A SPLINE-BASED FORWARD MODEL FOR OPTICAL DIFFUSE TOMOGRAPHY

Jean-Charles Baritaux, S. Chandra Sekhar, Michael Unser

Biomedical Imaging Group, EPFL, CH-1015 Lausanne, Switzerland

ABSTRACT

Reconstruction algorithms for Optical Diffuse Tomography (ODT) rely heavily on fast and accurate forward models. Arbitrary geometries and boundary conditions need to be handled rigorously since they are the only input to the inverse problem. From this perspective, Finite Element Methods (FEM) are good candidates to implement a forward model. However, these methods require to mesh the domain of interest, which is impractical on a routine basis. The other downside of the FEM is that the basis functions are often not compatible with the ones used for solving the inverse problem, which typically have less degrees of freedom. In this work, we tackle the 2D problem, and propose a forward model that uses a mesh-free discretization based on linear B-Splines. It combines the advantages of the FEM, while offering a fast and much simpler way of handling complex geometries. Another motivation for this work is that the underlying B-spline model is equally suitable for the subsequent reconstruction part of the process (solving the inverse problem). In particular, it is compatible with wavelets and multiresolution-type signal representations.

Index Terms— Diffuse Optical Tomography, Forward model, Mesh-free Finite Element Method, B-Spline

1. INTRODUCTION

In ODT, one is interested in recovering optical properties of a diffusive medium, as well as positions and intensities of some internal light sources, from peripheral light measurements [1]. Typically, the medium is a biological tissue where the propagation of light is strongly affected by absorption and scattering events. In the wavelength window of interest (around the near infra-red), scattering dominates over absorption : instead of traveling in straight line, photons undergo random changes of direction, due to interactions with the medium [1, 2].

For high scatterer densities ($> 1\%$), and on large enough paths, photon transport is aptly approximated by a diffusion process [3]. The following diffusion equation relates the fluence rate φ to the diffusion and absorption coefficients (respectively D and μ_a), and to a source term q in a domain Ω :

$$-\nabla.(D\nabla\varphi(\mathbf{r})) + \mu_a\varphi(\mathbf{r}) = q(\mathbf{r})\,, \quad \forall\,\mathbf{r}\in\Omega \tag{1}$$

Note that D and μ_a can also be space-varying. The following Robin (or Poisson) boundary conditions are associated :

$$\varphi(\mathbf{r}) + 2\,D\,\frac{\partial\varphi}{\partial\mathbf{n}}(\mathbf{r}) = 0\,, \quad \forall\,\mathbf{r}\in\partial\Omega \tag{2}$$

The forward model consists in solving (1) and (2). It is a crucial step in building the system matrix which is then used for solving the inverse problem.

The forward model in ODT is usually solved by analytical Green's functions techniques (assuming some idealized geometry), or by using standard FEM packages that can handle arbitrary geometries as well as space-varying coefficients. This type of formulation constitutes the state-of-the-art in the field of ODT, and has led to the reconstruction of reasonably accurate concentration maps of fluorophores [2, 4].

In this paper, we propose a Spline-based alternative that we believe to be better suited for the ODT inverse problem. We focus the 2D case, but our approach is extendable to 3D as well.

Our method is similar in spirit to the FEM, and it has the following advantages :

- In contrast with the FEM, it does not require any preliminary meshing of the domain of interest. It uses basis functions on a regular cartesian grid.
- Our method implicitly accounts for the boundary conditions (2), by including them in the system matrix (using the weak formulation of the diffusion equation). By contrast, in the FEM, boundary conditions are usually imposed through the choice of tailor-made boundary basis functions.
- Our method can handle arbitrary (polygonal) geometries, as opposed to analytical methods which are applicable only to a few selected geometries for which explicit Green's functions expressions are available (slab and cylinder for instance).
- The approach can handle space-varying diffusion and absorption properties of the medium.

This work was supported by the grant 20020-109415 from the Swiss Foundation, by the grant 1033 from the Hasler Foundation, and by the School of Engineering of EPFL.

- The reconstruction space admits a multi-resolution decomposition, and can also be spanned by wavelets. This is a key advantage for solving the inverse problem in a subsequent step : it is conceptually better to use the same basis functions for the forward and the inverse problem.

2. DESCRIPTION OF THE FORWARD MODEL

2.1. Weak formulation of the diffusion equation

By multiplying (1) with a test function ψ and integrating over the domain, we get the following equality :

$$\int_\Omega \left(-\nabla.(D\,\nabla\varphi) + \mu_a\,\varphi\right)\,\psi\,\mathrm{d}\mathbf{r} = \int_\Omega q\,\psi\,\mathrm{d}\mathbf{r} \tag{3}$$

Using integration by parts (Green's theorem), and incorporating the boundary conditions, we derive another integral relation :

$$\int_\Omega \left(D\,\nabla\varphi.\nabla\psi + \mu_a\,\varphi\,\psi\right)\,\mathrm{d}\mathbf{r} + \frac{1}{2}\int_{\partial\Omega}\varphi\,\psi\,\mathrm{d}\sigma = \int_\Omega q\,\psi\,\mathrm{d}\mathbf{r} \tag{4}$$

The set of equalities (4) written for every test function ψ is called the *weak formulation* of (1) with boundary conditions given by (2). In the following, the left hand side of (4) is denoted $a(\varphi,\psi)$, and the right hand side $L(\psi)$. With these notations, equation (4) reads : $a\,(\varphi,\,\psi) = L(\psi)$. In this expression, $a(.,.)$ is bilinear symmetric and L is linear.

Let us call V the set of test functions; provided V is a large enough set, there exists a unique $\varphi \in V$ such that

$$\forall\,\psi \in V,\, a(\varphi,\psi) = L(\psi) \tag{5}$$

Solving (5) is equivalent to solving (1) and (2), with the advantage that it is simpler to compute numerical solutions of (5) (see [5]).

The left hand side of equation (4) is composed of two terms : a domain integral and a boundary integral. Our method relies on two fundamental observations :

- the boundary integral in (4) accounts for the boundary conditions. Hence, when (4) is satisfied for every test function, both (1) and (2) are automatically satisfied. This is interesting because we solve the two equations by manipulating a single expression.

- thanks to the present formulation we do not need to tailor the test functions to the boundary. The choice of the tests functions is flexible, and there is no constraint on their value on the boundary. In other words, *the basis functions do not depend on the geometry of* Ω.

2.2. B-Spline discretization

Our goal is to discretize equation (4) by projecting it onto a B-spline basis in order to compute an approximate solution of the diffusion equation. We consider the following basis functions

$$\psi^h_{\mathbf{k}}(\mathbf{x}) = \beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right),\quad \mathbf{k}\in\mathbb{Z}^2$$

where $\beta^1(x,y) = \mathrm{tri}(x)\,\mathrm{tri}(y)$, $\mathrm{tri}(t)$ being the hat function. These functions are bivariate B-splines of degree 1 defined on a regular cartesian grid of pitch h.

We look for an approximate solution of (4) in the space spanned by the B-splines. Let us expand φ on the spline basis :

$$\varphi\,(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} c\,[\mathbf{k}]\;\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right)$$

This yields

$$a(\varphi,\,\psi) = \sum_{\mathbf{k}\in\mathbb{Z}^2} c\,[\mathbf{k}]\;a\left(\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right),\,\psi\right)$$

For $\mathbf{n} \in \mathbb{Z}^2$, we denote $S_{\mathbf{n}}$ the grid cell (a square) whose lower left corner has coordinates $h\,\mathbf{n}$. We assume that the optical coefficients have been discretized on the grid : $D[\mathbf{n}]$ (resp. $\mu_a[\mathbf{n}]$) is the value of D (resp. μ_a) in $S_{\mathbf{n}}$. By decomposing the integrals over the cells of the grid, we can go further in the discretization of $a(.,.)$:

$$\begin{aligned}
&a\,(\varphi,\,\psi) = \\
&\textstyle\sum_{\mathbf{k}\in\mathbb{Z}^2} c[\mathbf{k}] \sum_{\mathbf{n}\in\mathbb{Z}^2} \Big\{ D[\mathbf{n}] \int_{S_{\mathbf{n}}\cap\Omega} \nabla\beta^1\left(\tfrac{\mathbf{x}}{h} - \mathbf{k}\right)\,\nabla\psi\,\mathrm{d}\mathbf{x} \\
&+\mu_a[\mathbf{n}] \textstyle\int_{S_{\mathbf{n}}\cap\Omega} \beta^1\left(\tfrac{\mathbf{x}}{h} - \mathbf{k}\right)\,\psi\,\mathrm{d}\mathbf{x} \\
&+ \tfrac{1}{2}\textstyle\int_{S_{\mathbf{n}}\cap\partial\Omega} \beta^1\left(\tfrac{\mathbf{x}}{h} - \mathbf{k}\right)\,\psi\,\mathrm{d}\sigma \Big\}
\end{aligned}$$

Notice that in this expression, the support of every basis function spans only four grid cells.

The source term is also discretized using the spline basis :

$$q(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} q\,[\mathbf{k}]\;\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right)$$

This operation amounts to interpolating q using β^1 splines. With this expression of q, $L(\psi)$ becomes :

$$L(\psi) = \sum_{\mathbf{k}\in\mathbb{Z}^2} q\,[\mathbf{k}] \int_\Omega \beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right)\,\psi(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

Now, we use the B-splines as our test functions, which yields the following set of equations :

$$\sum_{\mathbf{k}\in\mathbb{Z}^2} c\,[\mathbf{k}]\;a\left(\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right),\,\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{l}\right)\right) = L\left(\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{l}\right)\right),\quad \forall\mathbf{l}\in\mathbb{Z}^2$$

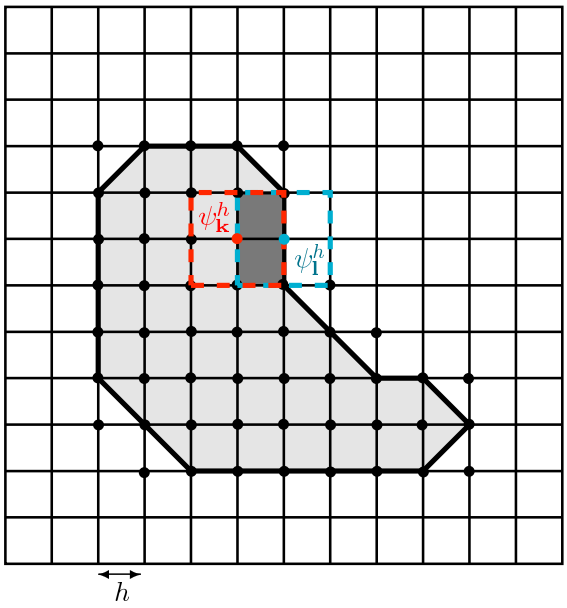

Fig. 1. Example of a cartesian grid with a discretized domain Ω (light grey) and its discretized boundary (thick black segments). In blue and red, the support of two basis functions. In dark grey, the intersection of their supports with Ω (used for the integral computation). The black dots label the bivariate β^1 splines appearing in the system matrix.

Given the expressions for $a(.,.)$ and L, the only non-zero terms in the above equality are those involving functions whose support intersects Ω. Since Ω is bounded, there is only a finite number of such functions (we use I to denote their indices). Fig 1 illustrates the discretization process. On this figure the black dots label elements of hI. We also represented the support of one basis function.

In the end, we are left with a linear system of equations :

$$\mathbf{A}\,\mathbf{c} = \mathbf{b}, \tag{6}$$

where we noted

$$\mathbf{A}_{\mathbf{k},\mathbf{l}} = a\left(\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right), \beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{l}\right)\right), \quad (\mathbf{k},\mathbf{l}) \in I \times I,$$

and $\mathbf{c}$ is the vector of coefficients $c[\mathbf{k}]$, $\mathbf{k} \in I$. The vector $\mathbf{b}$ comes from the discretization of L :

$$\mathbf{b} = \mathbf{L}\,\mathbf{q}, \text{ where } \mathbf{q} = (q[\mathbf{k}])_{\mathbf{k}\in I}, \text{ and}$$

$$\mathbf{L}_{\mathbf{k},\mathbf{l}} = \int_\Omega \beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right)\,\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{l}\right)\,\mathrm{d}\mathbf{x}, \quad (\mathbf{k},\mathbf{l}) \in I \times I.$$

$\mathbf{A}$ is the system matrix of our forward model. It is called the rigidity matrix, and it is symmetric, positive definite. In order to compute an approximate solution of (5) we solve the linear system (6). Now, recall our remarks of Section 2.1. First, although our basis functions are defined on a cartesian grid, they are still adapted to solve the diffusion equation for an arbitrary geometry. Second, the boundary conditions are implicitly included in the system matrix. The accuracy of our approximation is controlled by h, the pitch of the grid.

At this point, we need a method to generate the rigidity matrix, and the vector $\mathbf{b}$. From the discretized expressions of $a(.,.)$ and L, we notice that the quantities of interest to generate $\mathbf{A}$ and $\mathbf{b}$ are the integrals over the cells of the grid.

2.3. The algorithm

1) Domain and boundary discretization

To compute the domain and contour integrals involved in the expressions of $a(.,.)$ and L, it is necessary to have a representation of the boundary of Ω. This is essential for specifying the limits of the domain of integration, and also to perform the contour integral.

To that end, we consider the cartesian mesh in which the basis functions are defined, and we use an approximation of the domain boundary that is composed of line segments. We only allow line segments that link a grid node to one of its 8 neighbors. In other words, we use edges and diagonals of the grid squares to approximate the boundary. The quality of the approximation obviously dependent on the step size h.

Additionally, at this stage we determine the set I of grid nodes involved in the computation of $\mathbf{A}$ (those for which the basis function intersects Ω).

2) Evalutation of the domain integrals :
$\int_{S_\mathbf{n}\cap\Omega} \nabla\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right)\,\nabla\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{l}\right)\,\mathrm{d}\mathbf{x}$ *and*
$\int_{S_\mathbf{n}\cap\Omega} \beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right)\,\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{l}\right)\,\mathrm{d}\mathbf{x}$

To automate the computations we exploit the properties of the grid, and the polygonal boundary geometry. Thanks to the previous step, there are only two ways for the boundary to cross a grid cell (the diagonals). Taking into account the relative positions of the basis functions, we come up with 144 cases in total for the value of the integral over a cell. This number can be reduced by symmetry considerations. The geometrical situation is illustrated Fig 1. Using formulas such as $(\beta^1 * \beta^1)(k) = \beta^3(k)$, we have closed-form expressions in all possible cases. Since there is a finite number of cases, we can build a table associating a geometrical configuration to the value of the domain integral.

3) Evalutation of the contour integrals :
$\int_{S_\mathbf{n}\cap\partial\Omega} \beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{k}\right)\,\beta^1\left(\frac{\mathbf{x}}{h} - \mathbf{l}\right)\,\mathrm{d}\mathbf{x}$

We proceed as in *2)* for evaluating the contour integrals. There is a finite number of cases, which enables us to build another table. This, in turn, takes care of the boundary conditions.

4) Generation of the rigidity matrix

Using our tables of pre-computed integral values, we can straightforwardly generate the system matrix in an automated way. The contribution of the optical parameters (D and μ_a) is incorporated as shown in the discretized expression of $a(.,.)$ (see Section 2.2). We consider that the coefficients have been discretized on the grid, and we weigh the integral on a cell by the value of a parameter on that same cell.

5) Generation of the vector $\mathbf{b}$

The same approach is used to compute $\mathbf{b}$: we generate the matrix $\mathbf{L}$ from the pre-computed tables, and we deduce $\mathbf{b}$ from the coefficients $(q[\mathrm{k}])_{\mathbf{k}\in\mathbb{Z}^2}$.

3. RESULTS

We have implemented a forward model based on the strategy described in Section 2.3. We tested the method on various configurations of the geometry, source number and positions. The results were successfully validated against results obtained from a Fourier domain analysis on the one hand, and from a more standard finite differences discretization on the other hand. With the Fourier approach we could test the model in the case of an infinite medium with zero boundary conditions at infinity, while finite differences enabled to test on a bounded domain with boundary conditions (2).

In Fig 2, we present the B-spline solution of the forward model for an oval geometry, with two light sources. This result was computed for a sample size of 20mm, using constant diffusion and absorption coefficients ($D = 2.73$ cm^{-1} , $\mu_a = 1\mathrm{cm}^{-1}$), and a 64×64 grid. Computation time is around 1min in that example, which is encouraging considering that the algorithm has not yet been optimized to solve large sparse linear systems ($\mathbf{A}$ is a 4096×4096 matrix, and we used the standard routine of Matlab on a Mac workstation equipped with a 2.66GHz dual core processor and 1GB of memory).

4. DISCUSSION

The modeling part is crucial in the ODT problem. Hence, an appropriate forward model needs to be built before one can even start to address the inverse problem. It is important to have a versatile enough model that is able to incorporate information provided by other imaging modalities. In practice, ODT can be coupled to a Computed Tomography (CT) imaging device, which acquires the geometry of the sample, and some information about its internal structure.

Although our approach is probably not as sophisticated as advanced mesh-based FEM in terms of adaptation to the geometry, we believe that it provides a good tradeoff between computation time, numerical accuracy and simplicity. Therefore, it is a good candidate for the forward model used in solving an inverse problem. Using the weak formulation, we have

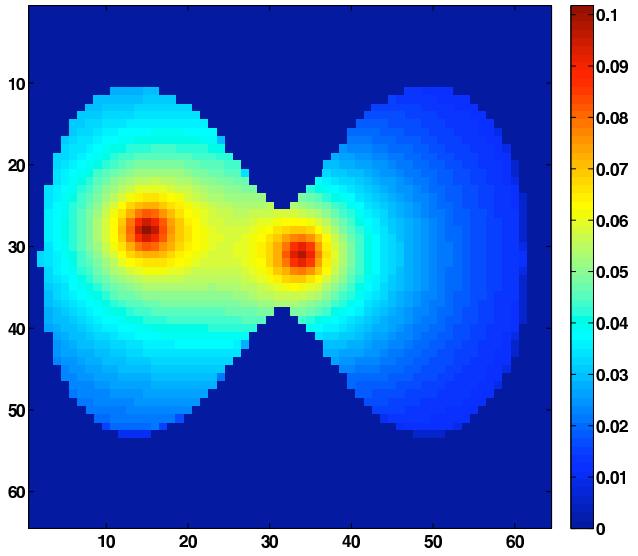

Fig. 2. Forward model solved for 2 point sources placed in an oval domain.

a very general model (D and μ_a can vary spatially, the geometry is arbitrary, boundary conditions are included), but we are not limited by meshing problems because we work with a cartesian grid. On top of providing acceptable accuracy ($O(h^2)$), the choice of β^1 splines as the basis functions keeps the computations simple, and yields sparse and structured rigidity matrices because they have a small support and a single degree of freedom. Finally, considering that the forward model will be used in a reconstruction algorithm, splines are interesting because they naturally span multi-resolution spaces. This is a significant advantage if one is interested in applying wavelet techniques, or multigrid approaches for reconstruction.

5. REFERENCES

[1] S R Arridge, "Optical tomography in medical imaging," *Inverse Problems*, vol. 15, no. 2, 1999.

[2] S R Arridge, "A finite element approach for modeling photon transport in tissue," *Medical Physics*, vol. 20, no. 2, pp. 299–309, Mar/Apr 1993.

[3] Ishimaru A., *Wave Propagation and Scattering in Random Media*, Academic Press, New York, 1078.

[4] Wenxiang Cong, Ge Wang, Durairaj Kumar, Yi Liu, Ming Jiang, Lihong Wang, Eric Hoffman, Geoffrey McLennan, Paul McCray, Joseph Zabner, and Alexander Cong, "Practical reconstruction method for bioluminescence tomography," *Optics Express*, vol. 13, no. 18, pp. 6756–6771, 2005.

[5] Grégoire Allaire, *Analyse numérique et optimisation*, Éditions de l'École Polytechnique, 2005.