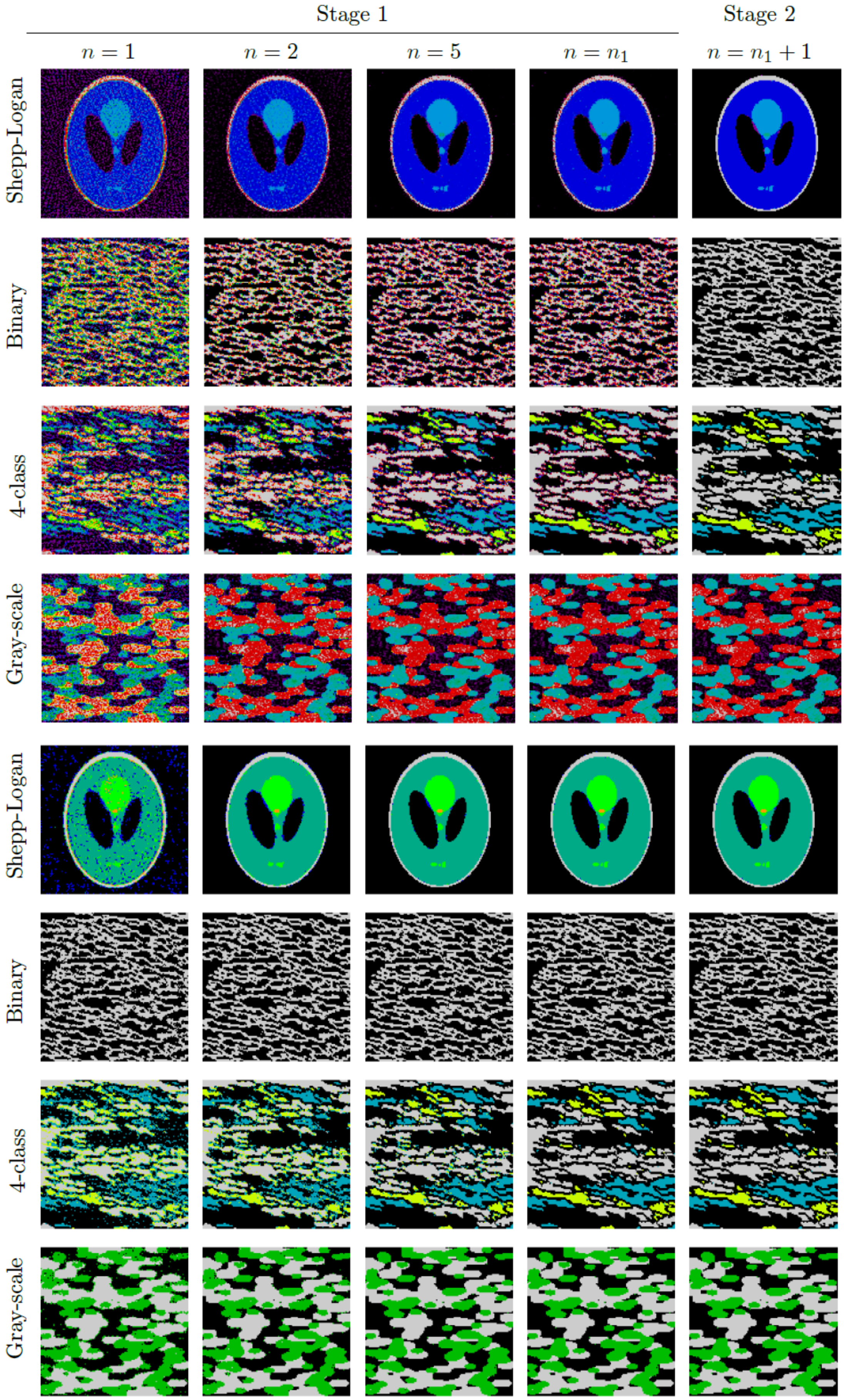

Figure 4. Images $\boldsymbol{x}^n$ (top) and segmentations $\boldsymbol{s}^n$ (bottom) for selected iterations during the SRS-Tik algorithm. For colorbars, see Figs. 5 and 6.

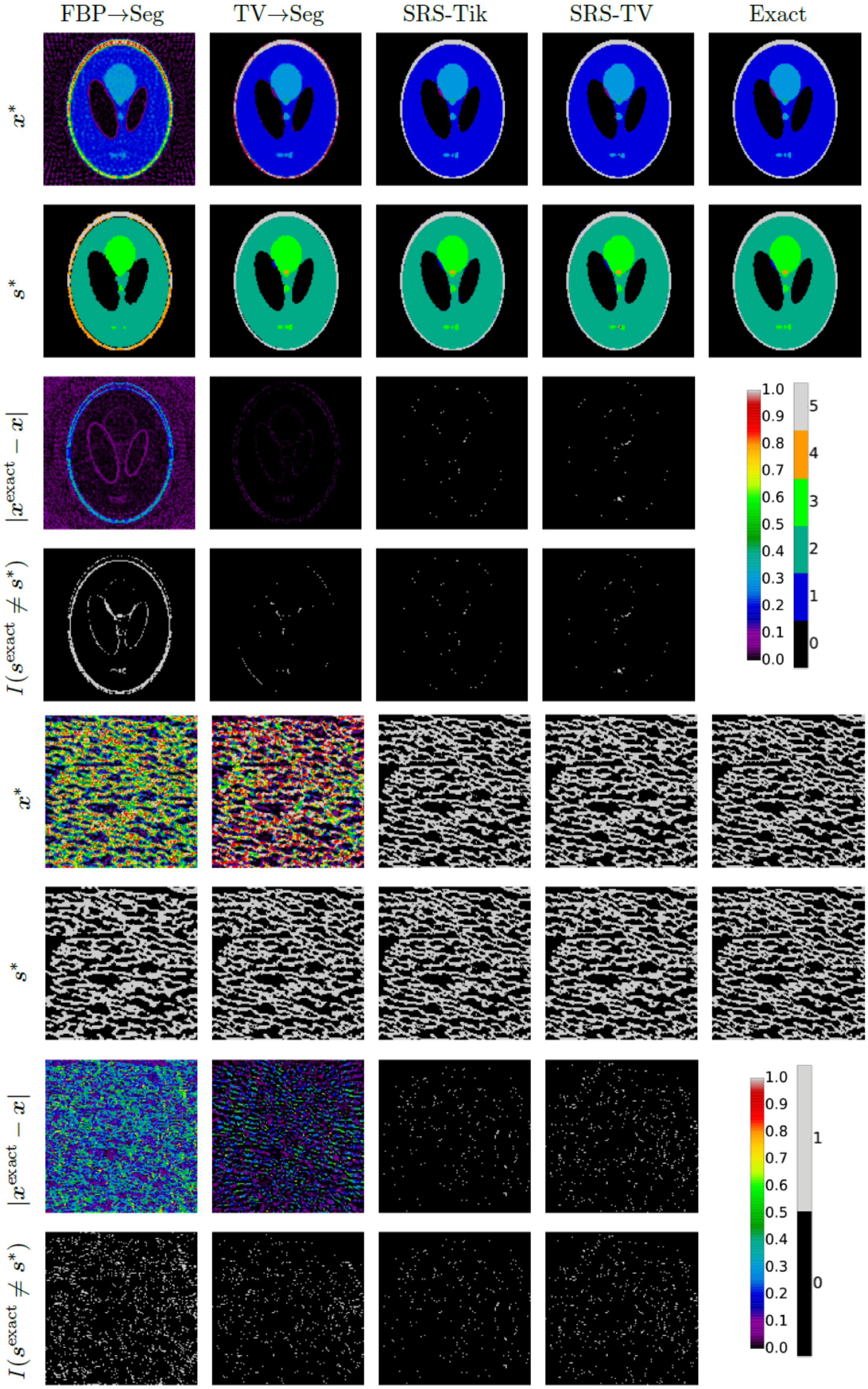

Figure 5. Results for the Shepp-Logan test problem (four top rows) and the binary test problem (bottom four rows). The left and right colorbars are for the reconstructions and segmentations, respectively.

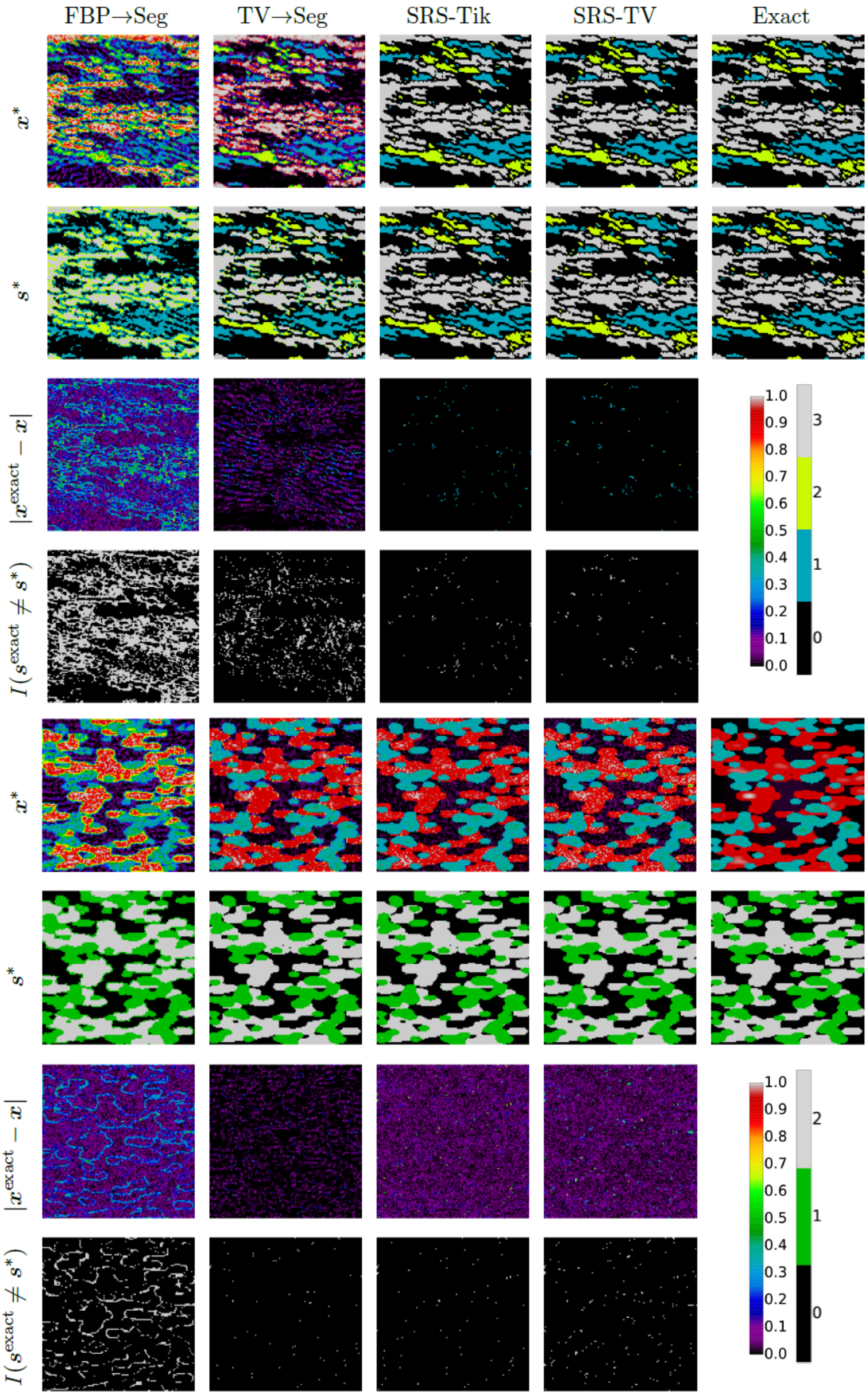

Figure 6. Results for 4-class test problem (top four rows) and the gray-scale test problem (bottom four rows). The left and right colorbars are for the reconstructions and segmentations, respectively.

Table 3. Sensitivity experiments with the 4-class phantom and the SRS-Tik algorithm. The true estimates are $\mu_1 = 0$, $\mu_2 = 0.33$, $\mu_3 = 0.66$, $\mu_4 = 1$ and the results in the previous section were obtained with $\sigma_k = 10^{-4}$, $k = 1, 2, 3, 4$. For these values the errors are $\varepsilon_{\mathrm{rec}} = 0.047$ and $\varepsilon_{\mathrm{seg}} = 0.0057$, cf. Table 2. See the text for an explanation of the experiments.

Experiment	μ	0.5	0.55	0.6	0.7	0.75	0.8
1	$\varepsilon_{\mathrm{rec}}$	0.14	0.106	0.074	0.056	0.095	0.139
	$\varepsilon_{\mathrm{seg}}$	0.05	0.027	0.013	0.007	0.023	0.06
2	$\varepsilon_{\mathrm{rec}}$	0.27	0.247	0.165	0.082	0.122	0.151
	$\varepsilon_{\mathrm{seg}}$	0.146	0.130	0.065	0.018	0.040	0.076
3	$\varepsilon_{\mathrm{rec}}$	0.132	0.103	0.083	0.067	0.105	0.142
	$\varepsilon_{\mathrm{seg}}$	0.042	0.025	0.016	0.011	0.033	0.065

is that we underestimate jumps in the image intensities [18].

We note that the TV reconstructions are slightly under-regularized, i.e., they include some amount of noise. Our empirical experience is that a bit of under-regularization in TV give less segmentation errors, because a slightly under-regularized TV reconstruction is more likely to include small features of the ground truth image.

For test problems 1, 2 and 3 the results from our SRS algorithm are more accurate than those from the classical approach, both for the reconstructions and the segmentations. This is especially true for test problems 2 and 3 which contain a lot of fine-structure, and for all test problems our reconstructions look significantly sharper than those from FBP and TV. Moreover we avoid the staircasing effect associated with TV reconstructions.

From a computational point of view, the TV regularization term R_{TV} (8) in algorithm SRS-TV is more difficult to handle than the Tikhonov term R_{Tik} (9) used in SRS-Tik. As a consequence, for SRS-TV the range of appropriate regularization parameters is quite small and the computing time is large, while for SRS-Tik the range for appropriate parameters is larger and the computing time is significantly smaller. From a practical point of view this makes the SRS-Tik algorithm more favorable.

Another advantage of SRS-Tik over SRS-TV is that during the iterations the former allows more flexibility in the location of sharp transitions in the probabilities in the HMMFM. This is because TV insists on putting discontinuities in these probabilities, while Tik allows more smooth transitions, and hence the location of the edges is done in a more flexible way. This is particularly evident from test problem 3, see the images during the iterations shown in Fig. 4. That means that in the pixels that are close to the detected edge the σ_j is rather big and that allows to tune the edge better during the iterations.

In conclusion we propose to use the Tikhonov regularization term and the corresponding SRS-Tik algorithm, because it requires less computing time, the choice of the regularization parameter is less critical, and its reconstructions and segmentations are competitive with those of algorithm SRS-TV.

5.2. *Robustness Test*

In our algorithm we assume that the user specifies the parameters μ_k and σ_k. It is interesting to see how sensitive the computed reconstruction and segmentation are to the estimates μ_k of the attenuation coefficients and the associated σ_k. To study this we changed one of the estimates to a wrong value and analyzed how the resulting reconstruction and segmentation change, compared to results obtained with the correct estimate.

The results are shown in Table 3 for the SRS-Tik algorithm with the same choice of λ_{noise} and λ_{class} as before applied to the 4-class phantom. In all the experiments we kept

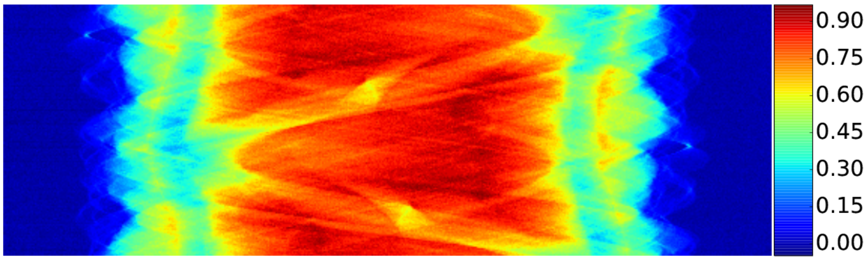

Figure 7. The measured data (the sinogram) for the real-data problem.

μ_1, μ_2 and μ_4 at the correct values and varied μ_3 around its correct value 0.66. We show the reconstruction and segmentation errors (19) and (20) for the following three experiments:

(1) Here we kept $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 10^{-4}$.
(2) Here $\sigma_1 = \sigma_2 = \sigma_4 = 10^{-4}$ while $\sigma_3 = 2 \cdot 10^{-4}$.
(3) Here $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = 2 \cdot 10^{-4}$.

When μ_3 assumes an incorrect value then clearly the errors increase, but we note that the results are still better than the results obtained with the TV→Seg algorithm for which the errors are $\varepsilon_{\text{rec}} = 0.16$ and $\varepsilon_{\text{seg}} = 0.077$ (from Table 2). Hence, our algorithm can handle an incorrect estimate as long as it does not take a value close to one of the other estimates (in which case the segmentation becomes wrong).

One might expect that we can compensate for a wrong estimate μ_3 with a bigger σ_3. Unfortunately, this is not true for our algorithm as documented in the table. In fact, changing only one σ_k tends to increase the errors, and in particular the segmentation error, because many pixels are mislabel when the parameters σ_k are unbalanced. As also shown in the table, increasing all σ_k to the same value is preferable, and we note that we obtain almost the same errors when all σ_2 are twice as large.

The conclusion of this experiment is that we should use balanced values for σ_k and that we cannot use σ_k to compensate for an incorrect μ_k.

5.3. *Test Using Real Data*

We complete our performance studies with some experiments using real data. The data was collected using a micro-CT scanner with fan-beam geometry that scans the object slice-by-slice. For our reconstruction we use 167 projections, with 512 rays in each projection. The reconstructed image contains 362 × 362 pixels, and hence the amount of pixels is approximately 1.5 times larger than the amount of data. The data, in the form of the sinogram, is shown in Fig. 7.

The scanned object is a candy that contains air, a nut at the center, pieces of nuts, chocolate, and waffle, enclosed in a thin aluminum foil. Nuts and waffle have almost the same attenuation coefficients, so we treat them as the same class. The foil, which has a high attenuation coefficient, is very thin and since the resolution is too low to accurately represent the foil, the reconstructed coefficients are inaccurate; for this reason we do not account for the foil in our classes. As priors for the classes we used the following mean values and standard deviations.

- Air: $\mu_1 = 0$, $\sigma_1 = 0.001$.
- Nuts and waffle: $\mu_2 = 0.0033$, $\sigma_2 = 0.001$.
- Chocolate: $\mu_3 = 0.0044$, $\sigma_3 = 0.001$.

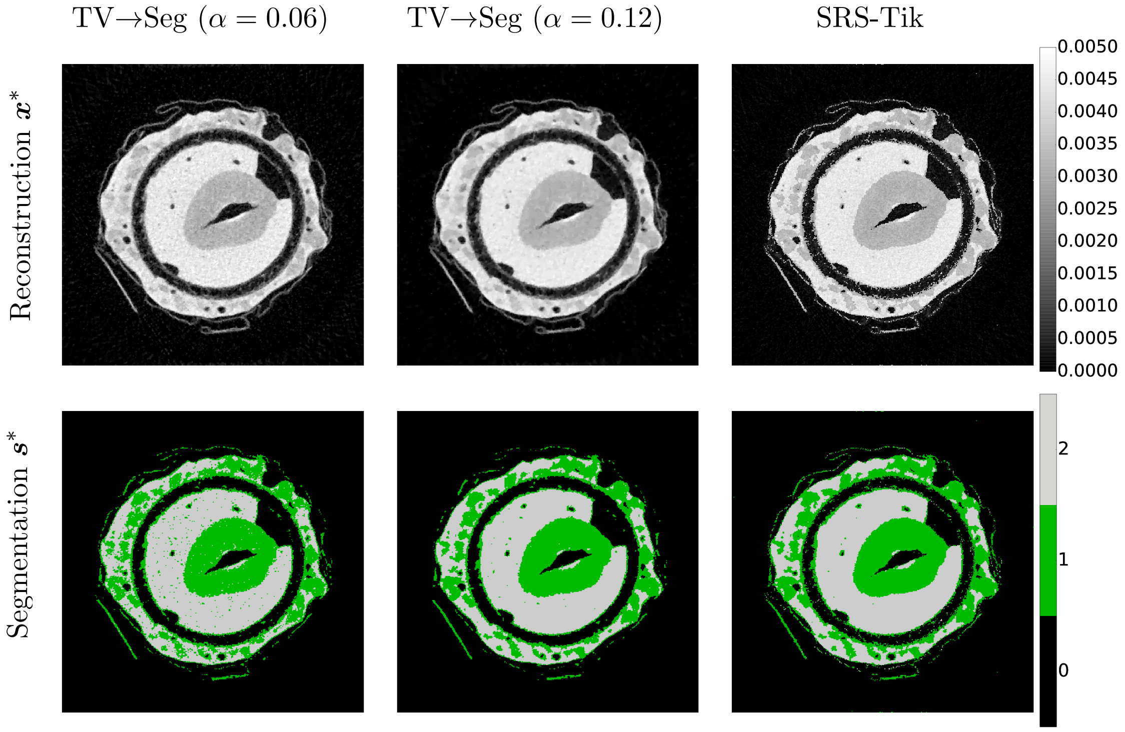

Figure 8. Results for the real-data problem; all reconstructs/segmentations have the same color scale.

Knowing that FBP is not well suited for underdetermined problems, we compare two algorithms: TV → segmentation (using again the alpha-expansion segmentation algorithm) and our SRS-Tik algorithm. The matrix A for this problem was generated with the function `fanbeamtomo` from AIR Tools [15].

The reconstructions are shown in Fig. 8. We computed TV reconstructions for two different values of the regularization parameter α in (18); for the small value we are able to recover small details in the segmentation, but there are many misclassified pixels; for the larger parameter we have more homogeneous areas in the segmentation but we miss many small details (for example, it is difficult to see the edges between different phases). In conclusion, the TV → segmentation algorithm does not perform so well.

In the SRS-Tik reconstruction the edges are always clearly visible and overall image looks significantly sharper than TV reconstruction. Although our reconstruction looks slightly more noisy (because we do not use a smoothing prior in the reconstruction), we emphasize that the majority of the noise-looking pixels are actually correlated with the texture of the reconstructed object. Our segmentation of the object identifies more correct pixels than the TV → segmentation algorithm.

Although we do not know the foil's precise attenuation coefficient, in the SRS-Tik reconstruction the position of the foil is reconstructed correctly and better than in case TV→Seg algorithm – in both algorithms it is assigned to the second class.

6. Conclusion

We presented a new two-stage algorithm for simultaneous image reconstruction and segmentation, with application to absorption CT, that avoids the drawbacks of classical approaches where reconstruction and segmentation are done in two individual steps. Moreover, in contrast to approaches based on discrete tomography, we produce both a reconstructed image and a segmentation. The novelties of our approach are: 1) application of a Tikhonov or Total Variation regularization term for these classes, 2) a two-stage formulation that uses simplified reconstruction models in each stage, and 3) a state-of-the-art numerical optimization algorithm for solving the underlying problem.

Numerical experiments with artificial and real data demonstrate that we are able to produce improved images and classifications, compared to the classical approach. We are able to produce sharp edges in the reconstructions, the segmentations are accurate, and the experiments demonstrate that a regularizer applied to the segmentation also has a strong effect on the reconstructed image. We recommend to use the variant of our algorithm that uses the Tikhonov regularization term, since the TV variant is much slower, requires more tuning of the regularization parameters, and does not give better results.

The main competitor to our method is the classical approach that uses TV regularization followed by segmentation; it has only one regularization parameter. However, a clear advantage of our method is that we obtain almost correct image intensities because we apply the regularization to the classes (instead of the image) and in case of complex images our algorithm reconstructs the structure significantly better. A disadvantage is that our algorithm, in its present formulation, has two regularization parameters; further research will address how to choose these parameters.

Acknowledgement

We thank our colleagues Martin S. Andersen for invaluable help with the optimization algorithms and Jakob Sauer Jørgensen for help with the real data computations.

Funding

This work is part of the project HD-Tomo funded by Advanced Grant No. 291405 from the European Research Council.

References

[1] Yoon S, Pineda AR, Fahrig R. Simultaneous segmentation and reconstruction: A level set method approach for limited view computed tomography. Med Phys. 2010;37(5):2329–2340.

[2] Batenburg KJ, Sijbers J, Poulsen HF, Knudsen E. DART: a robust algorithm for fast reconstruction of three-dimensional grain maps. J Appl Cryst. 2010;40:1464–1473.

[3] Batenburg KJ, Sijbers J. DART: a practical reconstruction algorithm for discrete tomography. IEEE Trans Image Process. 2011;20(9):2542–2553.

[4] Depypere M, Nuyts J, Laperre K, Carmeliet G, Maes F, Suetens P. The minimal entropy prior for simultaneous reconstruction and segmentation of in vivo microCT trabecular bone images. In: Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009. IEEE; 2009. p. 586–589.

[5] Ramlau R, Ring W. A Mumford-Shah level-set approach for the inversion and segmentation of x-ray tomography data. J Comput Phys. 2007;221:539–557.

[6] Klann E, Ramlau R, Ring W. A Mumford-Shah level-set approach for the inversion and segmentation of spect/ct data. Inverse Problems and Imaging. 2011;5:137–166.

[7] Klann E. A Mumford-Shah-like method for limited data tomography with an application to electron tomography. SIAM J Imaging Sciences. 2011;4:1029–1048.

[8] Van de Sompel D, Brady M. Simultaneous reconstruction and segmentation algorithm for positron emission tomography and transmission tomography. In: Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro; 2008. p. 1035–1038.

[9] Marroquin JL, Santana EA, Botello S. Hidden Markov measure field models for image segmentation. IEEE Trans Pattern Anal Mach Intell. 2003;25(11):1380–1387.

[10] Mohammad-Djafari A. Gauss-Markov-Potts priors for images in computer tomography resulting to joint optimal reconstruction and segmentation. International Journal of Tomography and Statistics. 2009;11:76–92.

[11] T L, Blaickner M, Knäusl B, Georg D, Neuwirth J, Baum R, Schuchardt C, Wiessalla S, Matz G. PET image segmentation using a Gaussian mixture model and Markov random fields. EJNMMI Physics. 2015;2:9.

[12] Hansen PC. Discrete Inverse Problems: Insight and Algorithms. SIAM, Philadephia; 2010.

[13] Bracewell RN, Riddle A. Inversion of fan-beam scans in radio astronomy. The Astrophysical Journal. 1967;150:427.

[14] Kak AC, Slaney M. Principles of Computerized Tomographic Imaging. SIAM, Philadelphia; 2001.

[15] Hansen PC, Saxild-Hansen M. AIR Tools – a MATLAB package of algebraic iterative reconstruction methods. J Comput Appl Math. 2012;236(8):2167–2178.

[16] Bian J, Siewerdsen JH, Han X, Sidky EY, Prince JL, Pelizzari CA, Pan X. Evaluation of sparse-view reconstruction from flat-panel-detector cone-beam CT. Physics in Medicine and Biology. 2010;55(22):6575.

[17] Sidky EY, Kao CM, Pan X. Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT. Journal of X-ray Science and Technology. 2006;14(2):119–139.

[18] Strong D, Chan T. Edge-preserving and scale-dependent properties of total variation regularization. Inverse Problems. 2003;19(6):S165.

[19] Sethian JA. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press; 1999.

[20] Osher S, Paragios N. Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer; 2003.

[21] Bae KT, Giger ML, Chen CT, Kahn Jr CE. Automatic segmentation of liver structure in CT images. Medical Physics. 1993;20(1):71–78.

[22] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. Int J Comp Vis. 1988; 1(4):321–331.

[23] Boykov YY, Jolly MP. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In: Proceedings of the Eighth IEEE International Conference on Computer Vision; Vol. 1. IEEE; 2001. p. 105–112.

[24] Boykov Y, Funka-Lea G. Graph cuts and efficient ND image segmentation. Int J Comp Vis. 2006;70(2):109–131.

[25] Brox T, Rousson M, Deriche R, Weickert J. Colour, texture, and motion in level set based segmentation and tracking. Image and Vision Computing. 2010;28(3):376–390.

[26] Mairal J, Bach F, Ponce J, Sapiro G, Zisserman A. Discriminative learned dictionaries for local image analysis. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008; 2008. p. 1–8.

[27] Gao Y, Bouix S, Shenton M, Tannenbaum A. Sparse texture active contour. IEEE Transactions on Image Processing. 2013;22(10):3866–3878.

[28] Mortensen EN, Barrett WA. Interactive segmentation with intelligent scissors. Graphical Models and Image Processing. 1998;60(5):349–384.

[29] Buzug T. Computed tomography: From photon statistics to modern cone-beam ct. Springer, NY; 2010.

[30] Mueller J, Siltanen S. Linear and nonlinear inverse problems with practical applications. SIAM, Philadephia; 2012.

[31] Bardsley JM, Nagy JG. Covariance-preconditioned iterative methods for nonnegatively constrained astronomical imaging. SIAM Journal on Matrix nalysis and Applications. 2006; 27(4):1184–1197.

[32] Björck Å. Numerical Methods for Least Squares Problems. SIAM, Philadelphia; 1996.

[33] Bertsekas DP. Nonlinear Programming, 2. ed. Athena Scientific, Belmont, MA; 1999.

[34] Jensen TL, Jørgensen JH, Hansen PC, Jensen SH. Implementation of an optimal first-order method for strongly convex total variation regularization. BIT Numer Math. 2012;52(2):329–

356.

[35] Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts. IEEE Trans Pattern Anal Mach Intell. 2001;23(11):1222–1239.

[36] Peixoto TP. The graph-tool Python library. 2014;Available from: http://figshare.com/articles/graph_tool/1164194.