IOP Publishing Phys. Med. Biol. 63 (2018) 235001 (19pp) https://doi.org/10.1088/1361-6560/aaeaf2

Physics in Medicine & Biology

IPEM Institute of Physics and Engineering in Medicine

CrossMark

RECEIVED
5 July 2018

REVISED
22 October 2018

ACCEPTED FOR PUBLICATION
24 October 2018

PUBLISHED
22 November 2018

PAPER

Comparison of five one-step reconstruction algorithms for spectral CT

Cyril Mory[1], Bruno Sixou[2], Salim Si-Mohamed[2,3], Loïc Boussel[2,3] and Simon Rit[1,4]

[1] Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, Centre Léon Bérard, F-69373, Lyon, France
[2] Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, U1206, F-69621, Lyon, France
[3] Radiology Department, Hospices Civils de Lyon, Lyon, France
[4] Author to whom any correspondence should be addressed.

E-mail: simon.rit@creatis.insa-lyon.fr

Keywords: spectral CT, one step, computed tomography

Supplementary material for this article is available online

Abstract

Over the last decade, dual-energy CT scanners have gone from prototypes to clinically available machines, and spectral photon counting CT scanners are following. They require a specific reconstruction process, consisting of two steps: material decomposition and tomographic reconstruction. Image-based methods perform reconstruction, then decomposition, while projection-based methods perform decomposition first, and then reconstruction. As an alternative, 'one-step inversion' methods have been proposed, which perform decomposition and reconstruction simultaneously. Unfortunately, one-step methods are typically slower than their two-step counterparts, and in most CT applications, reconstruction time is critical. This paper therefore proposes to compare the convergence speeds of five one-step algorithms. We adapted all these algorithms to solve the same problem: spectral photon-counting CT reconstruction from five energy bins, using a three materials decomposition basis and spatial regularization. The paper compares a Bayesian method which uses non-linear conjugate gradient for minimization (Cai *et al* 2013 *Med. Phys.* **40** 111916–31), three methods based on quadratic surrogates (Long and Fessler 2014 *IEEE Trans. Med. Imaging* **33** 1614–26, Weidinger *et al* 2016 *Int. J. Biomed. Imaging* **2016** 1–15, Mechlem *et al* 2018 *IEEE Trans. Med. Imaging* **37** 68–80), and a primal-dual method based on MOCCA, a modified Chambolle–Pock algorithm (Barber *et al* 2016 *Phys. Med. Biol.* **61** 3784). Some of these methods have been accelerated by using μ-preconditioning, i.e. by performing all internal computations not with the actual materials the object is made of, but with carefully chosen linear combinations of those. In this paper, we also evaluated the impact of three different μ-preconditioners on convergence speed. Our experiments on simulated data revealed vast differences in the number of iterations required to reach a common image quality objective: Mechlem *et al* (2018 *IEEE Trans. Med. Imaging* **37** 68–80) needed ten iterations, Cai *et al* (2013 *Med. Phys.* **40** 111916–31), Long and Fessler (2014 *IEEE Trans. Med. Imaging* **33** 1614–26) and Weidinger *et al* (2016 *Int. J. Biomed. Imaging* **2016** 1–15) several hundreds, and Barber *et al* (2016 *Phys. Med. Biol.* **61** 3784) several thousands. We also sum up other practical aspects, like memory footprint and the need to tune extra parameters.

1. Introduction

1.1. Data acquisition

Dual energy computed tomography (CT) systems are now commercially available, and spectral photon counting CT systems already exist as research prototypes. These new types of scanners provide information on the energy distribution of the x-ray photons that reach the detector: dual energy systems acquire two sets of projections (for low and high energies), while spectral photon counting systems group the incoming photons into so-called

'energy bins', which form a partition of the incident spectrum's bandwidth. For each projection's pixel, the 'raw' measurements acquired by a spectral photon counting CT scanner are therefore the number of photons in each energy bin, which are commonly referred to as 'photon counts'.

1.2. Attenuation model

Assuming that the object's attenuation can be obtained by linear combination of the attenuations of only a few materials, this energy-resolved information allows to reconstruct multiple volumes, each one representing a different material's concentration map. Alternatively, one can assume that the attenuation stems from only a few physical phenomena, e.g. Compton scattering, photo-electric effect and K-edge (Alvarez and Macovski 1976, Pan *et al* 2010), and reconstruct maps of the characteristic quantities of each of these physical phenomena. For the sake of readability, we present the case of spectral photon-counting CT, and decompose on a basis of materials, but all calculations are also valid for dual-energy CT, and with photo-electric effect and Compton scattering as decomposition basis. It is assumed that the linear attenuation coefficient $\tilde{\mu}_{je}$ of the imaged object in voxel j and at energy e can be separated into a function of space and a function of energy, i.e.

$$\tilde{\mu}_{je} = \sum_{m=1}^{N_m} \mu_{me}\, x_{jm} \tag{1}$$

where N_m is the number of materials, μ_{me} the mass-attenuation coefficient of material m at energy e and x_{jm} the concentration of material m in voxel j. The attenuation profiles $(\mu_{me})_{e=1..150\ \mathrm{keV}}$ are known and the aim is to find the concentration maps x_m.

1.3. Reconstruction methods

The most widespread methods to reconstruct material-specific volumes from photon counts can be divided into two categories: image-based and projection-based. In two-step image-based methods, each energy bin of the photon counts is log-transformed and reconstructed, just like the projections dataset of a standard CT. This process yields one volume per energy bin and these intermediate volumes are then decomposed into material-specific volumes (Maass *et al* 2009, Niu *et al* 2014). Unfortunately, the intermediate volumes are corrupted by beam hardening artefacts (Taguchi *et al* 2018), because the energy dependence of the attenuation coefficients is still averaged in each energy bin. Image-based methods often integrate some form of empirical beam-hardening correction (Maass *et al* 2009), but perfect beam-hardening correction would require knowledge of the material volumes. In two-step projection-based methods, on the other hand, multi-channel projections are first decomposed into material-specific projections, and then reconstructed, independently (Alvarez and Macovski 1976, Schlomka *et al* 2008) or jointly (Sawatzky *et al* 2014, Mory *et al* 2018). It turns out that for typical choices of materials (e.g. water, bone, and a high-Z contrast agent), the normalized attenuation profiles of the materials do not differ enough to yield a robust decomposition: the inevitable statistical noise on photon counts is often sufficient to cause aberrant material line integrals, which result in strong streak artefacts in the reconstructed volumes. Some methods were proposed to regularize the decomposition process (Brendel *et al* 2016, Ducros *et al* 2017), but how to regularize in the projection domain is a difficult problem: there are conditions for projections to be consistent, i.e. to be those of an actual object, called data-consistency conditions (DCC) (Clackdoyle *et al* 2016, Lesaint *et al* 2017), and regularization with priors used routinely for volumes (e.g. total variation) can generate inconsistent projections. In addition, projection-based methods require the spectral projections to have the same geometry (same source and detector positions), it cannot be applied to fast kV switching (So and Lee 2015) or dual source dual-energy systems (Flohr *et al* 2006) easily: corresponding projections must be interpolated, which is a potential source of errors.

Two-step methods also have a structural drawback: unless the first step provides a one-to-one mapping between inputs and outputs, it implies a loss of information, for which the second step cannot compensate. The aberrant pixels observed after projection-based decomposition illustrate that loss of information: the reconstruction step cannot go back to the photon counts, and therefore must rely on aberrant decomposition results. Recently, several methods have been proposed which reconstruct material-specific volumes directly from the photon counts (Cai *et al* 2013, Long and Fessler 2014, Barber *et al* 2016, Weidinger *et al* 2016, Chen *et al* 2017, Mechlem *et al* 2018). They are commonly referred to as 'one-step inversion', or simply 'one-step' methods. All of these methods are iterative: there is currently no analytical inversion formula for the material decomposition problem, let alone for one-step inversion. They consist in combining the forward models of the tomographic reconstruction and the material separation inverse problems, yielding a single (but more complex) forward model, which takes as input a set of material volumes and yields photon counts as output. In discrete form, the forward model reads

$$\bar{y}_{ib} = \sum_{e} s_{be} \exp\left(-\sum_{j} a_{ij} \sum_{m} \mu_{me}\, x_{jm}\right) \tag{2}$$

where $\bar{y}_{ib}$ is the expected photon count in pixel i and bin b, s_{be} is the effective spectrum, i.e. the number of photons of actual energy e that are expected to be detected in bin b if there was no attenuating object and a_{ij} is the coefficient of the forward projection matrix A at row i and column j. One-step inversion then means finding the volumes x_m for which the forward model yields photon counts y as similar as possible to those measured by the scanner. By construction, the drawbacks of two-step methods are circumvented, and there is no risk of losing information. One-step methods have been reported to yield improved image quality (Zhang *et al* 2014) and reduced noise-induced bias (Mechlem *et al* 2018) with respect to their two-step counterparts, but they are also slower to converge. This is likely to prevent their use in many applications, as it happened for regularized iterative reconstruction techniques on standard CT (Pan *et al* 2009). Since in the multi-energy CT case the amount of data contained in a single acquisition is multiplied by the number of energy bins, and the one-step problem is more complex than the reconstruction problem alone, the need for fast methods is even more acute than for standard CT.

Among the methods studied in this paper (Cai *et al* 2013, Long and Fessler 2014, Barber *et al* 2016, Mechlem *et al* 2018), only Cai *et al* (2013) reported a comparison with another one-step method (Sukovic and Clinthorne 2000) and an advanced image-based two-step method (Maass *et al* 2011). Barber *et al* (2016) showed many results, but only with the proposed method, and the other ones (Long and Fessler 2014, Weidinger *et al* 2016, Mechlem *et al* 2018) are compared to two-step methods, either projection-based or image-based (Mendonca *et al* 2014), with the exception that Mechlem *et al* (2018) contains a comparison with the method of Weidinger *et al* (2016) in terms of convergence speed. An independent comparison between many one-step methods is therefore still missing, although it would certainly benefit the spectral CT scanner manufacturers, saving them the hassle of investigating many methods to implement only one. We provide such a comparison in this paper, with experiments conducted on simulated data on a single slice. Since we provide the code we used, it will also help the research community: ranking a new one-step method against the ones studied in this paper will only require to implement the new method in the open source framework we provide.

All methods were first adapted to solve the same one-step problem: spectral photon-counting CT reconstruction from five energy bins using a three materials decomposition basis and spatial regularization. Since the five methods we studied use different data-attachment and regularization terms, as well as different optimization techniques, we defined an image quality objective, independant of the cost functions, and measured the number of iterations required to reach it with each method. These results constitute the main contribution of the present paper. For each method, we also study its memory footprint, state whether it requires to tune additional parameters, and whether or not the derivation of the algorithm uses simplifying assumptions/approximations.

In Barber *et al* (2016) and Fessler (2003), it is proposed to perform internal computations not with the actual materials into which one wishes to decompose the object, but with carefully chosen linear combinations of these materials. Since the materials are characterized only by their attenuation coefficient μ as a function of the incident energy, Barber *et al* call this step μ-preconditioning. In this paper, we present three different ways to perform μ-preconditioning and evaluate their impact on convergence speed. Since most methods were not explicitly designed for μ-preconditioning, a small adaptation is required (and described in the present paper) for the regularization step.

2. Material and methods

2.1. Compared methods

We have studied five one-step reconstruction methods, adapted them to solve the three-material, five-bin, unconstrained and regularized one-step spectral reconstruction problem, and compared their behavior. These methods are:

- Cai2013 (Cai *et al* 2013), which uses non-linear conjugate gradient with a heuristical step size,
- Long2014, Weidinger2016 and Mechlem2018 (Long and Fessler 2014, Weidinger *et al* 2016, Mechlem *et al* 2018), which use Separable Quadratic Surrogates (SQS) to minimize cost functions,
- Barber2016 (Barber *et al* 2016), which relies on a generalization of the Chambolle–Pock algorithm (Chambolle and Pock 2011).

All methods use the polychromatic Beer–Lambert law as forward model (the way to estimate the expected photon counts $\bar{y}$ from material maps x_m) which is reminded in its discrete form in equation (2). They slightly differ by the cost functions they attempt to minimize, which are summarized in table 1. The data-attachment terms differ because they model the noise differently: Cai2013 optimizes a Gaussian observation model in a Bayesian framework where all the others minimize the Poisson negative log-likelihood. All regularization terms are different but share some similarities, involving the $\ell 1$-norm (or a twice-differentiable approximation of it) of the spatial gradient of each reconstructed volume.

Since three methods out of five use them, it is useful to remind the definition of Separable Quadratic Surrogates (SQS) and how they are used. Surrogates are defined as follows: let $\Psi : \mathbb{R}^N \to \mathbb{R}$ and $x_0 \in \mathbb{R}^N$. The function $\Phi_{x_0} : \mathbb{R}^N \to \mathbb{R}$ is a surrogate of Ψ at x_0 if and only if

Table 1. Summary of the cost functions minimized by the studied methods. ϕ^{Huber}, ϕ^{Long} and ϕ^{Green} are twice-differentiable approximations of the absolute value function, defined in equations (4)–(6), respectively. ∇ is the 3-components spatial gradient, and the neighborhood $\mathcal{N}$ is arbitrary, but always set to all 26 immediate neighbors of the voxel v. The regularization of Barber2016 uses the convex indicator function χ to enforce an upper bound on the total variation $\|(|\nabla x|)\|_1$ of x. The numerical values of the parameters of the cost function used for the simulated experiment are listed in table 2.

	Cost function $\mathcal{D}(x) + \sum_m \mathcal{R}(x_m, w_m)$		
Method	Parameters of the cost function	Data attachment $\mathcal{D}$	Regularization $\mathcal{R}$
Cai2013	k_d, w_m, δ_m	$\|W(y-\bar{y})\|_2^2 + \log(\bar{y})$ with $W = \text{diag}[k_d\bar{y}]^{-1/2}$	$w_m \phi^{\text{Huber}}(\nabla x, \delta_m)$
Long2014	w_m, δ_m		$w_m \sum_{v=1}^{N_v} \sum_{\tilde{v}\in\mathcal{N}} \phi^{\text{Long}}(z_{k-1,v} - z_{k-1,\tilde{v}}, \delta_m)$
Weidinger2016	w_m	$\bar{y} - y\log\bar{y}$	$w_m \sum_{v=1}^{N_v} \sum_{\tilde{v}\in\mathcal{N}} \phi^{\text{Green}}(z_{k-1,v} - z_{k-1,\tilde{v}})$
Mechlem2018	w_m, δ_m		$w_m \sum_{v=1}^{N_v} \sum_{\tilde{v}\in\mathcal{N}} \phi^{\text{Huber}}(z_{k-1,v} - z_{k-1,\tilde{v}}, \delta_m)$
Barber2016	w_m		$\chi(x, w_m) = \begin{cases} 0 & \text{if } \|(|\nabla x|)\|_1 \leqslant w_m, \\ +\infty & \text{else.} \end{cases}$

$$\begin{cases} \forall x \in \mathbb{R}^N, \Phi_{x_0}(x) \geqslant \Psi(x) \\ \Phi_{x_0}(x_0) = \Psi(x_0), \Phi'_{x_0}(x_0) = \Psi'(x_0) \end{cases} \tag{3}$$

i.e. if Φ_{x_0} is above Ψ on $\mathbb{R}^N$, and tangent to Ψ at x_0. Note that we have used $\mathbb{R}^N$ as domain for simplicity, but in Long2014 the domain actually used is $\mathbb{R}^{+N}$. The algorithms based on surrogates consist in finding a surrogate Φ_{x_0} to the cost function Ψ at the current iterate x_0, minimizing the surrogate instead of the original cost function, adopting that minimizer as the new iterate x_1, and starting over. Obviously, the surrogates must be chosen carefully in order to be easier to minimize than the original cost function. In Long and Fessler (2014), Weidinger *et al* (2016) and Mechlem *et al* (2018), surrogates are derived one after the other: first a convex one, then a quadratic one, and finally a separable quadratic one, each new one being a surrogate of all the previous ones, and therefore of the original cost function.

2.2. Preliminary comments

We have tried to stick as much as possible to the original algorithms, but some modifications were required. In this section, we briefly introduce each method, and describe the modifications we had to perform. Though we have tried to make the present paper as self-contained as possible, it is impossible to explain all the methods in details without ending up duplicating large passages of the original papers. The detailed maths can be found in our notes (see section 2.7), in which we guide the reader through the equations and re-do most calculations, for all methods but Barber2016. For Barber2016, our notes only provide implementation suggestions, because the original paper is already written in a very didactic way. However, since we had to adapt the five studied methods to deal with a common problem, we provide the pseudo-code of all methods in appendix. In these pseudo-codes, as much as possible, we used common notations for all methods, in particular for the matrices that describe detector response, incident spectrum and material attenuations. A drawback of this approach is that the pseudo-code notations no longer correspond to those of the original papers. Throughout the paper, when referring to variable names, we use the ones from the pseudo-code. We use a contraction of the first author's name and publication year (e.g. 'Cai2013') to refer to each method. When the distinction is relevant, it should be clear from the context whether we mean the method implemented and modified by us, or the one described in the original papers.

2.2.1. *Cai2013*

Cai2013 (Cai *et al* 2013) was originally published for dual energy CT. The cost function is built with a variable y that is the ratios between photon counts with object and photon counts without object (instead of just the photon counts with object, in the other methods). The variance of these ratios y is assumed to be proportional to y, with a factor k_d that has to be estimated. We generalized Cai2013 (Cai *et al* 2013) to 3 materials, which is straightforward, and k_d is determined by a simulation with the same spectrum, detector response and geometry as in the experiments, but without object. It yields $k_d = 1.59 \times 10^{-4}$. The regularization term is the Huber function of the spatial gradient of the volumes, with the Huber function defined as

$$\phi^{\text{Huber}}(x, \delta) = \begin{cases} x^2 & \text{if } |x| < \delta, \\ 2\delta|x| - \delta^2 & \text{else.} \end{cases} \tag{4}$$

The cost function is minimized by a non-linear conjugate gradient, with a heuristically determined step size. Given the descent direction, the step size is computed as the one that would minimize the cost function's second-order Taylor expansion. Then the cost function is computed at the candidate destination point, and the step size divided by 2 if it causes an increase of the cost instead of a decrease. According to the authors, the case where the step length had to be divided by 2 never occurred in the experiments they conducted (which were limited to dual-energy CT). In ours, it did occur, and we had to implement a reset mechanism: if, after ten attempts to divide the step length by 2, we still obtain an increase of the cost function, we assume that the descent direction yielded by non-linear CG is erroneous, and try using the opposite gradient instead, as in an ordinary gradient descent. Since it is an ad hoc adaptation, and only for a special case, we have not included it into the pseudo-code in the appendix.

2.2.2. *Long2014*

Long2014 (Long and Fessler 2014) defines the following hyperbola for approximating the absolute value function in the regularization:

$$\phi^{\mathrm{Long}}(x,\delta) = \frac{\delta^2}{3}\left(\sqrt{1+3\left(\frac{x}{\delta}\right)^2}-1\right). \tag{5}$$

The authors sequentially derive the surrogates of the cost function to minimize it: first a convex one, then a quadratic one, and finally a separable quadratic one. Deriving the quadratic surrogate requires to assume that the material maps are non-negative. It was originally published as a method for reconstructing three or more materials from dual energy CT scans. To that end, it adds a lot of *a priori* information, in the form of equality and inequality constraints on the reconstructed volumes. Performing minimization while complying with all constraints requires an inner loop of quadratic programming using the generalized sequential minimal optimization (GSMO) algorithm (Keerthi and Gilbert 2002). Since in our case, there are more bins than the number of materials, we have removed all constraints in our implementation of Long2014, which allowed to replace the quadratic programming inner loop by a single iteration of Newton's method. Long2014 uses ordered subsets (OS) to speed up convergence. See section 2.4 for a quick reminder on OS.

2.2.3. *Weidinger2016*

Weidinger2016 (Weidinger *et al* 2016) uses the Green potential (Green 1990) for approximating the absolute value function in the regularization:

$$\phi^{\mathrm{Green}}(x) = \frac{27}{128}\log\cosh\left(\frac{16}{3\sqrt{3}}x\right). \tag{6}$$

In Weidinger2016, as in Long2014, surrogates are derived sequentially. However, contrary to Long2014, Weidinger2016 does not assume that the material maps are non-negative, and therefore cannot find a rigorous quadratic surrogate. The quadratic function is actually not a surrogate of the convex surrogate, but its second-order Taylor approximation. This means that, if its derivative is negative, that quadratic function violates the conditions for being a surrogate. Such a choice simplifies the calculations, but the method is no longer guaranteed to converge. In our experiments, though, it did converge.

2.2.4. *Mechlem2018*

Mechlem2018 (Mechlem *et al* 2018) builds upon Weidinger2016 using the Huber function for the regularization (equation (4)), meaning it also uses the second-order Taylor approximation of the convex surrogate as if it were an actual quadratic surrogate, which implies the same risk on convergence as in Weidinger2016. Still, it adds three features with respect to Weidinger2016: calibration, Ordered Subsets and Nesterov acceleration. The calibration part allows to estimate the product of the incident spectrum and detector response, in cases where it is either not known at all, or known with insufficient accuracy. In this study, we have assumed perfect knowledge of these system characteristics, and therefore dropped the calibration part. In addition to that, Mechlem2018 (Mechlem *et al* 2018) suggests to compute the Hessian (and its inverse) only once, thus updating only the gradient at each iteration. This approach is only valid if the Hessian is near-constant, i.e. if the estimated volumes undergo little changes over the course of the iterations. Therefore, the paper proposes to start the iterations from a previously known approximate result, like a set of two-step reconstructed volumes. Since we start from zero-filled estimates, we do not use this acceleration.

2.2.5. *Barber2016*

Barber2016 (Barber *et al* 2016) provides the calculations to minimize either a transmission Poisson likelihood or a least-squares data-attachment term. We chose to focus on the transmission Poisson likelihood term. As suggested by Sidky *et al* (2018), we have replaced the exponential function in equation (2) with the soft exponential defined by

$$\mathrm{softexp}(x) = \begin{cases} \exp(x) & \text{if } x \leqslant 0, \\ x+1 & \text{else.} \end{cases} \tag{7}$$

Regularization is handled with a total variation (TV) term, and the cost function is minimized using the mirrored convex/concave (MOCCA) algorithm (Barber and Sidky 2015). MOCCA is a primal-dual algorithm designed to minimize non-convex functions that can be expressed as the difference of two convex functions. As explained in section 3.1 of Barber *et al* (2016), the algorithm should have an outer and an inner loop, but the inner loop is set to perform only a single iteration to speed up convergence. Unfortunately, that modification can cause the algorithm to diverge, so an additional parameter λ has been introduced and tuned to avoid that. Roughly, λ is tuned to the largest value that does not cause divergence. We also adapted the parameter θ of the Chambolle–Pock algorithm (see the last line of equation 7 in Chambolle and Pock (2011)), which must lie in [0;1], and in Barber *et al* (2016) is implicitly set to 1. With $\theta = 1$, examining the sequence of iterates, we observed fast and ample oscillations, and artifacts constantly appearing and disappearing. These oscillations and artifacts did not significantly decrease in intensity over the course of the iterations. Setting $\theta = 0.5$ got rid of them, yielding a more stable sequence of iterates. Computing the primal and dual step sizes Σ and T involves the element-wise absolute value of products of matrices, which is not trivial if both matrices have positive and negative values and are too large to be actually stored. The computation of the step sizes proved particularly tedious to implement, so we restricted our study to a μ-preconditioning method (see section 2.3) that does not generate negative attenuation values (i.e. simply normalizing the attenuation coefficient vectors), which makes simplifications possible. This μ-preconditioning is different from the one presented in the original paper (Barber *et al* 2016) (i.e. orthonormalizing the attenuation coefficient vectors), but our study on Cai2013 in section 3.4 shows a better convergence speed for normalization than for orthonormalization, so it is unlikely that switching to orthonormalization for Barber2016 would greatly improve its convergence speed.

2.3. Mu-preconditioning

We form the matrix M as follows:

$$M = \begin{pmatrix} \mu_{\mathrm{I},1\ \mathrm{keV}} & \mu_{\mathrm{Gd},1\ \mathrm{keV}} & \mu_{\mathrm{H_2O},1\ \mathrm{keV}} \\ \vdots & \vdots & \vdots \\ \mu_{\mathrm{I},150\ \mathrm{keV}} & \mu_{\mathrm{Gd},150\ \mathrm{keV}} & \mu_{\mathrm{H_2O},150\ \mathrm{keV}} \end{pmatrix} \tag{8}$$

where μ_{me} is the mass-attenuation coefficient of material m for incident x-ray photons of energy e, I stands for iodine, Gd for gadolinium and H_2O for water. For $e \leqslant 150$ keV, $\mu_{\mathrm{H_2O},e}$ and $\mu_{\mathrm{Gd},e}$ typically differ by 2 orders of magnitude, which causes conditioning issues in the resolution of the one-step inversion problem, and slows down convergence. Section 3.4 of Barber *et al* (2016) proposes to modify the material attenuation matrix M to speed up and stabilize the internal computations. They call this step 'μ-preconditioning'. It can be interpreted as creating synthetic materials, which are linear combinations of the original ones, possibly with negative coefficients. We present three different μ-preconditioning methods. Each one yields a synthetic material attenuation matrix $\widetilde{M}$ and a matrix P such that $MP = \widetilde{M}$:

- Normalization: $\widetilde{M}$ is obtained by dividing each column of M by its norm. This has the advantage of not introducing negative attenuations. The added complexity associated with the use of negative attenuation coefficients, like the use of absolute values in the calculation of Σ and T in Barber *et al* (2016), can be avoided with this synthetic materials basis.
- Orthonormalization: $\widetilde{M}$ is obtained either using the Gram–Schmidt algorithm on M, or diagonalizing M^TM (as proposed in Barber *et al* (2016)). This will generate negative attenuation coefficients (for the dot products between columns to be zero).
- Fessler's method:

$$K = \frac{SM}{S\mathbb{1}} \quad \text{i.e.} \quad (K)_{b,m} = \frac{\sum_e s_{be}\,\mu_{me}}{\sum_e s_{be}} \tag{9}$$

with s_{be} the effective spectrum at energy e in bin b, and μ_{me} the attenuation of material m at energy e. Each value $(K)_{b,m}$ is a weighted mean of the attenuation coefficients of material m over the whole energy spectrum, the weights being the number of photons seen by the detector in bin b. Roughly, it evaluates how much the material m attenuates the photons that will end up being detected in bin b. Now, the problem of determining which linear combination of materials causes the observed attenuation would be much easier if each material attenuated the photons of only one bin, i.e. if K was diagonal. Therefore, we define $\widetilde{M} = MK^{-1}$, i.e. $P = K^{-1}$. When K is not square, e.g. when there are more bins that materials, K cannot be inverted, but P can be defined as the Moore–Penrose pseudo inverse of K, i.e. $P = (K^TK)^{-1}K^T$. This approach was proposed in Fessler (2003) for dual energy CT, but its extension to more materials and more bins is straightforward.

When the algorithm is fed a matrix $\widetilde{M} = MP$ of the attenuations of synthetic materials, it reconstructs synthetic material volumes $\tilde{x}$. No change is required for the data-attachment term, but the regularization must be perfomed on the original materials volumes $x = P\tilde{x}$, since we want the original material volumes to be regular, not the synthetic ones. The changes in algorithms implied by μ-preconditioning appear in the pseudo code of Cai2013 and Barber2016 (not in the SQS methods, since they are insensitive to μ-preconditioning) in section 5.

2.4. Ordered subsets

In some methods, iterations are divided into sub-iterations, and during each sub-iteration, only a subset of the projections through the current volume is computed, compared to the measured ones, and back-projected to obtain an update for the current volume (Hudson and Larkin 1994). The subsets form a partition of the full projections dataset. After one iteration, i.e. one cycle through all subsets, each projection has been forward and back projected once and only once. At each iteration, OS-methods therefore perform as many updates of the reconstructed volume as the number of subsets, which results in faster convergence per iteration (or equivalently, per forward/back projection), at the cost of longer iterations, and potential instabilities if the measured data contains inconsistencies, e.g. noise or motion.

2.5. Simulation setup

We have designed a simple two-dimensional 3-materials phantom (see figure 1), consisting of a large square of water at 1 g ml^{-1}, a small square of iodine at 10 mg ml^{-1}, and a small square of gadolinium at 10 mg ml^{-1}. The iodine and gadolinium squares are inside the water square, but do not overlap. Note that the K-edge of iodine is at 33.2 keV, which is too low to reliably use it as a K-edge contrast agent. On the other hand, the basis water/iodine can be used to represent biological samples (Si-Mohamed *et al* 2017), just like the Compton/photoelectric basis, with the bones appearing in the iodine material map. Therefore, we emphasize that water/iodine/gadolinium is a 'two materials for biological tissues, one K-edge contrast agent' situation, and not a dual contrast agent one. The phantom consists of a single slice, which has 256^2 voxels, of size 1 mm $\times$ 1 mm. Through this phantom, 725 parallel projections were simulated, with 362 pixels per projection, using the sparse forward projection matrix generated by the AIR toolbox (Hansen and Saxild-Hansen 2012). Detector pixels too have a size of 1 mm. The line integrals obtained were then converted to photon counts, following the classical polychromatic Beer–Lambert attenuation law. For the incident spectrum, we used the same one as in Ducros *et al* (2017) which, quoting the paper, 'was simulated using the SpekCalc software (Poludniowski *et al* 2009), considering a tube voltage of 120 kV, a 12° anode angle, and a filter of 1.2 mm of Al'. Figure 2 shows the number of incident photons per keV on each pixel.

The detector response was simulated according to the model presented in appendix A.2 of Schlomka *et al* (2008), and is shown in figure 3. Following the design of the Philips spectral CT scanner prototype installed in Lyon, France, 5 bins were simulated. The thresholds were set to 30, 51, 62, 72 and 83 keV.

In the end, the photon counts were corrupted with Poisson noise. This model therefore neglects pile-up, scatter, charge sharing and probably many other complex effects. Since the problem is non-convex, initialization matters. It is unclear what should be the starting point of the iterations: in the original papers, Cai2013 and Barber2016 use zero-filled material slices as a starting point, while the SQS-based methods are intialized with a pre-existing reconstruction (e.g. one obtained from a two-step method). In this study, we have adopted a zero initialization for all methods.

2.6. Comparison method

In any iterative reconstruction method, the convergence speed, image quality at convergence and regularization parameters depend on each other. Therefore, in order to obtain meaningful comparison results on convergence speed, one must first set the regularization parameters to values that yield comparable image quality at convergence for all methods. Only then can one measure how fast a given convergence criterion is reached. To set the regularization parameters, we ran many simulations, exploring the space of parameters with multiplicative steps of 3 or 10. We then selected parameters yielding similar standard deviations in the iodine, gadolinium and water ROIs, as shown in table 3. The regularization parameters are summarized in table 2. Finally, the criterion we used for the convergence speeds comparison is the number of iterations each method required for all materials to reach their target concentration in their ROI, within either 20% or 10% tolerance. Besides that, we also provide:

- the concentrations of iodine and gadolinium in their respective ROIs over the course of the iterations,
- the time per iteration with our implementation. Measurements were performed on a machine equipped with an Intel Xeon E5-2620 v4 CPU running at 2.10 GHz and an Nvidia GeForce GTX Titan X, and running OpenSuse Leap 42.2 and Matlab R2017b,
- the mean and standard deviation (std) in each ROI at the last iteration, which were used to adjust regularization parameters,

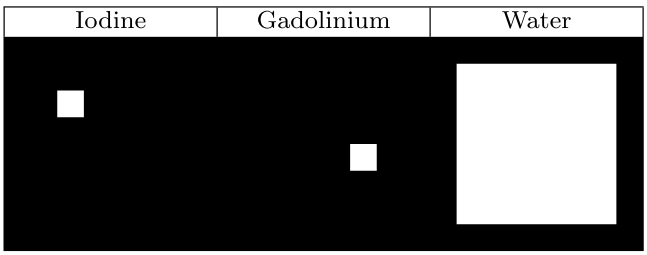

Figure 1. Ground truth of the phantom.

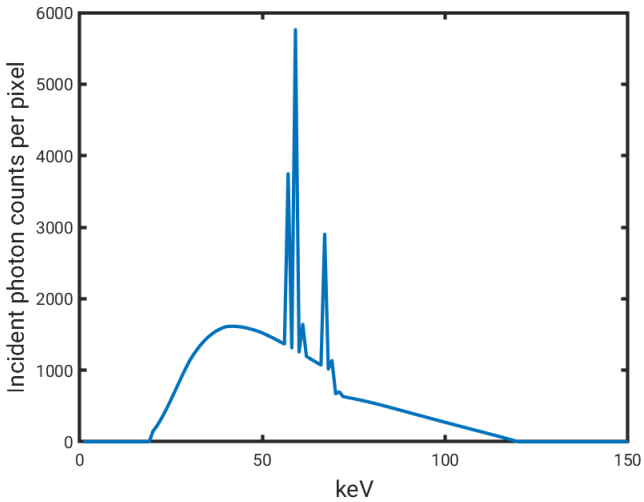

Figure 2. Incident spectrum on each pixel.

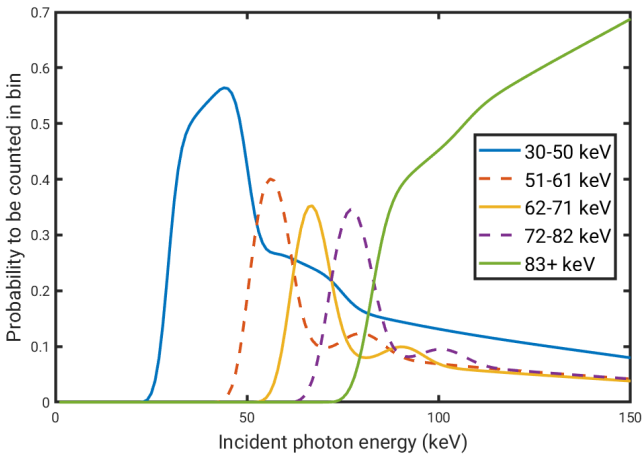

Figure 3. Spectral response of each detector bin.

- the theoretical memory footprint, since the actual memory footprint depends too much on the implementation.

Since storing all iterates is impractical, for all methods except Mechlem2018 we stored one iterate every ten, which is why table 4 contains mostly multiples of ten.

2.7. Software

For this study, we re-implemented all of the five methods compared here in a lightweight Matlab framework. That code is open source, available at https://github.com/SimonRit/OneStepSpectralCT, and we appreciate any relevant contribution to it. In particular, we encourage the authors of the five methods we have studied to check that code against their own implementation. Together with the code, we provide detailed notes on how to re-do most of the calculations of the original papers.

Table 2. Regularization parameters of the five methods used to obtain the reconstructions (see table 1). For Barber2016, the 'regularization' parameters are actually the TV limits for iodine, gadolinium and water. The δ parameter is used in the Huber function or its equivalent, in the regularization terms of the methods.

Method	δ_{iodine}	$\delta_{\text{gadolinium}}$	δ_{water}	w_{iodine}	$w_{\text{gadolinium}}$	w_{water}
Cai2013	0.001	0.001	0.1	100 000	100 000	30
Long2014	0.001	0.001	0.1	100 000	100 000	10
Weidinger2016	N/A	N/A	N/A	30 000	30 000	3
Barber2016	N/A	N/A	N/A	100	100	5000
Mechlem2018	0.001	0.001	0.1	30 000	30 000	3

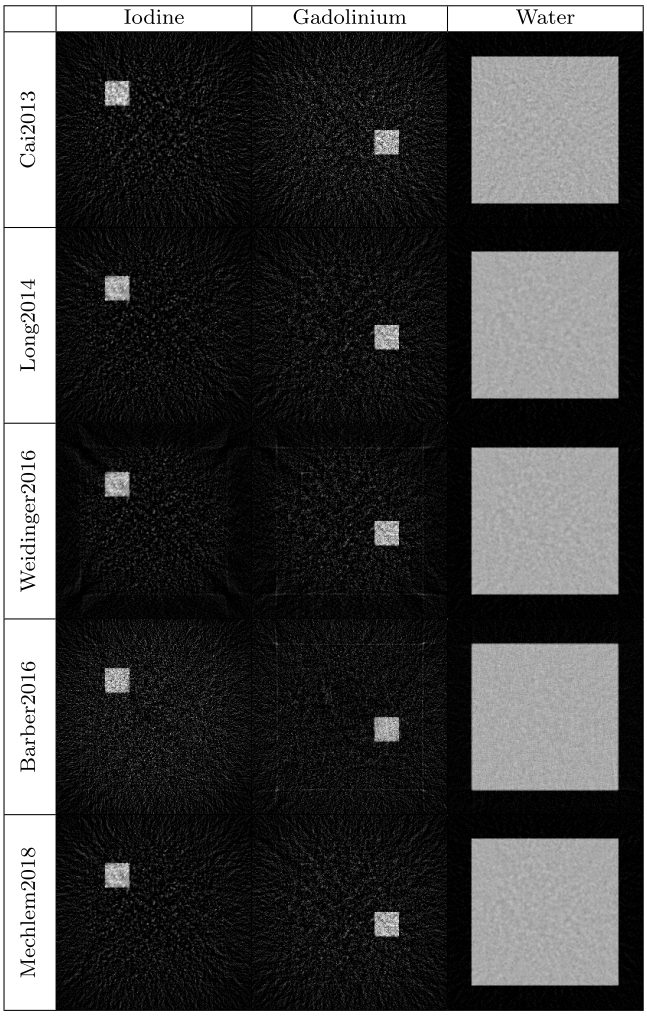

Figure 4. Reconstructed slices with the five methods, with grayscale ranges equal to [0; 15] mg ml^{-1} for iodine and gadolinium, and [0; 1.5] g ml^{-1} for water.

3. Results

Figure 4 shows the last iterate computed for each of the methods presented here. For Cai2013, we used Fessler's method for μ-preconditioning. A video of the iterates, on which it is easy to evaluate the relative convergence rates of the various methods, is available as supplementary material (stacks.iop.org/PMB/63/235001/mmedia). It has the same grayscale and is arranged the same way as figure 4.

3.1. Algorithm parameters

In addition to the quantitative criteria on convergence speed and memory footprint, a few other important aspects of the algorithms deserve to be mentioned here. All methods require parameters for regularization and Cai2013 needs a parameter k_d characterizing the detector for its data attachment term (see table 1). The rest of the parameters are free parameters for optimizing the cost function. Our implementation of Cai2013 requires a threshold on how many times one tries to halve the step size before resetting the descent direction (see

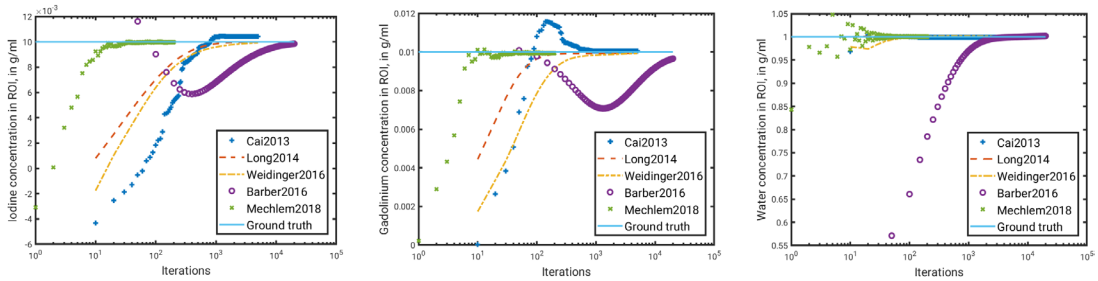

Figure 5. Concentration of iodine (left), gadolinium (middle) and water (right) over the course of the iterations for the five methods, in the ROI where it is expected eroded by 2 pixels.

section 2.2.1). Barber2016 has a parameter $\lambda = 10^{-4}$, and a parameter $\theta = 0.5$ (see section 2.2.5). Long2014 splits the projections into 20 subsets, and Mechlem2018 into four subsets, in order to speed up convergence. Mechlem2018 also uses Nesterov's momentum technique, which speeds up convergence even more, but makes it a bit unstable: when splitting the projections into too many subsets (typically more than six), Mechlem2018 sometimes diverges: the iterates accumulate strong streak artefacts, then they reach a point where the Hessian of the SQS is not invertible, which stops the algorithm.

3.2. Quantitative criteria

We show in figure 5 the evolution of the concentrations of iodine and gadolinium over the course of iterations, for all five methods, with a logarithmic scale only on the x-axis. The concentrations are measured in the square where iodine or gadolinium is expected, eroded by 2 pixels in order to avoid partial volume effects. In all methods, the pixels at the border of the region of interest (ROI) have significantly lower concentrations. The full line is the ground truth concentration value in the ROI. Since tracking the mean concentration in a ROI can only capture information on how well the low frequencies are reconstructed, figure 6 shows, for each method, the normalized $\ell 2$ difference with the last iterate, computed as follows:

$$\text{Normalized } \ell 2(x_k) = \sum_{m}^{N_m} \frac{\sum_{v=1}^{N_v} \left(x_{k,v,m} - x_{N_{\text{iterations}},v,m}\right)^2}{N_m \sum_{v=1}^{N_v} \text{GroundTruth}_{v,m}^2} \tag{10}$$

where k, v and m are respectively the indices for iteration, voxel and material, N_v is the number of voxels of one reconstructed material map and N_m the number of materials.

Table 3 contains the mean and standard deviation of iodine, gadolinium and water concentrations in the ROI where they are expected eroded by 2 pixels, on the final iterate. Standard deviations of each material are roughly the same between all algorithms, yet there are some outliers: the stds of Cai2013 in gadolinium and of Barber2016 in iodine. This is a consequence of the fact that those two methods are the slowest ones, and consequently we were not able to sample the parameter space for them with a fine enough grid. While it is unfortunate, it highlights the fact that slowly converging methods are hard to tune, and that additional parameters (λ in Barber2016) imply a longer tuning time.

Table 4 shows the total number of iterations performed, the time per iteration, and the number of iterations it took for all materials to reach target concentration in their respective ROIs, within 20% or 10% tolerance. On our test case, Mechlem2018 converges 1–3 orders of magnitude faster than its competitors.

3.3. Memory footprint of each method

We provide here a simple analysis of the memory footprint of each method, by listing the variables that have to be stored and recalling their size. The results are summarized in table 5, and the detailed calculations for each method are presented in this section. The measured photon counts, which constitute the main input of these reconstruction algorithms, are assumed to be available on disk, so they never have to be fully loaded in memory and are not mentioned in this section. Obviously, re-reading them from disk at each iteration is highly inefficient, and one should load them in memory if possible. We used the notations of the pseudo codes in the appendix for the names of the variables, and the following notations for various quantities: N_v is the number of voxels of one reconstructed material map, N_m the number of materials, N_p the number of pixels of all projections for one bin, N_b the number of energy bins in photon counts measurements and N_e the number of energies used in internal calculations. Despite the fact that we performed 2D simulations, we assume for these calculations that the reconstructed volumes are 3D (which matters for the computation of the spatial gradient). Note that the code we provide keeps many intermediate variables in memory, for convenience and speed considerations, and therefore uses much more memory than the minimum required.

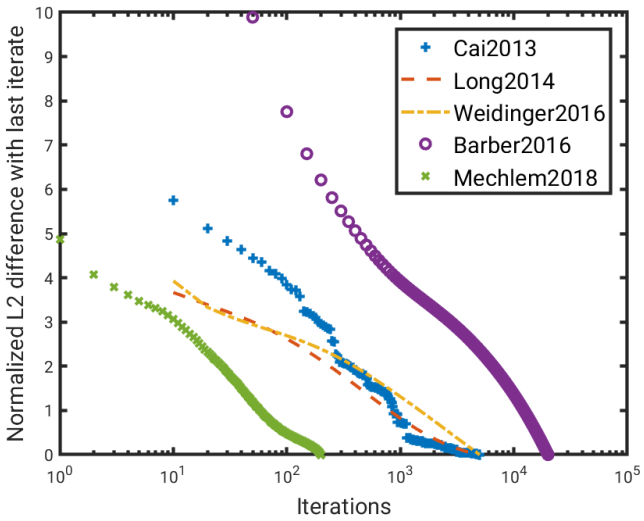

Figure 6. Normalized $\ell 2$ difference with the last iterate.

Table 3. Mean and standard deviation of iodine, gadolinium and water concentrations in the ROI where they are expected eroded by 2 pixels on the final iterate.

	Iodine (mg ml^{-1})		Gadolinium (mg ml^{-1})		Water (g ml^{-1})	
	Mean	std	Mean	std	Mean	std
Cai2013	10.4	2.14	10.1	3.75	0.999	0.0518
Long2014	9.98	1.79	9.93	2.50	1	0.0402
Weidinger2016	9.92	1.86	9.92	2.44	1	0.0425
Barber2016	9.85	3.34	9.65	2.19	1.002	0.064
Mechlem2018	9.97	1.94	9.94	2.70	1	0.0431

Table 4. Number of iterations performed and required to reach target concentrations within 20% and 10% tolerance.

Method	Iterations performed	Time per iter (s)	Within 20% tolerance	Within 10% tolerance
Cai2013	5000	5.2	270	430
Long2014	5000	1.33	140	280
Weidinger2016	5000	0.48	190	390
Barber2016	20 000	1.02	3850	8300
Mechlem2018	200	0.62	5	10

Table 5. Summary of the minimal memory footprints of the five studied methods.

Method	Minimal memory footprint (in number of floating point values)
Cai2013	$4N_vN_b$ with Fessler's preconditioning, $4N_vN_m$ otherwise
Long2014, Weidinger2016	$\left(4+\frac{N_m+1}{2}\right)N_vN_m$
Mechlem2018	$\left(6+\frac{N_m+1}{2}\right)N_vN_m$
Barber2016	$17N_vN_m+3N_pN_b$

3.3.1. Cai2013

The following variables must be stored and updated at each iteration: x_k, d_k, g_k and g_{k-1}, which each contain a set of material volumes. Since all computations can be performed pixel-wise, the expected photon counts $\bar{y}$ do not have to be fully stored in memory, and neither does any of the intermediate variables in the calculation of the gradient and Hessian of the data-attachement term, so we do not count them. Lastly, the spatial gradient of x_k, which is three times as large as x_k, is used in the calculation of g and of d^THd. More precisely, it is the divergence of the first and second derivative of the Huber function of the gradient that is used. These calculations can be computed piecewise, with overlapping pieces since the gradient and divergence require neighboring voxels. In total, at least $4N_vN_m$ floating point values must be kept in memory, and storing $3N_vN_m$ additional values for the gradient of x_k avoids redundant computations, and the problem of having to divide the volumes into overlapping pieces.

Table 6. Memory footprint of Barber2016, for each variable used in appendix C.6 of Barber *et al* (2016). N_v, N_m, N_p and N_b are defined in section 3.3 of the present paper. The words 'pixelwise' and 'voxelwise' mean that the variable can be computed one pixel (or voxel) at a time, thus its computation requires negligible amounts of memory.

Variable	Stored?	Size in memory (in floating point values)
x_0	Yes	N_vN_m
$x^{(k-1)}$ $x^{(k)}$ $x^{(k+1)}$	Only $x^{(k-1)}$	N_vN_m
Σ_{sino}	Yes	N_pN_b
Σ_{grad} w	Only one of both	$3N_vN_m$
T	Yes: used only once, but computed as the sum of two very different calculations, unlikely to occur voxelwise in parallel	N_vN_m
z	No: used only once, pixelwise	0
$u_{\text{sino}}^{(k-1)}$ $u_{\text{sino}}^{(k)}$ $u_{\text{sino}}^{(k+1)}$	Not $u_{\text{sino}}^{(k+1)}$, because it replaces $u_{\text{sino}}^{(k)}$	$2N_pN_b$
$\bar{y}$ b E	No: used only to compute $u_{\text{sino}}^{(k+1)}$, pixelwise	0
y_{grad}^{+}	Yes	$3N_vN_m$
$y_{\text{grad}}^{(k)}$ $y_{\text{grad}}^{(k+1)}$	Only one of both	$3N_vN_m$
g^{+}	Yes: to be projected onto L1 ball	$3N_vN_m$
$\hat{g}^{+}$	No: used only once, voxelwise	0
$x^{(k)}$ $x^{(k+1)}$	Yes	$2N_vN_m$

3.3.2. *Long2014, Weidinger2016 and Mechlem2018*

In all three SQS-based methods we have implemented, the following variables must be stored and updated at each iteration: the current iterate (x_k as in Long and Fessler (2014)), of size N_vN_m, and the gradient and Hessian of the surrogate at x_k, of size N_vN_m and $N_vN_m^2$, respectively. Since the Hessian is known to be symmetric, it can be stored in $N_vN_m(N_m+1)/2$ floating point values. The gradient and the Hessian of the surrogate of the data-attachment term both result from one forward and one back projection. By performing these one angle at a time, they can be computed with negligible amounts of memory. The gradient and Hessian of the surrogate of the regularization term both require a memory size of N_vN_m (the Hessian has $N_vN_m^2$ elements, but is known to be diagonal), and internal computations can be done in place. Since typically they are calculated separately, and then added to those of the regularization terms, their requirements in memory add up. In total, the memory footprint is at least $\left(4+\frac{N_m+1}{2}\right)N_vN_m$ floating point values for Long2014 and Weidinger2016. For Mechlem2018, two additional Nesterov variables must be stored (z_k and u_k), taking the memory footprint to a minimum of $\left(6+\frac{N_m+1}{2}\right)N_vN_m$ floating point values.

3.3.3. *Barber2016*

This method makes use of more intermediate variables than the other four. Table 6 lists the variables used in the pseudo code given in appendix, whether they must be stored or not, why and how many floating point values are therefore required. In total, it uses at least $17N_vN_m + 3N_pN_b$ floating point values.

3.4. Impact of mu-preconditioning

None of the SQS-based methods is impacted by μ-preconditioning: once mapped back to the original materials, the iterates are almost identical to the ones obtained without μ-preconditioning, with difference maps two orders of magnitude less intense than the reconstructed slices. The only impact is with Fessler's method (Fessler 2003): since in our case (three materials, five energy bins) it artificially increases the number of materials to

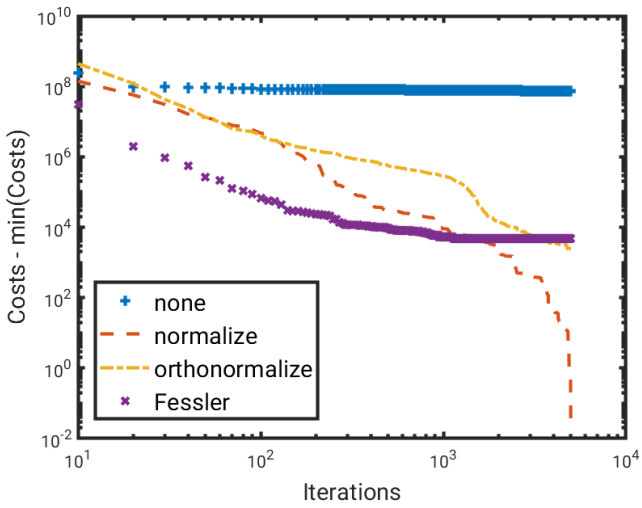

Figure 7. Cost function of Cai2013 over the course of the iterations minus its mininum, depending on the μ-preconditioning method used.

that of the number of bins, the Hessians of the SQS are no longer invertible, so this μ-preconditioning cannot be used at all with SQS-based methods. On Cai2013, however, μ-preconditioning has a strong impact. Figure 7 shows the evolution of the cost function over the course of the iterations minus its minimum, using Cai2013 without μ-preconditioning, with normalization, orthonormalization and Fessler's method. Since all four of these experiments aim to minimize the same cost function, the minimum we subtracted is the minimum over all iterations and all four experiments (reached, as can be seen on the graph, with μ-preconditioning by normalization). It clearly shows that Fessler's method has the fastest initial convergence, which is confirmed by looking at the iterates (not shown here). After more than 1000 iterations, the experiment using normalization reached a lower value of the cost function than the one using Fessler's method.

4. Discussion

We have presented a comparison of five one-step inversion methods for spectral CT, focused on two aspects important for implementation into a real scanner: convergence speed and memory footprint. This choice of criteria is in part due to the initial aim of our study, which was to determine which algorithm should be implemented to process large real datasets in reasonable time. But more importantly, it was dictated by the behavior of the algorithms themselves: while it seemed impossible to make a convincing case on which method performed best in terms of image quality, the differences in convergence speed were striking. We provide the complete Matlab code we have used for this study (see section 2.7), so that readers interested in conducting comparisons on other aspects have the means to do so.

All algorithms were adapted to find the three material maps that minimized a cost function composed of a data attachment term for five energy bins and a spatial regularization minimizing the $\ell 1$-norm (or a twice-differentiable approximation of it) of the spatial gradient of each reconstructed volume. The original algorithms were generally designed for other problems and our conclusions hold only for our specific problem. For examples, the following modifications of the problem would require another comparison: reconstructing two maps from dual-energy measurements as in Cai *et al* (2013) or constraining the solutions with positivity as in Long and Fessler (2014).

There are other one-step inversion methods in the literature than the ones we selected, e.g. Sukovic and Clinthorne (2000), Chen *et al* (2017), and it is likely that more will be proposed in the future. We hope that others will be interested in implementing their method in the Matlab framework we provide, to extend the comparison.

A limitation of the present study is the small size of the data used in the experiments, which was dictated by our implementation using a sparse system matrix stored in computer memory. It is therefore possible that on larger datasets, e.g. on a full real helical acquisition, the relative convergence speeds of the studied methods would be different. However, the other observations made in the paper would remain valid.

Another limitation is that since one-step inversion is a non-convex problem, the results may strongly depend on the initialization. Mechlem *et al* recommend to initialize the algorithms with reconstruction results from a two-step method. We have not tried that approach, and have instead initialized all reconstructions with zero-filled slices. Different initialization strategies may lead to different conclusions on relative convergence speeds.

In several ways, the simulation we have studied is simplistic: in a real acquisition, the spectrum is usually filtered to remove most low-energy photons, made pixel-dependent by a bow tie filter, and the pixels and voxels

are much finer than the ones we used (around 0.2 mm instead of 1 mm). Since the same sparse system matrix was used both for simulation and reconstruction, this work adopts an inverse-crime setup (Wirgin 2004), with statistical noise. It does not take into account any advanced physical effect like scatter, pile-up or charge sharing. However, these observations hold for all methods, so the comparison remains valid.

Sidky *et al* have recently proposed a more efficient μ-preconditioning method (Sidky *et al* 2018), better-suited to three-materials reconstructions than the one described in Barber2016. In the experiment they present, this new μ-preconditioning method speeds up convergence by a factor 10. The slow convergence of Barber2016 in our experiments must therefore be mitigated by the fact that we did not implement that method. However, a factor 10 would only make its convergence speed similar to Cai2013, Long2014 and Weidinger2016, and our other observations on Barber2016 (mostly its high memory footprint) remain valid.

Our implementation of Mechlem2018 differs from that of Weidinger2016 only by the use of Nesterov's momentum technique and a small number of ordered subsets, and yet greatly outperforms Weidinger2016. Therefore, we confirm the observation made in the discussion of Mechlem2018 and in several other papers (Kim *et al* 2015, Hansen and Sørensen 2018): Nesterov's momentum technique combined with ordered subsets highly accelerates the convergence speed per iteration. This acceleration might cause numerical instabilities which would make Mechlem2018 more difficult to use with real data.

5. Conclusion

We have adapted five one-step methods for spectral CT (Cai2013 (Cai *et al* 2013), Long2014 (Long and Fessler 2014), Weidinger2016 (Weidinger *et al* 2016), Barber2016 (Barber *et al* 2016), Mechlem2018 (Mechlem *et al* 2018)) to reconstruct three material maps from five energy bins with spatial regularization in each material map. We have compared them in terms of practical aspects: convergence speed to reach a fixed image quality threshold, and memory footprint. They are obviously not the only criteria for the choice of a method, and many other parameters should probably be considered such as image quality at convergence or numerical stability. Yet, Mechlem2018 solved this one-step problem in 1–3 orders of magnitude faster than its competitors, and such a vast difference should not be overlooked. Alternative optimizations or modifications of the problem would require reinvestigating this conclusion, and we provide the source code of the software used in this study to facilitate such investigations.

Acknowledgments

This work was performed within the framework of the EU's H2020 research and innovation program under the grant agreement No. 668142, the SIRIC LYricAN Grant INCa-INSERM-DGOS-12563, and the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the program 'Investissements d'Avenir' (ANR-11-IDEX-0007) operated by the ANR. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

Appendix

Since we performed some modifications on almost all methods, we provide in this section the complete pseudo code of our implementations. This pseudo code makes extensive use of matrix notation, so we define some matrices and their size, in format [lines, columns] (see section 3.3 for the definitions of the N values): M is the (preconditioned) material mass-attenuation matrix, and P the μ-preconditioning matrix, defined in section 2.3. P and M are $[N_m, N_b]$ and $[N_e, N_b]$ in the case of Fessler's μ-preconditioning, and $[N_m, N_m]$ and $[N_e, N_m]$ otherwise. $A[N_p, N_v]$ is the forward projection matrix for one material, and $S[N_b, N_e]$ the effective spectrum, as used in equation (2). In some algorithms, those matrices are redefined using Kronecker products (noted $\otimes$). They still perform the same operation, only extended to several materials, bins, pixels or voxels. When the original matrix is needed, it is indexed by the pixel or voxel. Some calculations cannot be easily translated into Kronecker products, therefore we sometimes use the symbol $*$, which $*$ stands for element-wise product with implicit extension, e.g.

$$\begin{pmatrix}1\\2\end{pmatrix} * \begin{pmatrix}0 & 1 & 2\\3 & 4 & 5\end{pmatrix} = \begin{pmatrix}0 & 1 & 2\\6 & 8 & 10\end{pmatrix}. \tag{A.1}$$

Of course, it would have been highly inefficient to actually implement the presented methods using Kronecker products, so the pseudo code does not reflect the way things are actually implemented in the Matlab code we provide. In vectors storing volumes, the order of the dimensions is 'x, y, z, material'. In vectors storing photon counts, it is 'x, y, projection, energy bin'. If μ-preconditioning is used, the initial guess, which we denote X, must be multiplied by P^{-1} to obtain x_0. In all methods except Barber2016, regularization involves the use of a twice-differentiable approximation of the $\ell 1$ norm. Cai2013 and Mechlem2018 use the Huber function, Weidinger2016 uses the Green

prior, and Long2014 a custom hyperbola. We denote all of these functions with a unique symbol ϕ which, as for Cai2013 (see section A.1), already contains the regularization weights (one per material) and, if applicable, the thresholds (one per material). For SQS-based methods, the regularization is computed by examining, for each voxel, the surrounding voxels that belong to a neighborhood $\mathcal{N}$. For Long2014 and Mechlem2018, which use subsets, we define $A_s = \mathrm{sub}(A, s)$ which selects rows in the projection matrix A corresponding to the projections in subset s. Which projection goes into which subset is determined by the following procedure: we compute a random permutation of $[1, 2, \ldots, N_{\text{projections}}]$, and divide the result into N_{subsets} equal parts.

A.1. Pseudo-code of Cai2013

Contrary to the other four methods, Cai2013 defines S as the normalized spectrum instead of the absolute one, and y as transmission ratios instead of photon counts. Let us define $\vec{w} = (w_1, \ldots, w_{N_m})$ and $\vec{\delta}$ the vectors of regularization weights, and of Huber function thresholds, each holding one value per material. In Cai2013, the regularization term is the sum over all voxels of the Huber function of the spatial gradient of the material volumes, i.e.

$$\begin{aligned} \Phi :& \mathbb{R}^{3N_vN_m} \times \mathbb{R}^{N_m} \times \mathbb{R}^{N_m} \to \mathbb{R} \\ & (t, \vec{w}, \vec{\delta}) \to \sum_{m=1}^{N_m} w_m \sum_{i=1}^{3N_v} \phi(t_i, \delta_m) \end{aligned} \tag{A.2}$$

where in practice, t is always the spatial gradient of the material volumes. The gradient and Hessian of Φ are denoted Φ' and Φ'', respectively, and are of size $[3N_vN_m, 1]$ and $[3N_vN_m, 3N_vN_m]$.

Algorithm 1. Cai2013 pseudo code.

$P = P \otimes I_{N_v}$
$A = I_{N_m} \otimes A$
$M = M \otimes I_{N_p}$
$S = S \otimes I_{N_p}$
$x_0 \leftarrow P^{-1}X$
$g_0 \leftarrow \mathbf{0}$
$d_0 \leftarrow \mathbf{0}$
$Q \leftarrow \exp(-MAx_0)$
$\bar{y} \leftarrow SQ$
for $k \leftarrow 1$ **to** $N_{iterations}$ **do**
  $z \leftarrow \left(\frac{y^2-\bar{y}^2}{k_d\bar{y}^2} - \frac{1}{\bar{y}}\right)$
  $x_{real} \leftarrow Px_{k-1}$
  $g_k \leftarrow A^TM^TQS^Tz + P^T\nabla^T\Phi'\left(\nabla x_{real}, \vec{W}, \vec{\delta}\right)$
  $\beta \leftarrow \frac{\langle g_k - g_{k-1}, g_{k-1}\rangle}{\|g_{k-1}\|_2^2}$
  if $\beta < 0$ **or** $k = 1$ **then** $\beta \leftarrow 0$
  $d_k \leftarrow -g_k + \beta d_k$
  $v \leftarrow \mathrm{diag}[\frac{2y^2}{k_d\bar{y}^3} - \frac{1}{\bar{y}^2}]$
  $d_{real} \leftarrow Pd_k$
  $d_k^THd_k \leftarrow d_k^TA^TM^TQ(S^TvSQ - \mathrm{diag}[S^Tz])MAd_k$
  $\quad + d_{real}^TP^T\nabla^T\Phi''\left(\nabla x_{real}, \vec{W}, \vec{\delta}\right)\nabla Pd_{real}$
  $\alpha \leftarrow -\frac{\langle g_k, d_k\rangle}{d_k^THd_k}$
  $x_k \leftarrow x_{k-1} + \alpha d_k$
  while $cost(x_k) > cost(x_{k-1})$ **do**
    $\alpha \leftarrow \alpha/2$
    $x_k \leftarrow x_{k-1} + \alpha d_k$;
  end
  $Q \leftarrow \exp(-MAx_k)$
  $\bar{y} \leftarrow SQ$
end
return $\boldsymbol{Px_k}$

A.2. Pseudo-code of Long2014

It uses the optimal_curvature function, which is defined on $\mathbb{R}^+$ as follows:

$$\text{optimal\_curvature}(x) = 2\frac{1 - e^{-x} - xe^{-x}}{x^2} \text{ if } x > 0, 1 \text{ if } x = 0. \tag{A.3}$$

When the input to optimal_curvature is a vector, it is applied element-wise.

Algorithm 2. Long2014 pseudo code.

$\pi = A * (A\mathbb{1}_{N_v,1})$
$A = I_{N_m} \otimes A$
$L = (M \otimes \mathbb{1}_{1,N_m}) * (\mathbb{1}_{1,N_m} \otimes M)$
$M = M \otimes I_{N_p}$
$S = S \otimes I_{N_p}$
$x_0 \leftarrow X$
for $k \leftarrow 1$ **to** $N_{iterations}$ **do**
  for $s \leftarrow 1$ **to** $N_{subsets}$ **do**
    $A_s = \text{sub}(A, s)$
    $\pi_s = \text{sub}(\pi, s)$
    $Q \leftarrow \exp(-MA_s x_{k-1})$
    $\bar{y} \leftarrow SQ$
    $g_{data} \leftarrow A_s^T \left((\frac{y}{\bar{y}} - 1) * SQM \right)$
    $c = \text{optimal\_curvature}(MRx_{k-1})$
    for $p \leftarrow 1$ **to** *Npixels in subset* s **do**
      $C_p = \sum_{b=1}^{N_b} \left(S_p(c_p * L) \right)$
    end
    $H_{data} \leftarrow \text{reshape}(\pi_s^T C, [N_v, N_m, N_m])$
    for $v \leftarrow 1$ **to** N_v **do**
      $g_{regul,v} \leftarrow \frac{2}{N_{\text{subsets}}} \sum_{\tilde{v} \in \mathcal{N}} \phi'(x_{k-1,v} - x_{k-1,\tilde{v}})$
      $H_{regul,v} \leftarrow 4 \text{ diag} \left[\sum_{\tilde{v} \in \mathcal{N}} \phi''(x_{k-1,v} - x_{k-1,\tilde{v}}) \right]$
      $g_v \leftarrow g_{data,v} + g_{regul}$
      $H_v \leftarrow H_{data,v} + H_{regul}$
      $x_{k,v} \leftarrow x_{k-1,v} - H_v^{-1} g_v$
    end
  end
end
return x_k

A.3. Pseudo-code of Weidinger2016

Algorithm 3. Weidinger2016 pseudo code.

$A = I_{N_m} \otimes A$
$L = (M \otimes \mathbb{1}_{1,N_m}) * (\mathbb{1}_{1,N_m} \otimes M)$
$M = M \otimes I_{N_p}$
$S = S \otimes I_{N_p}$
$x_0 \leftarrow X$
for $k \leftarrow 1$ **to** $N_{iterations}$ **do**
  $Q \leftarrow \exp(-MAx_{k-1})$
  $\bar{y} \leftarrow SQ$
  $g_{data} \leftarrow A^T \left((\frac{y}{\bar{y}} - 1) * (SQM) \right)$
  $H_{data} \leftarrow A^T ((A\mathbb{1}_{N_v,1}) * (SQL))$
  for $v \leftarrow 1$ **to** N_v **do**
    $g_{regul,v} \leftarrow 2 \sum_{\tilde{v} \in \mathcal{N}} \phi'(x_{k-1,v} - x_{k-1,\tilde{v}})$
    $H_{regul,v} \leftarrow 4 \text{ diag} \left[\sum_{\tilde{v} \in \mathcal{N}} \phi''(x_{k-1,v} - x_{k-1,\tilde{v}}) \right]$
    $g_v \leftarrow g_{data,v} + g_{regul}$
    $H_v \leftarrow H_{data,v} + H_{regul}$
    $x_{k,v} \leftarrow x_{k-1,v} - H_v^{-1} g_v$
  end
end

19 return x_k

A.4. Pseudo-code of Mechlem2018

It differs of that of Weidinger2016 only by the use of subsets and Nesterov's momentum technique.

Algorithm 4. Mechlem2018 pseudo code.

$A = I_{N_m} \otimes A$
$L = (M \otimes \mathbb{1}_{1,N_m}) * (\mathbb{1}_{1,N_m} \otimes M)$
$M = M \otimes I_{N_p}$
$S = S \otimes I_{N_p}$
$z_0 \leftarrow X$
for $k \leftarrow 1$ **to** $N_{iterations}$ **do**
  for $s \leftarrow 1$ **to** $N_{subsets}$ **do**
    $A_s = \mathrm{sub}(A, s)$
    $Q \leftarrow \exp(-MA_s z_{k-1})$
    $\bar{y} \leftarrow SQ$
    $g_{data} \leftarrow A_s^T \left(\left(\frac{y}{\bar{y}} - 1\right) * (SQM) \right)$
    $H_{data} \leftarrow A_s^T \left((A_s \mathbb{1}_{N_v,1}) * (SQL) \right)$
    for $v \leftarrow 1$ **to** N_v **do**
      $g_{regul,v} \leftarrow \frac{2}{N_{\text{subsets}}} \sum_{\tilde{v} \in \mathcal{N}} \phi' (z_{k-1,v} - z_{k-1,\tilde{v}})$
      $H_{regul,v} \leftarrow 4\,\mathrm{diag} \left[\sum_{\tilde{v} \in \mathcal{N}} \phi'' (z_{k-1,v} - z_{k-1,\tilde{v}}) \right]$
      $g_v \leftarrow g_{data,v} + g_{regul}$
      $H_v \leftarrow H_{data,v} + H_{regul}$
      $d_{k,v} \leftarrow z_{k-1,v} - H_v^{-1} g_v$
    end
    $t_k \leftarrow \frac{1}{2} \left(1 + \sqrt{1 + 4t_{k-1}^2} \right)$
    $\alpha_k \leftarrow z_k - d_k$
    $u_k \leftarrow u_k - t_{k-1} d_k$
    $z_k \leftarrow \alpha_k + \frac{t_k}{\sum_{l=0}^{k} t_l} (u_k - \alpha_k)$;
  end
end
return z_k

A.5. Pseudo-code of Barber2016

All other four methods use a weighted regularization term, with weights controlling the strength of the regularization: the larger the weight, the stronger the regularization. Barber2016 works differently: it enforces a constraint, stating that the TV on each material must not exceed a threshold γ_m (one per material). Therefore, the larger the γ_ms, the looser the TV constraint.

Algorithm 5. Barber2016 pseudo code.

$Z_{pe,mv} \leftarrow M_{me} A_{pv}$
$A = I_{N_m} \otimes A$
$M = M \otimes I_{N_p}$
$S = S \otimes I_{N_p}$
$P = P \otimes I_{N_v}$
$x^{(0)} \leftarrow P^{-1} X$
for $k \leftarrow 1$ **to** $N_{iterations}$ **do**
  $x_0 \leftarrow \bar{x}^{(k)}$
  $B_{bp,p'e}(x) \leftarrow \frac{s_{bpe} \exp\left(-(Zx)_{pe}\right)}{\sum_{e'} s_{bpe'} \exp\left(-(Zx)_{pe'}\right)} I_{pp'}$
  with $I_{pp'} = 1$ if $p = p'$ and 0 otherwise
  $K \leftarrow B(x_0) Z$
  $\bar{y} \leftarrow S \exp(-MAx_0)$
  $E \leftarrow max(\bar{y} - y, 0)$
  $b \leftarrow (\bar{y} - E) K x_0 + \bar{y} - y$
  $\Sigma_{\text{sino}} \leftarrow \frac{1}{\lambda} \mathrm{diag} [K \mathbb{1}]^{-1}$

16 $\Sigma_{\text{grad}} \leftarrow \frac{1}{\lambda}\text{diag}\left[|\nabla|P\mathbb{1}\right]^{-1}$

17 $T \leftarrow \lambda\,\text{diag}\left[K^T\mathbb{1} \quad (|\nabla|P\mathbb{1})^T\right]^{-1}$

18 $\forall m, w_m \leftarrow \Sigma_{\text{grad},m}^{1/2}$

19 $z \leftarrow \Sigma_{\text{sino}}^{-1}\left(u_{\text{sino}}^{(k-1)} - u_{\text{sino}}^{(k)} + \Sigma_{\text{sino}}K\bar{x}^{(k-1)}\right)$

20 $u_{\text{sino}}^{(k+1)} \leftarrow (\bar{y} + \Sigma_{\text{sino}})^{-1}\left[\bar{y}(u_{\text{sino}}^{(k)} + \Sigma_{\text{sino}}K\bar{x}^{(k)}) - \Sigma_{\text{sino}}(b + Ez)\right]$

21 $\forall m, u_{\text{grad},m}^{+} \leftarrow u_{\text{grad},m}^{(n)} + \Sigma_{\text{grad},m}\nabla(P\bar{x}^{(k)})_m$

22 $\forall m, g_m^{+} \leftarrow |u_{\text{grad},m}^{+}/w_m|$

23 $\forall m, \hat{g}_m^{+} \leftarrow \text{sign}(u_{\text{grad},m}^{+}/w_m)$

24 $\forall m, u_{\text{grad},m}^{(k+1)} \leftarrow u_{\text{grad},m}^{+} - w_m\hat{g}_m^{+}\text{Proj}(g_m^{+}; g, \|g/w_m\| \leqslant \gamma_m)$

25 $x^{(k+1)} \leftarrow \bar{x}^{(k)} - T\begin{pmatrix} K \\ \nabla P \end{pmatrix}^T \begin{pmatrix} u_{\text{sino}}^{(k+1)} \\ u_{\text{grad}}^{(k+1)} \end{pmatrix}$

26 $\bar{x}^{(k+1)} \leftarrow x^{(k+1)} + \theta(x^{(k+1)} - x^{(k)})$

27 end

28 return $\boldsymbol{Px}^{(k)}$

ORCID iDs

Cyril Mory https://orcid.org/0000-0002-1813-6577

References

Alvarez R E and Macovski A 1976 Energy-selective reconstructions in x-ray computerized tomography *Phys. Med. Biol.* **21** 733–44

Barber R F and Sidky E Y 2016 MOCCA: mirrored convex/concave optimization for nonconvex composite functions *J. Mach. Learn. Res.* **17** 5006–56

Barber R F, Sidky E Y, Schmidt T G and Pan X 2016 An algorithm for constrained one-step inversion of spectral CT data *Phys. Med. Biol.* **61** 3784

Brendel B, Bergner F, Brown K and Koehler T 2016 Penalized likelihood decomposition for dual layer spectral CT *Proc. of the 4th Int. Meeting in X-Ray Computed Tomography (Bamberg, Germany) pp* 41–4

Cai C, Rodet T, Legoupil S and Mohammad-Djafari A 2013 A full-spectral Bayesian reconstruction approach based on the material decomposition model applied in dual-energy computed tomography *Med. Phys.* **40** 111916–31

Chambolle A and Pock T 2011 A first-order primal-dual algorithm for convex problems with applications to imaging *J. Math. Imaging Vis.* **40** 120–45

Chen B, Zhang Z, Sidky E Y, Xia D and Pan X 2017 Image reconstruction and scan configurations enabled by optimization-based algorithms in multispectral CT *Phys. Med. Biol.* **62** 8763

Clackdoyle R, Desbat L, Lesaint J and Rit S 2016 Data consistency conditions for cone-beam projections on a circular trajectory *IEEE Signal Process. Lett.* **23** 1746–50

Ducros N, Abascal J F P J, Sixou B, Rit S and Peyrin F 2017 Regularization of nonlinear decomposition of spectral x-ray projection images *Med. Phys.* **44** e174–e187

Fessler J A 2003 Method for statistically reconstructing images from a plurality of transmission measurements having energy diversity and image reconstructor apparatus utilizing the method *US Patent Specification* US6754298B2

Flohr T G *et al* 2006 First performance evaluation of a dual-source CT (DSCT) system *Eur. Radiol.* **16** 256–68

Green P J 1990 Bayesian reconstructions from emission tomography data using a modified EM algorithm *IEEE Trans. Med. Imaging* **9** 84–93

Hansen D C and Sørensen T S 2018 Fast 4D cone-beam CT from 60 s acquisitions *Phys. Imaging Radiat. Oncol.* **5** 69–75

Hansen P C and Saxild-Hansen M 2012 AIR tools—a MATLAB package of algebraic iterative reconstruction methods *J. Comput. Appl. Math.* **236** 2167–78

Hudson H M and Larkin R S 1994 Accelerated image reconstruction using ordered subsets of projection data *IEEE IEEE Trans. Med. Imaging* **13** 601–9

Keerthi S S and Gilbert E G 2002 Convergence of a generalized SMO algorithm for SVM classifier design *Mach. Learn.* **46** 351–60

Kim D, Ramani S and Fessler J A 2015 Combining ordered subsets and momentum for accelerated x-ray CT image reconstruction *IEEE Trans. Med. Imaging* **34** 167–78

Lesaint J, Rit S, Clackdoyle R and Desbat L 2017 Calibration for circular cone-beam CT based on consistency conditions *IEEE Trans. Radiat. Plasma Med. Sci.* **1** 517–26

Long Y and Fessler J A 2014 Multi-material decomposition using statistical image reconstruction for spectral CT *IEEE Trans. Med. Imaging* **33** 1614–26

Maass C, Baer M and Kachelriess M 2009 Image-based dual energy CT using optimized precorrection functions: a practical new approach of material decomposition in image domain *Med. Phys.* **36** 3818–29

Maass N, Sawall S, Knaup M and Kachelriess M 2011 Empirical multiple energy calibration (EMEC) for material-selective CT *Proc. IEEE Nuclear Science Symp. and Medical Imaging Conf. (NSS/MIC)* pp 4222–9

Mechlem K, Ehn S, Sellerer T, Braig E, Münzel D, Pfeiffer F and Noël P B 2018 Joint statistical iterative material image reconstruction for spectral computed tomography using a semi-empirical forward model *IEEE Trans. Med. Imaging* **37** 68–80

Mendonca P R S, Lamb P and Sahani D V 2014 A flexible method for multi-material decomposition of dual-energy CT images *IEEE Trans. Med. Imaging* **33** 99–116

Mory C, Brendel B, Erhard K and Rit S 2018 Generalized least squares for spectral and dual energy CT: a simulation study *Proc. of the 5th Int. Conf. on Image Formation in X-Ray Computed Tomography (Salt Lake City, USA, 20–23 May 2018)* pp 98–101

Niu T, Dong X, Petrongolo M and Zhu L 2014 Iterative image-domain decomposition for dual-energy CT *Med. Phys.* **41** 041901

Pan D *et al* 2010 Computed tomography in color: NanoK-enhanced spectral CT molecular imaging *Angew. Chem.* **122** 9829–33

Pan X, Sidky E Y and Vannier M 2009 Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction? *Inverse Problems* **25** 123009

Poludniowski G, Landry G, DeBlois F, Evans P M and Verhaegen F 2009 SpekCalc: a program to calculate photon spectra from tungsten anode x-ray tubes *Phys. Med. Biol.* **54** N433–8

Zhang R, Thibault J B, Bouman C A, Sauer K D and Hsieh J 2014 Model-based iterative reconstruction for dual-energy x-ray CT using a joint quadratic likelihood model *IEEE Trans. Med. Imaging* **33** 117–34

Sawatzky A, Xu Q, Schirra C O and Anastasio M A 2014 Proximal ADMM for multi-channel image reconstruction in spectral x-ray CT *IEEE Trans. Med. Imaging* **33** 1657–68

Schlomka J P *et al* 2008 Experimental feasibility of multi-energy photon-counting K-edge imaging in pre-clinical computed tomography *Phys. Med. Biol.* **53** 4031

Si-Mohamed S *et al* 2017 Evaluation of spectral photon counting computed tomography K-edge imaging for determination of gold nanoparticle biodistribution *in vivo Nanoscale* **9** 18246–57

Sidky E Y, Barber R F, Gilat-Schmidt T and Pan X 2018 Three material decomposition for spectral computed tomography enabled by block-diagonal step-preconditioning *Proc. of the 5th Int. Conf. on Image Formation in X-Ray Computed Tomography (Salt Lake City, USA, 20–23 May 2018)* pp 268–71

So A and Lee T Y 2015 Rapid kV switching dual-energy CT imaging *Dual-Energy CT in Cardiovascular Imaging* (Cham: Springer) pp 45–60

Sukovic P and Clinthorne N H 2000 Penalized weighted least-squares image reconstruction for dual energy x-ray transmission tomography *IEEE Trans. Med. Imaging* **19** 1075–81

Taguchi K, Itoh T, Fuld M, Fournie E, Lee O and Noguchi K 2018 *'X-map 2.0' and 'iBHC2' for Consistent Edema Signal Enhancement for Acute Ischemic Stroke using Non-Contrast-Enhanced Dual-Energy CT Proc. of the 5th Int. Conf. on Image Formation in X-Ray Computed Tomography (Salt Lake City, USA, 20–23 May 2018)* pp 86–9

Weidinger T, Buzug T M, Flohr T, Kappler S and Stierstorfer K 2016 Polychromatic iterative statistical material image reconstruction for photon-counting computed tomography *Int. J. Biomed. Imaging* **2016** 1–15

Wirgin A 2004 The inverse crime (arXiv:math-ph/0401050)