

Journal of Medical Imaging

MedicalImaging.SPIEDigitalLibrary.org

Primal-dual approach to optical tomography with discretized path integral with efficient formulations

Bingzhi Yuan
Toru Tamaki
Bisser Raytchev
Kazufumi Kaneda

SPIE.

Bingzhi Yuan, Toru Tamaki, Bisser Raytchev, Kazufumi Kaneda, "Primal-dual approach to optical tomography with discretized path integral with efficient formulations," *J. Med. Imag.* **4**(3), 033501 (2017), doi: 10.1117/1.JMI.4.3.033501.

Primal-dual approach to optical tomography with discretized path integral with efficient formulations

Bingzhi Yuan, Toru Tamaki,* Bisser Raytchev, and Kazufumi Kaneda

Hiroshima University, Department of Information Engineering, Graduate School of Engineering, Higashi-Hiroshima, Japan

Abstract. We propose an efficient optical tomography with discretized path integral. We first introduce the primal-dual approach to solve the inverse problem formulated as a constraint optimization problem. Next, we develop efficient formulations for computing Jacobian and Hessian of the cost function of the constraint nonlinear optimization problem. Numerical experiments show that the proposed formulation is faster (23.14 ± 1.32 s) than the previous work with the log-barrier interior point method (91.17 ± 1.48 s) for the Shepp–Logan phantom with a grid size of 24×24 , while keeping the quality of the estimation results (root-mean-square error increasing by up to 12%). © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JMI.4.3.033501](https://doi.org/10.1117/1.JMI.4.3.033501)]

Keywords: optical tomography; multiple scattering; path integral; primal-dual interior point method.

Paper 17106R received Apr. 18, 2017; accepted for publication Jun. 28, 2017; published online Jul. 19, 2017.

1 Introduction

Optical tomography^{1–8} is a technique for reconstructing inside an object by illuminating it with a light probe and observing the light penetrating through the object. In contrast to x-ray computed tomography, which uses x-rays instead of light, safer tomographic methods are demanded and scattering optical tomography methods are recently attracting computer vision researchers' attention.^{9–15}

We investigate a recently proposed optical tomography with discretized path integral.^{12,13,15} Their method takes advantages of the path integral formulation and formulates the inverse problem of optical tomography as a constraint nonlinear least squared optimization problem. This method benefits from various optimization techniques, which is not the case for voting^{11,14} or genetic algorithms.¹⁰ They solved the constraint optimization problem by using the log-barrier (LB) interior point method¹⁶ with inner loops of Newton method¹² and quasi-Newton method.¹⁵ They have shown that optical tomography with discretized path integral produces better estimation results compared to a standard diffuse optical tomography (DOT),¹⁵ however, its high computation cost is a problem for further development.

In this paper, we propose two contributions to tackle the problem. First, we introduce the primal-dual (PD) approach to solve the constraint optimization because it is known that the PD interior point method is more efficient than the LB method.¹⁶ Second, we propose new formulations of equations. The main bottle-neck of the previous approaches^{12,13,15} is Jacobian and Hessian which are computationally demanding as the number of paths increases. Our new formulations are equivalent to the previous ones, but much more efficient. Numerical simulations show that the proposed approach accelerates the estimation by a factor of 100. (Conference versions of this paper were presented.^{17,18} This paper extends those versions with the extended description of the PD approach and the

efficient formulations, and additional experiments with optimized codes with comparisons.)

There exists a number of acceleration methods for tomographic reconstruction; such as dimension reduction,¹⁹ power acceleration,²⁰ and, most importantly, graphic processing units (GPUs).²¹ GPUs have been used for accelerating a forward problem²² and forward and backward projections.^{23–25} In addition, thanks to the recent progress of general-purpose computing on GPUs, GPUs have become popular for speeding up iterative computations for solving compressive sensing formulations²⁶ and large linear systems.^{27,28} We have not implemented the proposed approach with any GPUs in this paper, however, the use of GPUs would be beneficial for the proposed formulations because computing a Jacobian matrix can be further accelerated.²⁹

In the following, we briefly summarize the constraint problem to be solved in Sec. 2. Then in Sec. 3, we develop the PD method to solve the problem by taking into account the structure of the problem. In Sec. 4, we derive new efficient formulations to compute Jacobian and Hessian, with comparisons to and discussions of the previous formulations. We show simulation results in Sec. 5 to show the improvement of the proposed method in terms of computation cost.

2 Preliminary

In this section, we briefly review the formulation of optical tomography with a discretized path integral. Details can be found in Ref. 15.

We are interested in estimating the extinction coefficients σ , of each voxel in a two-dimensional (2-D) medium as shown in Fig. 1. Extinction coefficients represent how much light attenuates at each point. We follow the 2-D layer model,¹⁵ that is, we assume the following layer scattering with the following properties. Suppose that the 2-D medium has a layered structure and is discretized into voxels of an $M \times N$ grid; it has M layers of N voxels. Therefore, the problem is to estimate extinction coefficients of each voxel in the grid.

*Address all correspondence to: Toru Tamaki, E-mail: tamaki@hiroshima-u.ac.jp

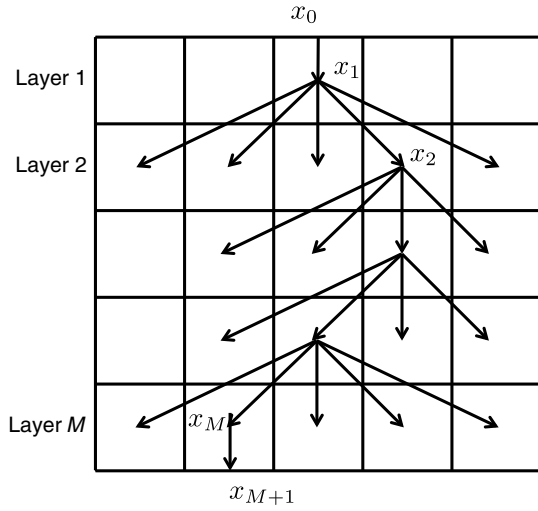


Fig. 1 2-D layered model of scattering.¹⁵ This example of a 5×5 grid shows a path consisting of vertices x_1, \dots, x_M located at the centers of voxels in a grid with M parallel layers. x_0 is a light source located on the top surface, and x_{M+1} is a detector at the bottom.

With this layered medium, we use an observation model of the light transport between a light source and a detector: emitting light to each of the voxels at the top layer and capturing light from each voxel from the bottom layer (see Fig. 2). More specifically, the light source point x_0 is located on the boundary of the top surface of the voxels in the top layer. The detector point x_{M+1} is located on the boundary of the bottom surface of the voxels in the bottom layer. Then, forward scattering happens layer by layer; light is scattered at the center of a voxel in a layer and goes to the center of a voxel in the next (below) layer. By connecting the centers x_1, \dots, x_M of voxels of each layer, we have a path $x_0, x_1, \dots, x_M, x_{M+1}$ of light scattering connecting the light source and the detector. Let i and j be voxel indices of the light source and detector locations, respectively. By restricting the light paths only to those connecting i and j , the observed light I_{ij} by the detector is the sum of contributions of all paths connecting i and j . This is written as follows:

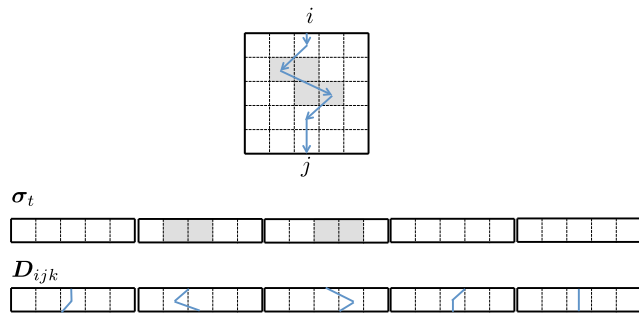


Fig. 2 An example of a path in a 5×5 grid. The path \mathbf{D}_{ijk} is one of paths that connect locations i and j . The grid is serialized to vector $\boldsymbol{\sigma}_t$, and also to vector \mathbf{D}_{ijk} separately, but in a consistent order. These vectors are used to represent the exponential attenuation of light along the path by inner product followed by the exponential function as $\exp(-\boldsymbol{\sigma}_t^T \mathbf{D}_{ijk})$ in Eq. (3).

$$I_{ij} = I_0 \sum_{k=1}^{N_{ij}} H_{ijk} e^{-\boldsymbol{\sigma}_t^T \mathbf{D}_{ijk}}, \quad (1)$$

where $\boldsymbol{\sigma}_t \in \mathbb{R}^{NM}$ is a vector to be estimated, and each element is the extinction coefficient of a voxel, as shown Fig. 2. Vector $\mathbf{D}_k \in \mathbb{R}^{NM}$ represents a complete path k connecting i and j , and each element is the length of the part of the path segment passing through the corresponding voxel. Factor H_{ijk} encodes scattering coefficients and the phase function, I_0 is the intensity of the light source, and N_{ij} is the number of paths connecting i and j . Parameters H_{ijk} , I_0 , N_{ij} , \mathbf{D}_k are given and fixed. The problem is to estimate $\boldsymbol{\sigma}_t$ based on observations I_{ij} .

By changing positions of the light source i and the detector j , we obtain a set of observations $\{I_{ij}\}$, resulting in the following nonlinear least squares problem under box constraints for extinction coefficients to be positive

$$\min_{\boldsymbol{\sigma}_t} f \quad \text{s.t. } \sigma_{tl} \leq \boldsymbol{\sigma}_t \leq \sigma_{tu}, \quad (2)$$

where f is the cost function

$$f = \sum_{i=1}^N \sum_{j=1}^N \left| I_{ij} - I_0 \sum_{k=1}^{N_{ij}} H_{ijk} e^{-\boldsymbol{\sigma}_t^T \mathbf{D}_{ijk}} \right|^2, \quad (3)$$

and σ_{tl} and σ_{tu} are lower and upper bounds, respectively. The box constraints are due to the nature of the extinction coefficient being positive (i.e., $\sigma_{tl} > 0$) and the numerical stability of excluding unrealistic large values.

3 Primal-Dual Interior Point Method of the Inverse Problem

Here, we develop a PD interior point method to solve the inverse problem [Eq. (2)]. It is an inequality constraint optimization with box constraints, which is straightforward in applying a standard PD method.¹⁶ However, we can use the structure of the box constraints, hence, we derive an efficient algorithm by using the problem structure.

3.1 Primal-Dual Method

We first rewrite the inequality constraint problem to an equivalent problem with equality constraints with slack variables $\mathbf{s} = (s_1, \dots, s_{2NM})^T$ as follows:

$$\min_{\boldsymbol{\sigma}_t} f \quad \text{s.t. } \mathbf{c} - \mathbf{s} = \mathbf{0}, \quad 0 \leq \mathbf{s}, \quad (4)$$

where \mathbf{c} is a vector of the box constraints

$$\mathbf{c} = \begin{pmatrix} c_1(\boldsymbol{\sigma}_t) \\ \vdots \\ c_{2NM}(\boldsymbol{\sigma}_t) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\sigma}_t - \sigma_{tl} \mathbf{1} \\ \sigma_{tu} \mathbf{1} - \boldsymbol{\sigma}_t \end{pmatrix}. \quad (5)$$

Here, c_i is the i 'th constraint, and $\mathbf{1}$ is a vector of ones.

The Lagrangian L of the above problem is

$$L(\boldsymbol{\sigma}_t, \mathbf{s}, \mathbf{z}) = f - \sum_{i=1}^{2NM} z_i (c_i - s_i) = f - \mathbf{z}^T (\mathbf{c} - \mathbf{s}), \quad (6)$$

where \mathbf{z} is a vector of Lagrangian multipliers or dual variables.

The KKT conditions of Eq. (4) with duality gap μ is written as

$$\nabla f - A^T \mathbf{z} = \mathbf{0}, \quad S\mathbf{z} - \mu \mathbf{1} = \mathbf{0}, \quad \mathbf{c} - \mathbf{s} = \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0}, \quad (7)$$

where $S = \text{diag}(\mathbf{s})$, and

$$A = \nabla \mathbf{c} = \begin{pmatrix} \nabla c_1 \\ \vdots \\ \nabla c_{2NM} \end{pmatrix} = \begin{pmatrix} I \\ -I \end{pmatrix}. \quad (8)$$

Here, I is an identity matrix.

To solve the system of the KKT conditions by using Newton's method, we have the following system of equations:

$$\begin{pmatrix} \nabla^2 L & \mathbf{0} & -A^T \\ \mathbf{0} & Z & S \\ A & -I & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{p}_\sigma \\ \mathbf{p}_s \\ \mathbf{p}_z \end{pmatrix} = - \begin{pmatrix} \nabla f - A^T \mathbf{z} \\ S\mathbf{z} - \mu \mathbf{1} \\ \mathbf{c} - \mathbf{s} \end{pmatrix}, \quad (9)$$

where $Z = \text{diag}(\mathbf{z})$ and $\nabla^2 L$ is the Hessian of the Lagrangian.

3.2 Solving the System Efficiently

The matrix in Eq. (9) is of the size $5NM \times 5NM$, which is sparse but large and computationally expensive to solve. We, therefore, develop an efficient way to solve the system by using the problem structure.

First, the system is explicitly written as follows:

$$\begin{cases} \nabla^2 L \mathbf{p}_\sigma - A^T \mathbf{p}_z = -\nabla f + A^T \mathbf{z} \\ Z \mathbf{p}_s + S \mathbf{p}_z = -S\mathbf{z} + \mu \mathbf{1} \\ A \mathbf{p}_\sigma - \mathbf{p}_s = -\mathbf{c} + \mathbf{s} \end{cases}. \quad (10)$$

Substituting the last equation into the second one yields

$$A^T S^{-1} Z (A \mathbf{p}_\sigma + \mathbf{c} - \mathbf{s}) + A^T \mathbf{p}_z = -A^T \mathbf{z} + \mu A^T S^{-1} \mathbf{1}, \quad (11)$$

and we add both sides to the first equation to obtain

$$\begin{aligned} \nabla^2 L \mathbf{p}_\sigma + A^T S^{-1} Z (A \mathbf{p}_\sigma + \mathbf{c} - \mathbf{s}) \\ = -A^T \mathbf{z} + \mu A^T S^{-1} \mathbf{1} - \nabla f + A^T \mathbf{z}. \end{aligned} \quad (12)$$

Here, we define $\mathbf{w} = \mathbf{s}^{-1} \odot \mathbf{z}$ and $\mathbf{y} = \mu \mathbf{s}^{-1} - \mathbf{w} \odot \mathbf{c} + \mathbf{z}$, where \odot is the Hadamard (element-wise) product, and \mathbf{s}^{-1} is a vector of element-wise reciprocals of \mathbf{s} . Then, we have

$$\mathbf{p}_\sigma = (\nabla^2 L + A^T S^{-1} Z A)^{-1} (-\nabla f + A^T \mathbf{y}), \quad (13)$$

$$\mathbf{p}_z = \mu \mathbf{s}^{-1} - \mathbf{w} \odot \mathbf{p}_s - \mathbf{z}. \quad (14)$$

By exploiting the structure of matrix A and defining $\mathbf{w} = (\mathbf{w}_l^T, \mathbf{w}_u^T)^T$ to split a vector into two parts corresponding to lower and upper bound constraints, we have

$$A^T S^{-1} Z A = \begin{pmatrix} I & -I \end{pmatrix} \text{diag}(\mathbf{w}) \begin{pmatrix} I \\ -I \end{pmatrix} = \text{diag}(\mathbf{w}_l + \mathbf{w}_u). \quad (15)$$

Similarly, we define $\mathbf{y} = (\mathbf{y}_l^T, \mathbf{y}_u^T)^T$ to simplify $A^T \mathbf{y}$ as $A^T \mathbf{y} = \mathbf{y}_l - \mathbf{y}_u$ and then

$$A \mathbf{p}_\sigma = \begin{pmatrix} I \\ -I \end{pmatrix} \mathbf{p}_\sigma = \begin{pmatrix} \mathbf{p}_\sigma \\ -\mathbf{p}_\sigma \end{pmatrix}. \quad (16)$$

Now, the solution is given by

$$\begin{cases} \mathbf{p}_\sigma = [\nabla^2 L + \text{diag}(\mathbf{w}_l + \mathbf{w}_u)]^{-1} (-\nabla f + \mathbf{y}_l - \mathbf{y}_u) \\ \mathbf{p}_s = \begin{pmatrix} \mathbf{p}_\sigma \\ -\mathbf{p}_\sigma \end{pmatrix} + \mathbf{c} - \mathbf{s} \\ \mathbf{p}_z = \mu \mathbf{s}^{-1} - \mathbf{w} \odot \mathbf{p}_s - \mathbf{z} \end{cases}, \quad (17)$$

which involves the inversion of the size $NM \times NM$.

3.3 Update Variables

Once \mathbf{p}_σ , \mathbf{p}_s , and \mathbf{p}_z are obtained, we then estimate the step length to update the parameters.¹⁶ The maximum of the step lengths is given by the following rule:

$$\begin{cases} \alpha_s^{\max} = \max\{\alpha \in [0, 1] | \mathbf{s} + \alpha \mathbf{p}_s \geq (1 - \tau) \mathbf{s}\} \\ \alpha_z^{\max} = \max\{\alpha \in [0, 1] | \mathbf{z} + \alpha \mathbf{p}_z \geq (1 - \tau) \mathbf{z}\} \end{cases}, \quad (18)$$

with $\tau \in (0, 1)$ used (for example, $\tau = 0.995$). This prevents variables \mathbf{s} and \mathbf{z} from approaching the lower boundary.

Next, we perform the backtracking line search³⁰ to estimate acceptable step lengths α_s and α_z . To this end, we use the following exact merit function ϕ with $\eta \in (0, 1)$:

$$\phi(\boldsymbol{\sigma}_t, \mathbf{s}) = f - \mu \sum_{i=1}^{2MN} \log(s_i) + v \|\mathbf{c}(x) - \mathbf{s}\|, \quad (19)$$

and make a sufficient decrease requirement

$$\phi(\boldsymbol{\sigma}_t + \alpha_s \mathbf{p}_\sigma, \mathbf{s} + \alpha_s \mathbf{p}_s) \leq \phi(\boldsymbol{\sigma}_t, \mathbf{s}) + \eta \alpha_s D_\phi(\boldsymbol{\sigma}_t, \mathbf{s}; \mathbf{p}_\sigma, \mathbf{p}_s), \quad (20)$$

where $D_\phi(\boldsymbol{\sigma}_t, \mathbf{s}; \mathbf{p}_\sigma, \mathbf{p}_s)$ denotes the directional derivative of ϕ in the direction $(\mathbf{p}_\sigma, \mathbf{p}_s)$. The step lengths α_s and α_z are found in the ranges $\alpha_s \in (0, \alpha_s^{\max}]$ and $\alpha_z \in (0, \alpha_z^{\max}]$ so that Eq. (20) is satisfied.

Then, the parameters $\boldsymbol{\sigma}_t$, \mathbf{s} , and \mathbf{z} are updated as

$$\begin{cases} \boldsymbol{\sigma}_t \leftarrow \boldsymbol{\sigma}_t + \alpha_s \mathbf{p}_\sigma \\ \mathbf{s} \leftarrow \mathbf{s} + \alpha_s \mathbf{p}_s \\ \mathbf{z} \leftarrow \mathbf{z} + \alpha_z \mathbf{p}_z \end{cases}. \quad (21)$$

Once the following error function is smaller than a given threshold, the PD interior point method stops

$$E(\boldsymbol{\sigma}_t, \mathbf{s}, \mathbf{z}; \mu) = \max\{\|\nabla f - A^T \mathbf{z}\|, \|S\mathbf{z} - \mu \mathbf{1}\|, \|\mathbf{c} - \mathbf{s}\|\}. \quad (22)$$

Algorithm 1 summarizes the PD interior point method developed above. Note that the Hessian $\nabla^2 L$ can be approximated as B_k at each iteration k by the quasi-Newton method, instead of the full Hessian used by Newton's method. We will compare Newton's method and the quasi-Newton method in the section of experiments.

4 Efficient Formulations

The most computationally intensive part of the PD algorithm shown above is the computation of Hessians for Newton's method and Jacobians for Newton's and quasi-Newton methods.

Algorithm 1 Primal-dual interior point with line search.

Data: $\mu > 0$, $\sigma \in (0,1)$, $\epsilon_{\text{TOL}} > 0$, $\epsilon_\mu, \eta \in (0,1)$, $k = 0$.

Input: A feasible initial estimates σ_t , $\mathbf{s} > 0$, and $\mathbf{z} > 0$.

Input: $B_0 = I$ // Only for quasi-Newton

Result: Estimates of σ_t

- 1 **repeat**
- 2 **repeat** // inner loop
- 3 Compute the decent direction $\mathbf{p} = (\mathbf{p}_\sigma, \mathbf{p}_s, \mathbf{p}_z)$
- 4 Compute the step lengths α_s and α_z
- 5 Update σ_t , \mathbf{s} , \mathbf{z}
- 6 Update the approximation B_k // Only for quasi-Newton
- 7 Set $k \leftarrow k + 1$
- 8 **until** $E(\sigma_{tk}, \mathbf{s}_k, \mathbf{z}_k; \mu) \leq \epsilon_\mu$;
- 9 $\mu \leftarrow \sigma\mu$
- 10 $\epsilon_\mu \leftarrow \mu$
- 11 **until** $E(\sigma_{tk}, \mathbf{s}_k, \mathbf{z}_k; 0) \leq \epsilon_{\text{TOL}}$;

We propose here efficient formulations of Hessian and Jacobian of the problem, whose computational cost is much smaller than naive formulations used in the previous approaches.

First, we show the naive and old formulations of Hessian and Jacobian, then introduce our new formulations, followed by discussions on computational cost.

4.1 Previous Old Formulations for Inverse Problem

Here, we show how the previous approaches^{12,13,15} do. We call these the old formulations.

4.1.1 Jacobian: old formulation

Remember that the objective function f to be minimized is

$$f = \sum_{i=1}^N \sum_{j=1}^N \left[I_{ij}^2 - 2I_{ij}I_0 \sum_{k=1}^{N_{ij}} H_{ijk} e^{-\sigma_t^T \mathbf{D}_{ijk}} + I_0^2 \sum_{k=1}^{N_{ij}} \sum_{l=1}^{N_{ij}} H_{ijk} H_{ijl} e^{-\sigma_t^T (\mathbf{D}_{ijk} + \mathbf{D}_{ijl})} \right]. \quad (23)$$

The gradient of f is given as follows by taking the derivative of the objective function:

$$\frac{\partial f}{\partial \sigma_t} = \sum_{i=1}^N \sum_{j=1}^N \left[2I_{ij}I_0 \sum_{k=1}^{N_{ij}} H_{ijk} e^{-\sigma_t^T \mathbf{D}_{ijk}} \mathbf{D}_{ijk} - I_0^2 \sum_{k=1}^{N_{ij}} \sum_{l=1}^{N_{ij}} H_{ijk} H_{ijl} e^{-\sigma_t^T (\mathbf{D}_{ijk} + \mathbf{D}_{ijl})} (\mathbf{D}_{ijk} + \mathbf{D}_{ijl}) \right]. \quad (24)$$

To simplify the equation, the following notations are introduced:

$$\mathbf{E}_{ij} = \begin{pmatrix} e^{-\sigma_t^T \mathbf{D}_{ij1}} \\ e^{-\sigma_t^T \mathbf{D}_{ij2}} \\ \vdots \\ e^{-\sigma_t^T \mathbf{D}_{ijN_{ij}}} \end{pmatrix}, \quad \mathbf{H}_{ij} = \begin{pmatrix} H_{ij1} \\ H_{ij2} \\ \vdots \\ H_{ijN_{ij}} \end{pmatrix},$$

$$D_{ij} = \begin{pmatrix} \mathbf{D}_{ij1} \\ \mathbf{D}_{ij2} \\ \vdots \\ \mathbf{D}_{ijN_{ij}} \end{pmatrix}, \quad (25)$$

$$\widetilde{D}_{ij} = \begin{pmatrix} \mathbf{D}_{ij1} + \mathbf{D}_{ij1} & \cdots & \mathbf{D}_{ij1} + \mathbf{D}_{ijN_{ij}} \\ \mathbf{D}_{ij2} + \mathbf{D}_{ij1} & \cdots & \mathbf{D}_{ij2} + \mathbf{D}_{ijN_{ij}} \\ \vdots & \ddots & \vdots \\ \mathbf{D}_{ijN_{ij}} + \mathbf{D}_{ij1} & \cdots & \mathbf{D}_{ijN_{ij}} + \mathbf{D}_{ijN_{ij}} \end{pmatrix}. \quad (26)$$

Now, f and the gradient are rewritten as follows:

$$f = \sum_{i=1}^N \sum_{j=1}^N [I_{ij}^2 - 2I_{ij}I_0 \mathbf{E}_{ij}^T \mathbf{H}_{ij} + I_0^2 (\mathbf{E}_{ij}^T \mathbf{H}_{ij})^2], \quad (27)$$

$$\frac{\partial f}{\partial \sigma_t} = \sum_{i=1}^N \sum_{j=1}^N (2I_{ij}I_0 \text{sum}[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) \otimes D_{ij}] - I_0^2 \text{sum}\{[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T] \otimes \widetilde{D}_{ij}\}), \quad (28)$$

where $\text{sum}\{\}$ stands for the sum over the elements of the container [Eq. (26)] of vectors, and \otimes denotes the tensor product.

4.1.2 Hessian: old formulation

We define the following notations:

$$\boldsymbol{\beta}_{ijx} = \begin{pmatrix} D_{ij1x} \\ D_{ij2x} \\ \vdots \\ D_{ijkx} \end{pmatrix}, \quad (29)$$

$$\boldsymbol{\gamma}_{ijx} = \begin{pmatrix} D_{ij1x} + D_{ij1x} & \cdots & D_{ij1x} + D_{ijkx} \\ D_{ij2x} + D_{ij1x} & \cdots & D_{ij2x} + D_{ijkx} \\ \vdots & \ddots & \vdots \\ D_{ijkx} + D_{ij1x} & \cdots & D_{ijkx} + D_{ijkx} \end{pmatrix}, \quad (30)$$

where D_{ijkx} stands for the x 'th element of \mathbf{D}_{ijk} .

Now, the second-order derivate of f can be represented as follows:

$$\frac{\partial^2 f}{\partial^2 \sigma_t} = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,NM} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,NM} \\ \vdots & \vdots & \cdots & \vdots \\ h_{NM,1} & h_{NM,2} & \cdots & h_{NM,NM} \end{pmatrix}, \quad (31)$$

where each element is given by

$$h_{xy} = - \sum_{i=1}^N \sum_{j=1}^N I_{ij} I_0 \text{sum}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij} \odot \boldsymbol{\beta}_{ijx} \odot \boldsymbol{\beta}_{ijy}) + \sum_{i=1}^N \sum_{j=1}^N I_0^2 \text{sum}[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T \odot \boldsymbol{\gamma}_{ijx} \odot \boldsymbol{\gamma}_{ijy}]. \quad (32)$$

4.2 Proposed New Efficient Formulation

The problem of the previous old formulations of Jacobian and Hessian shown above is the computation cost increasing as the number N_{ij} of paths increases. As discussed later, the computation cost is $O(N_{ij}^2)$ on average.

The idea of the proposed formulation is to explore the property of the exponential function and its derivative in the problem. As shown below, the computation cost can be reduced to $O(N_{ij})$ on average.

4.2.1 Jacobian: new formulation

First, we rewrite the cost function as follows:

$$f = \sum_{i=1}^N \sum_{j=1}^N r_{ij}^2, \quad (33)$$

where r_{ij} is a residual

$$r_{ij} = I_{ij} - I_0 \sum_{k=1}^{N_{ij}} H_{ijk} e^{-\boldsymbol{\sigma}_t^T \mathbf{D}_{ijk}} = I_{ij} - I_0 \mathbf{E}_{ij}^T \mathbf{H}_{ij}. \quad (34)$$

Now we use the chain rule of differentiation, and we have

$$\frac{\partial f}{\partial \boldsymbol{\sigma}_t} = \sum_{i=1}^N \sum_{j=1}^N 2r_{ij} \frac{\partial r_{ij}}{\partial \boldsymbol{\sigma}_t}, \quad (35)$$

where

$$\frac{\partial r_{ij}}{\partial \boldsymbol{\sigma}_t} = I_0 \sum_{k=1}^{N_{ij}} H_{ijk} e^{-\boldsymbol{\sigma}_t^T \mathbf{D}_{ijk}} \mathbf{D}_{ijk} = I_0 D_{ij} (\mathbf{E}_{ij} \odot \mathbf{H}_{ij}). \quad (36)$$

Here, we define [Note that this is not the same as the one defined in the previous approaches above, which is a structure used in MATLAB codes to store \mathbf{D}_{ijk} . Here, D_{ij} is an $(NM) \times N_{ij}$ matrix.]

$$D_{ij} = (\mathbf{D}_{ij1}, \mathbf{D}_{ij2}, \dots, \mathbf{D}_{ijN_{ij}}), \quad (37)$$

which has N_{ij} vectors of dimension NM , and $(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})$ is a coefficient vector. Therefore,

$$\frac{\partial f}{\partial \boldsymbol{\sigma}_t} = \sum_{i=1}^N \sum_{j=1}^N 2I_0 (I_{ij} - I_0 \mathbf{E}_{ij}^T \mathbf{H}_{ij}) D_{ij} (\mathbf{E}_{ij} \odot \mathbf{H}_{ij}). \quad (38)$$

4.2.2 Discussion

Suppose that the expectation of the number of paths is $\bar{N} = E[N_{ij}]$. Then, the computations for the proposed new formulation of Jacobian above are:

- $O(\bar{N})$ multiplications for $\mathbf{E}_{ij} \odot \mathbf{H}_{ij}$,
- $O(\bar{N})$ additions for $\mathbf{E}_{ij}^T \mathbf{H}_{ij}$,
- $O(\bar{N}NM)$ multiplications and additions for $D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})$ because there are $O(\bar{N})$ vectors of dimension NM ,

for each i and j . In total, it takes $O(\bar{N}N^3M)$ operations to compute NM elements of the Jacobian or $O(\bar{N}N^2)$ operations per element.

Contrary, for each i and j , the previous old formulation Eq. (28) needs:

- $O(\bar{N})$ multiplications for $\mathbf{E}_{ij} \odot \mathbf{H}_{ij}$,
- $O(\bar{N}NM)$ multiplications for $(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) \otimes D_{ij}$ because there are $O(\bar{N})$ vectors of dimension NM ,
- $O(\bar{N}NM)$ additions for $\text{sum}[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) \otimes D_{ij}]$,

for the first term, and

- $O(\bar{N}^2)$ multiplications for $(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T$,
- $O(\bar{N}^2NM)$ additions for computing \widetilde{D}_{ij} because there are $O(\bar{N}^2)$ vectors of dimension NM ,
- $O(\bar{N}^2NM)$ multiplications for $[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T] \otimes \widetilde{D}_{ij}$,
- $O(\bar{N}^2NM)$ additions for $\text{sum}\{[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T] \otimes \widetilde{D}_{ij}\}$,

for the second term. In total, it takes $O(\bar{N}^2N^3M)$ operations to compute NM elements of the Jacobian, or $O(\bar{N}^2N^2)$ operations per element. The difference is mainly caused by the second term of Eq. (28).

In summary, the proposed new formulation has the cost of $O(\bar{N}N^2)$ operations per element, whereas the previous old formulation has the cost of $O(\bar{N}^2N^2)$ operations per element. Table 1 summarizes the discussion above.

4.2.3 Hessian: new formulation

In the same manner, we can derive the Hessian as follows. From the Jacobian

Table 1 Comparison of the new and old formulations for computing the Jacobian.

Terms	New	Old
$\mathbf{E}_{ij} \odot \mathbf{H}_{ij}$	$O(\bar{N})$	$O(\bar{N})$
$\mathbf{E}_{ij}^T \mathbf{H}_{ij}$	$O(\bar{N})$	
$D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})$	$O(\bar{N}NM)$	
$(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) \otimes D_{ij}$		$O(\bar{N}NM)$
$\text{sum}[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) \otimes D_{ij}]$		$O(\bar{N}NM)$
$(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T$		$O(\bar{N}^2)$
\widetilde{D}_{ij}		$O(\bar{N}^2NM)$
$[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T] \otimes \widetilde{D}_{ij}$		$O(\bar{N}^2NM)$
$\text{sum}\{[(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T] \otimes \widetilde{D}_{ij}\}$		$O(\bar{N}^2NM)$
Per element	$O(\bar{N}N^2)$	$O(\bar{N}^2N^2)$

$$\frac{\partial f}{\partial \boldsymbol{\sigma}_t} = \sum_{i=1}^N \sum_{j=1}^N 2r_{ij} \frac{\partial r_{ij}}{\partial \boldsymbol{\sigma}_t}, \quad (39)$$

we obtain the Hessian by using the chain rule of differentiation

$$\frac{\partial^2 f}{\partial \boldsymbol{\sigma}_t^2} = \sum_{i=1}^N \sum_{j=1}^N 2 \frac{\partial r_{ij}}{\partial \boldsymbol{\sigma}_t} \frac{\partial r_{ij}}{\partial \boldsymbol{\sigma}_t}^T + 2r_{ij} \frac{\partial^2 r_{ij}}{\partial \boldsymbol{\sigma}_t^2}, \quad (40)$$

where

$$\frac{\partial^2 r_{ij}}{\partial \boldsymbol{\sigma}_t^2} = -I_0 \sum_{k=1}^{N_{ij}} H_{ijk} e^{-\boldsymbol{\sigma}_t^T \mathbf{D}_{ijk}} \mathbf{D}_{ijk} \mathbf{D}_{ijk}^T, \quad (41)$$

$$= -I_0 D_{ij} \text{diag}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) D_{ij}^T. \quad (42)$$

Now, the Hessian can be written as follows:

$$\begin{aligned} \frac{\partial^2 f}{\partial \boldsymbol{\sigma}_t^2} &= \sum_{i=1}^N \sum_{j=1}^N 2I_0^2 D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) [D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})]^T \\ &\quad - 2I_0 (I_{ij} - I_0 \mathbf{E}_{ij}^T \mathbf{H}_{ij}) D_{ij} \text{diag}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) D_{ij}^T. \end{aligned} \quad (43)$$

Note that $D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) [D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})]^T$ should not be expanded as $D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T D_{ij}^T$ because it involves a large matrix $(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T$, which is computationally intensive to compute.

4.2.4 Discussion

By reusing $\mathbf{E}_{ij} \odot \mathbf{H}_{ij}$ and $D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})$ computed for the Jacobian, the new formulation of Hessian needs:

- $O(N^2 M^2)$ multiplications for $D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) [D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})]^T$,
- $O(\bar{N} N M)$ multiplications for $D_{ij} \text{diag}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})$,
- $O(\bar{N} N^2 M^2)$ multiplications for $D_{ij} \text{diag}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) D_{ij}^T$,

for each i and j . In total, it takes $O(\bar{N} N^4 M^2)$ operations to compute $N^2 M^2$ elements of the Hessian or $O(\bar{N} N^2)$ operations per element.

Contrarily, for each element, the previous formulation [Eq. (32)] needs:

- $O(\bar{N}^2)$ multiplications for $\boldsymbol{\gamma}_{ijx}$,
- $O(\bar{N})$ multiplications and the sum for $(\mathbf{E}_{ij} \odot \mathbf{H}_{ij} \odot \boldsymbol{\beta}_{ijx} \odot \boldsymbol{\beta}_{ijy})$,
- $O(\bar{N}^2)$ multiplications for $(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T$,
- $O(\bar{N}^2)$ multiplications and the sum for $(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T \odot \boldsymbol{\gamma}_{ijx} \odot \boldsymbol{\gamma}_{ijy}$,
- $O(\bar{N}^2 N^2)$ additions for the sums of i and j .

In total, it takes $O(\bar{N}^2 N^2)$ operations to compute a single element of the Hessian.

In summary, the proposed new formulation has the cost of $O(\bar{N} N^2)$ operations per element, whereas the previous old formulation has the cost of $O(N^2 \bar{N}^2)$ operations per element. Table 2 summarizes the discussion above.

Table 2 Comparison of the new and old formulations for computing the Hessian.

Terms	New	Old
$D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) [D_{ij}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})]^T$	$O(N^2 M^2)$	
$D_{ij} \text{diag}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})$	$O(\bar{N} N M)$	
$D_{ij} \text{diag}(\mathbf{E}_{ij} \odot \mathbf{H}_{ij}) D_{ij}^T$	$O(\bar{N} N^2 M^2)$	
$(\mathbf{E}_{ij} \odot \mathbf{H}_{ij} \odot \boldsymbol{\beta}_{ijx} \odot \boldsymbol{\beta}_{ijy})$		$O(\bar{N})$
$(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T$		$O(\bar{N}^2)$
$(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})(\mathbf{E}_{ij} \odot \mathbf{H}_{ij})^T \odot \boldsymbol{\gamma}_{ijx} \odot \boldsymbol{\gamma}_{ijy}$		$O(\bar{N}^2)$
Sums of i, j		$O(N^2 \bar{N}^2)$
Per element	$O(\bar{N} N^2)$	$O(N^2 \bar{N}^2)$

5 Numerical Simulation

In this section, we report the results obtained by simulations using the proposed method by comparing PD and LB interior point methods, as well as old and new formulations of Jacobian and Hessian.

Since the mathematical model we used to describe the light transport in the forward problem is exactly the same as the model in the previous work,¹⁵ we use the same setup as follows. For the 2-D layered medium, the grid size was set to $N = M = 24$ with square voxels of size 1 (mm), i.e., the medium is 24 (mm) \times 24 (mm). The values of the extinction coefficients are set between 1.05 and 1.55 (mm^{-1}), and the lower and upper bounds (σ_{tl} and σ_{tu}) are set to be 1.0 and 2.0 (mm^{-1}), respectively. Values of the initial guess are 1.001 for all elements of $\boldsymbol{\sigma}_{t0}$, as well as \mathbf{s}_0 and \mathbf{z}_0 . Parameters used in Algorithm 1 are set as $\sigma = 0.5$, $\eta = 0.01$, $\epsilon_\mu = 1$, and $\epsilon_{\text{TOT}} = 0.02$.

5.1 Estimation Quality

The ground truth and the estimated results are shown in Fig. 3. The matrix plots in the top row of this figure represent five different media (a)–(e) used for the simulation, which were also used in the previous work.¹⁵ Note that the medium e is the Shepp–Logan phantom.³¹ Each voxel is shaded in gray according to the values of the extinction coefficients.

The following rows show the estimated results of different combinations of LB or PD methods, old or new formulas, and Newton's or quasi-Newton methods. The proposed method is PD-new-Newton/quasi-Newton; that is, the PD method with Newton's or quasi-Newton method by using the proposed new formulation. The row LB-old-quasi-Newton corresponds to the previous work¹⁵ that uses the LB method with the quasi-Newton method by using the old formulation, and the row LB-old-Newton corresponds to another prior work.¹²

As we can see, the results of different combinations almost look the same for each of the five media. This observation is also validated by the root-mean-square error (RMSE) shown in Table 3. The RMSE values of all combinations are more or less the same, while some variations appear due to the different update rules between Newton's and quasi-Newton methods, and different stopping conditions between LB and PD methods.

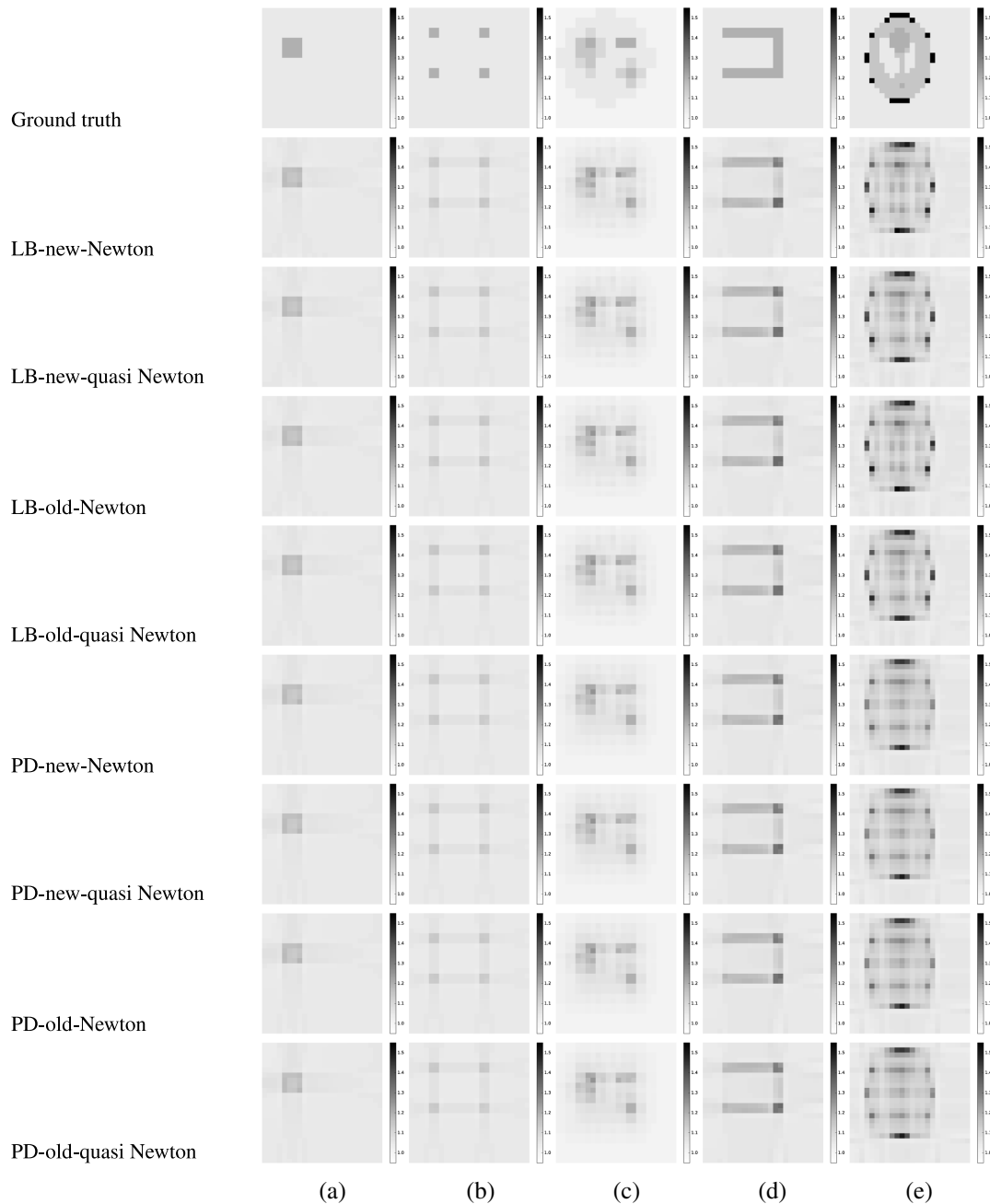


Fig. 3 Numerical simulation results for five media (a)–(e) in a grid of 24×24 . Darker shades of gray represent larger values of extinction coefficients (more light is absorbed at the voxel). The bars on the side show extinction coefficient values in gray scale. The first row shows ground truth for five different types of media used for the simulation. The following rows show the estimated results for different combinations of LB or PD methods, old or new formulas, and Newton's or quasi-Newton methods.

5.2 Estimation Time

The main goal of this paper is to develop an efficient way to solve the inverse problem. Table 3 shows the computation cost of different combinations. All experiments were performed on a Linux workstation (two Intel Xeon E5-2630 2.4 GHz CPUs, total 16 physical cores, with 256 GB memory). We implemented the method in MATLAB R2017a and did not explicitly use the parallel computation toolbox of MATLAB except the Hessian computation of LB/PD-old-Newton due to its slow computation. Parallel matrix multiplications are, however, automatically performed on MATLAB. Table 3 shows the

computation time for different computations in seconds. We report the average and standard deviation of 10 trials, except the cases of LB/PD-old-Newton which show the processing time of a single trial.

For any combination, our proposed new formulation is much faster than the old formulation. The uses of Newton's method greatly benefit from the efficient Hessian computation and the computation time reduces more than a factor of 100. However, the new formulation does not help to reduce the computation cost of quasi-Newton methods and the reduction is a factor of just 2 or 3. This is due to the fact that the quasi-Newton

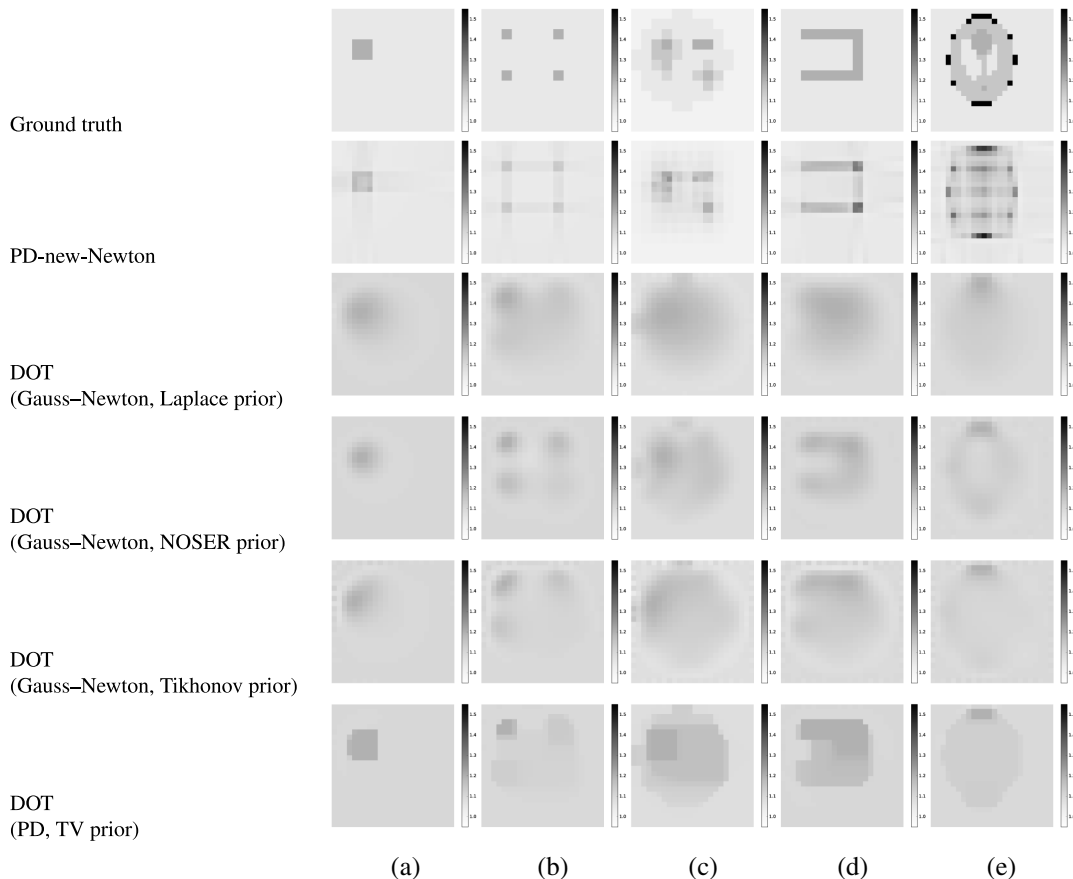


Fig. 4 Numerical simulation results for five media (a)–(e) in a grid of 24×24 . Darker shades of gray represent larger values of extinction coefficients (more light is absorbed at the voxel). The bars on the side show extinction coefficient values in gray scale. The first row shows ground truth for five different types of media used for the simulation. The following rows show the estimated results for different combinations of LB or PD methods, old or new formulas, and Newton's or quasi-Newton methods. Results of the previous work¹⁵ and DOT are shown as baselines in the last three rows.

method needs gradient vectors, and its computation is of the order NM (the number of voxels) and it is not so large in terms of the total computation cost. In contrast, Hessian computation in the Newton's method is of the size $NM \times NM$, which is quite large compared to the gradient. Our new formulation is, therefore, better when the Newton method is used.

With the quasi-Newton method, the PD approach seems to be comparable to the LB method. By comparing rows LB/PD-new-quasi-Newton, LB is faster than PD for denser media (c), (d), and (e). This might be caused by the different ways of approximations by the quasi-Newton method. For the LB method, the gradient is modified by the approximated Hessian. For the PD method, however, the approximated Hessian is used in the matrix to be solved, resulting in an update rule of \mathbf{p}_σ regularized by diagonal elements of \mathbf{w} in Eq. (17). Except the simplest medium (a), the fast combination is PD-new-Newton, which is proposed in this paper. This is due to the fast convergence of Newton's method compared to the quasi-Newton method, and also the fact that the PD method needs fewer iterations than the LB method. The qualities of results are almost the same as discussed above, then PD-new-Newton is the best when the working memory is enough for storing the Hessian. Otherwise, LB/PD-new-quasi-Newton is better to be used.

5.3 Comparison

We compare our methods to a standard DOT implemented in the Electrical Impedance Tomography and Diffuse Optical Tomography Reconstruction Software (EIDORS)^{32,33} in the same setting as the previous work:¹⁵ $N = M = 24$ medium of size $24 \text{ (mm)} \times 24 \text{ (mm)}$ with the five media (a)–(e). For solving DOT by EIDORS, we used $24 \times 24 \times 24 = 1152$ triangle elements. For boundary conditions, we placed 48 light sources and detectors at the same intervals around the medium. We used some different solvers and priors; Gauss-Newton method³⁴ with Laplace, NOSER,³⁵ and Tikhonov priors, and PD method with total variation prior.

Due to the diffusion approximation of DOT, the results in Fig. 4 for DOT with the Gauss-Newton method are blurred, and those for DOT with PD have a tendency of overestimating the high-coefficient value areas. In contrast, the results of PD-new-Newton (and other combinations in Fig. 3) are clearer and sharper for all combinations. This observation is also validated by the RMSE shown in Table 3. The RMSE values of PD-new-Newton are smaller than the values of DOT for all five media.

The obvious drawback of PD-new-Newton is its high computation cost. It is slower by a factor of 10 compared to DOT with the PD method and a factor of 100 to DOT with the Gauss-Newton method. A large part of the computation cost comes

Table 3 RMSEs and computation time for the numerical simulations for five different types of media (a)–(e) with grid size of 24×24 , for different combinations of LB or PD methods, old or new formulas, and Newton's or quasi-Newton methods. Each computation time shows the mean and standard deviation of 10 trials, except the combinations of "old-Newton." Note that RMSE values are exactly the same for 10 trials. Results of DOT methods are shown for comparison.

		a	b	c	d	e
RMSE	LB-new-Newton	0.008422	0.012643	0.014594	0.021246	0.052511
	LB-new-quasi-Newton	0.008646	0.012478	0.014444	0.020375	0.049811
	LB-old-Newton	0.008422	0.012643	0.014594	0.021246	0.052420
	LB-old-quasi-Newton	0.008646	0.012478	0.014444	0.020375	0.049818
	PD-new-Newton	0.009776	0.013190	0.014490	0.021251	0.055912
	PD-new-quasi-Newton	0.009754	0.013184	0.014502	0.021201	0.056085
	PD-old-Newton	0.009776	0.013190	0.014490	0.021251	0.055912
	PD-old-quasi-Newton	0.009754	0.013184	0.014502	0.021201	0.056084
	DOT (GN, Laplace prior)	0.059339	0.062984	0.078100	0.065001	0.087094
	DOT (GN, NOSER prior)	0.052053	0.057515	0.075478	0.059156	0.086397
	DOT (GN, Tikhonov prior)	0.054729	0.056196	0.073146	0.059284	0.087659
	DOT (PD, TV prior)	0.055047	0.059219	0.081811	0.070263	0.086107
	Computation time (s)	LB-new-Newton	60.00 ± 4.60	57.63 ± 1.41	61.90 ± 2.88	62.32 ± 1.22
LB-new-quasi-Newton		18.64 ± 0.90	17.22 ± 1.03	25.32 ± 1.21	20.73 ± 1.10	32.57 ± 0.58
LB-old-Newton		126100	12848	13383	14037	21577
LB-old-quasi-Newton		44.86 ± 1.33	42.58 ± 1.19	63.75 ± 1.76	54.05 ± 2.04	91.17 ± 1.48
PD-new-Newton		18.73 ± 2.18	16.52 ± 0.67	17.40 ± 0.93	18.28 ± 1.30	23.14 ± 1.32
PD-new-quasi-Newton		14.44 ± 0.78	13.07 ± 0.61	40.25 ± 1.01	30.77 ± 0.86	48.26 ± 1.59
PD-old-Newton		5673	5418	5824	5663	7547
PD-old-quasi-Newton		75.67 ± 1.34	67.18 ± 1.38	203.42 ± 3.42	155.69 ± 2.67	248.05 ± 4.86
DOT (GN, Laplace prior)		0.34 ± 0.04	0.41 ± 0.04	0.40 ± 0.03	0.40 ± 0.04	0.40 ± 0.03
DOT (GN, NOSER prior)		0.52 ± 0.04	0.52 ± 0.05	0.53 ± 0.05	0.50 ± 0.01	0.52 ± 0.03
DOT (GN, Tikhonov prior)		0.29 ± 0.01	0.29 ± 0.02	0.30 ± 0.03	0.29 ± 0.01	0.29 ± 0.02
DOT (PD, TV prior)		2.67 ± 0.07	2.68 ± 0.06	2.67 ± 0.07	2.64 ± 0.05	2.66 ± 0.07

from the computation of Hessian and Jacobian, which depends on the number of paths N_{ij} . Further reduction of computation cost is left for our future work.

6 Conclusion

In this paper, we have proposed a PD approach to optical tomography with a discretized path integral and also efficient formulation for computing Hessian and Jacobian. Numerical simulation examples with 2-D layered media are shown to demonstrate that the proposed method, called PD-new-Newton in the experiments, performed faster than the previous work (LB-old-quasi-Newton) while the estimated extinction coefficients of both methods were comparable. Compared to DOT,

the proposed method worked slower but produced better estimation results in terms of RMSE.

Disclosures

No other conflicts of interest, financial or otherwise, are declared by the authors.

Acknowledgments

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant No. JP26280061.

References

1. S. R. Arridge and M. Schweiger, "Image reconstruction in optical tomography," *Phil. Trans. R. Soc. B* **352**(1354), 717–726 (1997).
2. S. R. Arridge and J. C. Hebden, "Optical imaging in medicine: II. modelling and reconstruction," *Phys. Med. Biol.* **42**(5), 841–853 (1997).
3. J. C. Hebden, S. R. Arridge, and D. T. Delpy, "Optical imaging in medicine: I. experimental techniques," *Phys. Med. Biol.* **42**(5), 825–840 (1997).
4. S. R. Arridge, "Optical tomography in medical imaging," *Inverse Probl.* **15**, R41–R93 (1999).
5. S. R. Arridge and J. C. Schotland, "Optical tomography: forward and inverse problems," *Inverse Probl.* **25**(12), 123010 (2009).
6. G. Bal, "Inverse transport theory and applications," *Inverse Probl.* **25**(5), 053001 (2009).
7. K. Ren, "Recent developments in numerical techniques for transport-based medical imaging methods," *Commun. Comput. Phys.* **8**, 1–50 (2010).
8. A. Charette, J. Boulanger, and H. K. Kim, "An overview on recent radiation transport algorithm development for optical tomography imaging," *J. Quantum Spectrosc. Radiat. Transfer* **109**(17–18), 2743–2766 (2008).
9. Y. Mukaigawa, R. Raskar, and Y. Yagi, "Analysis of scattering light transport in translucent media," *IPSP Trans. Comput. Vision Appl.* **3**, 122–133 (2011).
10. Y. Dobashi et al., "An inverse problem approach for automatically adjusting the parameters for rendering clouds using photographs," *ACM Trans. Graph.* **31**(6), 1 (2012).
11. Y. Ishii et al., "Scattering tomography by Monte Carlo voting," in *IAPR Int. Conf. on Machine Vision Applications* (2013).
12. T. Tamaki et al., "Multiple-scattering optical tomography with layered material," in *2013 Int. Conf. on Signal-Image Technology & Internet-Based Systems*, pp. 93–99, IEEE (2013).
13. B. Yuan et al., "Layered optical tomography of multiple scattering media with combined constraint optimization," in *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pp. 1–6, IEEE (2015).
14. R. Akashi et al., "Scattering tomography using ellipsoidal mirror," in *21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV 2015)*, Mokpo, South Korea, pp. 1–5 (2015).
15. B. Yuan et al., "Optical tomography with discretized path integral," *J. Med. Imaging* **2**, 033501 (2015).
16. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge (2004).
17. B. Yuan et al., "Optical tomography with discretized path integrals: a comparison with log-barrier and primal-dual methods," in *The Korea-Japan joint workshop on Frontiers of Computer Vision (FCV 2016)*, pp. 378–382 (2016).
18. T. Tamaki et al., "Efficient formulations of optical tomography with discretized path integral," in *The 23th International Workshop on Frontiers of Computer Vision (FCV 2017)*, pp. 1–5 (2017).
19. G. Zhang et al., "Acceleration of dynamic fluorescence molecular tomography with principal component analysis," *Biomed. Opt. Express* **6**, 2036 (2015).
20. H. M. Huang and I. T. Hsiao, "Accelerating an ordered-subset low-dose X-ray cone beam computed tomography image reconstruction with a power factor and total variation minimization," *PLoS One* **11**(4), e0153421 (2016).
21. G. Pratz and L. Xing, "GPU computing in medical physics: a review," *Med. Phys.* **38**(5), 2685–2697 (2011).
22. J. Lobera et al., "High performance computing for a 3-D optical diffraction tomographic application in fluid velocimetry," *Opt. Express* **23**(4), 4021 (2015).
23. S. Ha et al., "GPU-accelerated forward and back-projections with spatially varying kernels for 3D DIRECT TOF PET reconstruction," *IEEE Trans. Nuclear Sci.* **60**(1), 166–173 (2013).
24. F. Xu and K. Mueller, "Accelerating popular tomographic reconstruction algorithms on commodity PC graphics hardware," *IEEE Trans. Nuclear Sci.* **52**(3), 654–663 (2005).
25. V. G. Nguyen and S. J. Lee, "GPU-accelerated iterative reconstruction from Compton scattered data using a matched pair of conic projector and backprojector," *Comput. Methods Programs Biomed.* **131**, 27–36 (2016).
26. R. Liu, Y. Luo, and H. Yu, "GPU-based acceleration for interior tomography," *IEEE Access* **2**, 841–855 (2014).
27. M. Schweiger, "GPU-accelerated finite element method for modelling light transport in diffuse optical tomography," *Int. J. Biomed. Imaging* **2011**, 1–11 (2011).
28. S. Q. Zheng et al., "A distributed multi-GPU system for high speed electron microscopic tomographic reconstruction," *Ultramicroscopy* **111**(8), 1137–1143 (2011).
29. A. Borsic, E. A. Attardo, and R. J. Halter, "Multi-GPU Jacobian accelerated computing for soft-field tomography," *Physiol. Meas.* **33**, 1703–1715 (2012).
30. J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York (2006).
31. L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *IEEE Trans. Nucl. Sci.* **21**, 21–43 (1974).
32. A. Adler and W. R. Lionheart, "EIDORS: towards a community-based extensible software base for EIT," in *6th Conf. on Biomedical Applications of Electrical Impedance Tomography*, London, UK, pp. 1–4 (2005).
33. A. Adler and W. R. Lionheart, "Uses and abuses of EIDORS: an extensible software base for EIT," *Physiol. Meas.* **27**(5), S25 (2006).
34. A. Adler and R. Guardo, "Electrical impedance tomography: regularized imaging and contrast detection," *IEEE Trans. Med. Imaging* **15**, 170–179 (1996).
35. M. Cheney et al., "NOSER: an algorithm for solving the inverse conductivity problem," *Int. J. Imaging Syst. Technol.* **2**(2), 66–75 (1990).

Bingzhi Yuan received his BE degree in software engineering from Beijing University of Posts and Telecommunications, China, and his ME degree in engineering from Hiroshima University, Japan, in 2010 and 2013, respectively. Currently, he is a PhD student at Hiroshima University.

Toru Tamaki received his BE, ME, and PhD degrees in information engineering from Nagoya University, Japan, in 1996, 1998, and 2001, respectively. After being an assistant professor at Niigata University, Japan, from 2001 to 2005, he is currently an associate professor in the Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Japan. His research interests include computer vision and image recognition.

Bisser Raytchev received his PhD in informatics from Tsukuba University, Japan, in 2000. After being a research associate at NTT Communication Science Labs and AIST, he is presently an assistant professor in the Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Japan. His current research interests include computer vision, pattern recognition, high-dimensional data analysis, and image processing.

Kazufumi Kaneda received his BE, ME, and DE degrees from Hiroshima University, Japan, in 1982, 1984, and 1991, respectively. He is a professor in the Department of Information Engineering, Graduate School of Engineering, Hiroshima University. In 1986, he joined Hiroshima University. He was a visiting researcher at Brigham Young University from 1991 to 1992. His research interests include computer graphics and scientific visualization.