INTEGRATED MODELING AND RECONSTRUCTION WITH SPARSITY CONSTRAINTS FOR FDOT

Jean-Charles Baritaux, Matthieu Guerquin-Kern, Michael Unser

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

ABSTRACT

We present a numerical framework for Fluorescence Diffuse Optical Tomography (fDOT) that combines a forward model together with an iterative reconstruction procedure. Using rapid linear solvers, we derived an efficient reconstruction strategy for quadratic regularizers. The method outperforms traditional reconstruction approaches. Starting from quadradic regularization, we then extend the framework to more general L_p constraints. We present reconstruction experiments that confirm the superiority of non-quadratic sparsity promoting regularization.

Index Terms— fDOT, finite elements, conjugate gradient, sparsity

1. INTRODUCTION

In fDOT near-infrared light is used to probe tissue for fluorescent markers. The goal is to recover the distribution of the markers. In the range of wavelength employed, scattering is the dominating phenomenon, and light propagation in tissue can be modeled by a diffusion process [1], [2]. Since we are dealing with fluorescence, there are two light fields involved : an excitation light field, denoted by v, and an emission light field denoted by u. Both are affected by scattering, and therefore satisfy a diffusion equation. The physical quantities of interest u and v are fluence rates, which correspond to the average amount of light energy locally flowing through an infinitesimal sphere. Mathematically this light propagation model translates into the following system of equations satisfied by u and v :

$$\begin{cases} -\nabla\cdot(D\nabla v)+\mu_a\,v=0 & \text{in }\Omega\\ v+a\,\boldsymbol{n}\cdot\nabla v=v^{ex} & \text{on }\partial\Omega_{ex}\\ v+a\,\boldsymbol{n}\cdot\nabla v=0 & \text{on }\partial\Omega\backslash\partial\Omega_{ex}\end{cases} \tag{1}$$

$$\begin{cases} -\nabla\cdot(D\nabla u)+\mu_a\,u=c\cdot v & \text{in }\Omega\\ u+a\,\boldsymbol{n}\cdot\nabla u=0 & \text{on }\partial\Omega\end{cases} \tag{2}$$

with the corresponding notation :

Ω	domain,
$\partial\Omega$	boundary of the domain,
$\partial\Omega_{ex}$	subset of the boundary where the excitation laser is pointed,
v^{ex}	fluence of the excitation laser on the boundary,
D	diffusion coefficient,
μ_a	absorption coefficient,
c	fluorophore distribution (to be recovered).

To make practical measurements, an excitation laser is shone on the object, and fluorescence light is collected using a camera. Hence, we have access only to boundary measurements. The geometry in this work is trans-illumination, meaning that the laser lights one side of the object, while the camera records the opposite side. Additionally a sweep is performed around the studied object, resulting in several excitation and emission light fields. N_x denotes the number of measurements composing a full sweep. u^k, v^k and m^k, $k=1..N_x$ are respectively the light fields and measurement results associated to each measurement in the sweep.

In this paper we propose a new Finite Elements based reconstruction algorithm for fDOT. The main contributions are :

- the development and implementation of an integrated modeling/reconstruction framework.
- an implicit description of the system; instead of assembling and storing a system matrix as is common practice in the field, our reconstruction procedure performs only those forward model computations needed at every iteration. This, combined with the fact that the forward model is implemented as a sparse linear system of equations, yields a gain in performance, and scalability compared to a direct method.
- the use of sparsity constraints for reconstruction.

After presenting our reconstruction strategy in Section 2, this paper details the numerical method that we propose. Sections 3 and 4 are devoted to the discretization and minimization schemes employed for quadratic regularizors, whereas Section 5 treats the case of sparsity constraints. In Section 6 we present results obtained with quadratic and "sparse" regularization. Finally, in Section 7, we discuss the computational cost of our strategy compared to a direct methods that use a pre-computed system matrix.

2. RECONSTRUCTION

In order to recover the concentration of fluorophore c, we employ a variational approach. Our estimate of the fluorophore concentration is found by minimizing a criterion. We consider the following families of functionals parametrized by $p\in[1,+\infty[$:

$$I_p(c)=\frac{1}{2}\sum_{k=1}^{N_x}\int_{\partial\Omega_k}(u^k(x)-m^k(x))^2\mathrm{d}\sigma(x)+\frac{\lambda}{2}\int_{\Omega}|c(x)|^p\mathrm{d}x$$

$$J_p(c)=\frac{1}{2}\sum_{k=1}^{N_x}\int_{\partial\Omega_k}(u^k(x)-m^k(x))^2\mathrm{d}\sigma(x)+\frac{\lambda}{2}\int_{\Omega}\|\nabla c(x)\|_2^p\mathrm{d}x$$

where $\sigma(x)$ is the surface measure on the boundary. They differ in the regularization criterion used. We use one or the other depending on the type of feature we want to promote in the estimated solution. Of particular interest are the two cases $p=2$ and $p=1$. For $p=2$ the functionals are quadratic; thus the minimization process is linear, which is interesting from the computational point of view. For $p=1$ we obtain so-called "sparsity" constraints after discretization. Note that $J_1(c)$ corresponds to TV-regularization. In that case the

This work was funded in part by the Swiss Foundation for Technology and Innovation.

minimization process is not linear, but we can use a sequence of quadratic functionals to approach the minimum, as we will see in Section 5. In the next section we explain the discretization process.

3. DISCRETIZATION

So far we have described the problem in a continuous framework. However, for computational purposes, we need to discretize both the propagation equations, and the functionals. To that end, we will use the Finite Elements Method (FEM) [3]. We start by defining a mesh of the physical domain which consists of N nodes. The fluence rates $u^k, k = 1..N_x$ and fluorophore concentration c are discretized on that mesh, although we could use distinct meshes for c (reconstruction) and the u^ks (forward model). Following the FEM, the functions u^k, m^k, $k = 1..N_x$ and c are approximated using polynomial interpolation by the functions u_h^k, m_h^k, $k = 1..N_x$ and c_h defined as :

$$\begin{aligned}
u_h^k(x) &= \sum_{j=1}^{N} u_{k,j}\phi_j(x) \approx u^k(x), \quad k = 1..N_x \\
m_h^k(x) &= \sum_{j=1}^{N} m_{k,j}\phi_j(x) \approx m^k(x), \quad k = 1..N_x \\
c_h(x) &= \sum_{j=1}^{N} c_j\phi_j(x) \approx c(x)
\end{aligned}$$

In these expressions, the ϕ_js are our interpolation functions. For this work we used linear interpolation on a triangular mesh (pyramid-like functions) so that the number of interpolation functions is N, the same as the number of nodes in the mesh. Thus, the quantities of interest u_h^k, m_h^k, $k = 1..N_x$ and c_h are uniquely specified by the vectors $\mathbf{u}_k = (u_{k,1}, \ldots, u_{k,N})^T$ and $\mathbf{m}_k = (m_{k,1}, \ldots, m_{k,N})^T$, $k = 1..N_x$, and $\mathbf{c} = (c_1, \ldots, c_N)^T$ respectively.

Now, we define the matrices that enter the computations :

$$\begin{aligned}
\mathbf{L}_{i,j} &= \int_\Omega \nabla\phi_i(x)\cdot\nabla\phi_j(x) + \kappa\,\phi_i(x)\phi_j(x)\mathrm{d}x \\
&\qquad + \frac{1}{a}\int_{\partial\Omega}\phi_i(x)\phi_j(x)\mathrm{d}\sigma, \\
\mathbf{S}_{i,j} &= \int_\Omega \phi_i(x)\phi_j(x)\mathrm{d}x, \\
(\mathbf{M}_k)_{i,j} &= \int_{\partial\Omega_k}\phi_i(x)\phi_j(x)\mathrm{d}\sigma, \\
(\mathbf{S}_k)_{i,j} &= \int_\Omega \phi_i(x)v^k(x)\phi_j(x)\mathrm{d}x, \\
\mathbf{G}_{i,j} &= \int_\Omega \nabla\phi_i(x)\cdot\nabla\phi_j(x)\mathrm{d}x, \\
\text{where} \quad & i = 1..N,\ j = 1..N,\ k = 1..N_x.
\end{aligned}$$

3.1. Propagation equations

The Finite Elements Method ensures that we get a numerical approximation of the solution of equation (2) by solving the linear system $\mathbf{L}\mathbf{u}_k = \mathbf{S}_k\mathbf{c}$ for $\mathbf{u}_k$. Here we made the assumption that the object is homogeneous and we defined $\kappa = \mu_a/D$. Besides, we assumed that the excitation light fields $v^k, k = 1..N_x$ are available. Indeed, these can be aptly approximated using the FEM as well.

3.2. Quadratic functionals

Let us proceed with the discretization of the quadratic functionals $I_2(c)$ and $J_2(c)$. We will do it explicitly for $I_2(c)$. By replacing c and u in the functional with their approximations c_h and u_h we get :

$$\begin{aligned}
I_2(c_h) &= \frac{1}{2}\sum_{k=1}^{N_x}\int_{\partial\Omega_k}(u_h^k(x) - m_h^k(x))^2\mathrm{d}\sigma(x) + \frac{\lambda}{2}\int_\Omega(c_h(x))^2\mathrm{d}x \\
&= \frac{1}{2}\sum_{k=1}^{N_x}\left\{\mathbf{u}_k^T\mathbf{M}_k\mathbf{u}_k - 2\mathbf{u}_k^T\mathbf{M}_k\mathbf{m}_k + \mathbf{m}_k^T\mathbf{M}_k\mathbf{m}_k\right\} + \frac{\lambda}{2}\mathbf{c}^T\mathbf{S}\mathbf{c}
\end{aligned}$$

where we made use of the matrices defined above. By using the fact that we have $\mathbf{L}\mathbf{u}_k = \mathbf{S}_k\mathbf{c}$ we obtain :

$$\begin{aligned}
I_2(\mathbf{c}) &= \frac{1}{2}\sum_{k=1}^{N_x}\left\{\mathbf{c}^T\mathbf{S}_k^T\mathbf{L}^{-T}\mathbf{M}_k\mathbf{L}^{-1}\mathbf{S}_k\mathbf{c} - 2\mathbf{c}^T\mathbf{S}_k^T\mathbf{L}^{-T}\mathbf{M}_k\mathbf{m}_k\right\} \\
&\qquad + \frac{\lambda}{2}\mathbf{c}^T\mathbf{S}\mathbf{c} + \text{cste} \\
&= \frac{1}{2}\mathbf{c}^T\left(\sum_{k=1}^{N_x}\mathbf{S}_k^T\mathbf{L}^{-T}\mathbf{M}_k\mathbf{L}^{-1}\mathbf{S}_k + \lambda\mathbf{S}\right)\mathbf{c} \\
&\qquad - \left(\sum_{k=1}^{N_x}\mathbf{S}_k\mathbf{L}^{-1}\mathbf{M}_k\mathbf{m}_k\right)^T\mathbf{c} \quad (3)
\end{aligned}$$

Finally we define $\mathbf{A}_1 = \sum_{k=1}^{N_x}\mathbf{S}_k^T\mathbf{L}^{-T}\mathbf{M}_k\mathbf{L}^{-1}\mathbf{S}_k + \lambda\mathbf{S}$ and $\mathbf{b} = \sum_{k=1}^{N_x}\mathbf{S}_k\mathbf{L}^{-1}\mathbf{M}_k\mathbf{m}_k$, and rewrite equation (3) in the lighter form $I_2(\mathbf{c}) = \frac{1}{2}\mathbf{c}^T\mathbf{A}_1\mathbf{c} - \mathbf{b}^T\mathbf{c}$. A similar derivation yields $J_2(\mathbf{c}) = \frac{1}{2}\mathbf{c}^T\mathbf{A}_2\mathbf{c} - \mathbf{b}^T\mathbf{c}$, with $\mathbf{A}_2 = \sum_{k=1}^{N_x}\mathbf{S}_k^T\mathbf{L}^{-T}\mathbf{M}_k\mathbf{L}^{-1}\mathbf{S}_k + \lambda\mathbf{G}$. Next, we propose a minimization scheme for these quadratic functionals.

4. CONJUGATE GRADIENT ALGORITHM

It is important to notice that the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are symmetric definite positive. As a consequence, we can apply the conjugate gradient algorithm (CG) to perform the minimization. To be specific, we will write the algorithm applied to the minimization of $J_2(\mathbf{c}) = \frac{1}{2}\mathbf{c}^T\mathbf{A}_2\mathbf{c} - \mathbf{b}^T\mathbf{c}$ for instance. $\mathbf{c}_0$ given (initial guess)

```
Compute $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}_2\mathbf{c}_0$, $\mathbf{w}_1 = \mathbf{r}_0$ and $\alpha_1 = \frac{\mathbf{r}_0^T\mathbf{w}_1}{\mathbf{w}_1^T\mathbf{A}_2\mathbf{w}_1}$
For n=1,2,3,...
    $\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n\mathbf{A}_2\mathbf{w}_n$,    if $\frac{\|\mathbf{r}_n\|}{\|\mathbf{b}\|}$ <tol, stop.
    $\beta_n = -\frac{\mathbf{r}_n^T\mathbf{A}_2\mathbf{w}_n}{\mathbf{w}_n^T\mathbf{A}_2\mathbf{w}_n}$
    $\mathbf{w}_{n+1} = \mathbf{r}_n + \beta_n\mathbf{w}_n$
    $\alpha_{n+1} = \frac{\mathbf{r}_n^T\mathbf{w}_{n+1}}{\mathbf{w}_{n+1}^T\mathbf{A}_2\mathbf{w}_{n+1}}$
    $\mathbf{c}_{n+1} = \mathbf{c}_n + \alpha_{n+1}\mathbf{w}_{n+1}$
end
```

In this algorithm, we repeatedly have to perform matrix-vector products with the matrix $\mathbf{A}_2$. That is, given a vector $\mathbf{v}$, we have to evaluate :

$$\mathbf{A}_2\mathbf{v} = \sum_{k=1}^{N_x} \mathbf{S}_k^T \mathbf{L}^{-T} \mathbf{M}_k \mathbf{L}^{-1} \mathbf{S}_k \mathbf{v} + \lambda \mathbf{G}\mathbf{v}$$

This expression involves multiplications by $\mathbf{L}^{-1}$. A feature of our algorithm is that we do not evaluate these products explicitly. Instead, we compute $\mathbf{x} = \mathbf{L}^{-1}\mathbf{y}$ by solving the linear system $\mathbf{L}\mathbf{x} = \mathbf{y}$ also using CG. Because $\mathbf{L}$ is a sparse symmetric definite positive matrix, we can show that this algorithm is faster than storing $\mathbf{L}^{-1}$ (which is a full matrix) and computing $\mathbf{L}^{-1}\mathbf{y}$ explicitly.

5. SPARSITY

We now tackle the case $p = 1$. We focus on the functional $J_1(c)$, but a similar reasoning can be made on $I_1(c)$, as well as more generally on $I_p(c)$ and $J_p(c)$. First, we approximate the regularization term in $J_1(c)$ using Gauss quadrature :

$$\int_\Omega \|\nabla c(x)\|_2 \, \mathrm{d}x \approx \sum_{e \in E} \omega_e \|\nabla c(x_e)\|_2$$

where E is the set of elements, x_e, $e \in E$ are fixed quadrature points and ω_e, $e \in E$ are quadrature weights. Second, we write $\nabla c_h(x) = \mathbf{D}(x)\mathbf{c}$, with

$$\mathbf{D}(x) = \begin{pmatrix} \partial_{x_1}\phi_1(x) & \cdots & \partial_{x_1}\phi_N(x) \\ \partial_{x_2}\phi_1(x) & \cdots & \partial_{x_2}\phi_N(x) \end{pmatrix}$$

Combining both, $J_1(c)$ is discretized as follows :

$$J_1(\mathbf{c}) = \frac{1}{2}\sum_{k=1}^{N_x} \left\|\mathbf{L}^{-1}\mathbf{S}_k\mathbf{c} - \mathbf{m}_k\right\|_{\mathbf{M}_k}^2 + \frac{\lambda}{2}\sum_{e \in E} \omega_e \|\mathbf{D}(x_e)\mathbf{c}\|_2$$

Among all the possible ways to evaluate the minimum of $J_1(\mathbf{c})$ we chose the iterative reweighted least-squares algorithm [4]. It is an empirical algorithm, and the intuitive idea behind it is simply to rewrite

$$\sum_{e \in E} \omega_e \|\mathbf{D}(x_e)\mathbf{c}\|_2 = \sum_{e \in E} \omega_e \|\mathbf{D}(x_e)\mathbf{c}\|_2^{-1} \|\mathbf{D}(x_e)\mathbf{c}\|_2^2$$

Thus, if the n-th iterate $\mathbf{c}^{(n)}$ lies close to the minimum, we have :

$$\sum_{e \in E} \omega_e \left\|\mathbf{D}(x_e)\mathbf{c}^{(n)}\right\|_2 \approx \sum_{e \in E} \omega_e \left\|\mathbf{D}(x_e)\mathbf{c}^{(n-1)}\right\|_2^{-1} \left\|\mathbf{D}(x_e)\mathbf{c}^{(n)}\right\|_2^2$$

Thanks to this weighting procedure we are back with a quadratic functional to minimize, which is readily done using the tools outlined in the previous section. Note that the number of quadrature points per element must be adapted to the interpolation scheme used (in our case, one point per element of the triangular mesh).

6. RESULTS

The numerical framework presented above has been implemented and tested in 2D. Here, we present results obtained for a homogeneous disk geometry ($\mu_a = 0.035\text{cm}^{-1}$ and $\mu_s = 1.55\text{cm}^{-1}$). Measurements were generated by simulation, and poisson noise was added. Simulated intensities were adjusted to match the intensities of real measurements so as to obtain a realistic noise level of the poisson noise. The diffusion and absorption constants were assumed to be known, but no other a priori knowledge was incorporated in the reconstruction algorithm. The measurement geometry is the one described in the introduction, i.e. trans-illumination. Recall that we only have access to measurements on one side of the boundary. The mesh used in these results is a triangulation composed of 435 elements. Figure 1 presents results for two extended sources. The results obtained with TV (cases (d) and (f)) are more accurate than with quadratic regularization (cases (c) and (e)). When the regularization parameter is set to a high value, the quadratic algorithm tends to locate the sources close to the boundary of the domain. Another problem of the quadratic regularization is that the source intensities are underestimated. These issues are overcome by the sparsity constraints. We also notice that the ”sparse” algorithm is less sensitive to the regularization parameter. Although (c) is a worse initial guess that (e), the TV algorithm is still able to recover position and intensity of the sources. Also note that pure sparsity (case (b)) performs as well as TV in that particular case.

7. COMPUTATIONAL COST

Here we compare the cost of the proposed linear method (for quadratic regularizor) to the cost of a direct method. We consider a 2D problem with n discretization cells per direction, resulting in a total of n^2 unknowns. Since we have a problem of size n^2, CG needs at most n^2 iterations to minimize the functional (remember that the size of the implicit system matrix is n^2). At every iteration, CG has to evaluate a matrix-vector multiplication which requires to solve two linear systems involving the matrix $\mathbf{L}$ as we saw in Section 4. The matrix $\mathbf{L}$ derives from the discretization of the PDE (2); it is possible to prove that CG solves the corresponding linear system in $O(n^3)$ for a 2D problem with n^2 unknowns. Overall, the cost of the algorithm is $O(n^2)$ iterations each with cost $O(n^3)$, which yields $O(n^5)$. If we express the cost in terms of the total number of unknowns $N = n^2$ we obtain $O(N^{5/2})$. On the other hand, the cost of a direct method (commonly used) is $O(N^3)$, so we already see a gain. For real life 3D problems the gain is even more important. The same analysis yields $O(N^{7/3})$ for the proposed method compared to $O(N^3)$ for a direct method.

8. CONCLUSION

We have demonstrated an efficient FEM strategy for fDOT that can adapt to a wide spectrum of regularization policies. Due to its reduced computational demands, it is potentially applicable to larger size problems than what can be afforded using conventional direct algorithms. This is especially valuable for 3D applications. Future work include the validation of that technique using actual phantom data, and the implementation of a full 3D system.

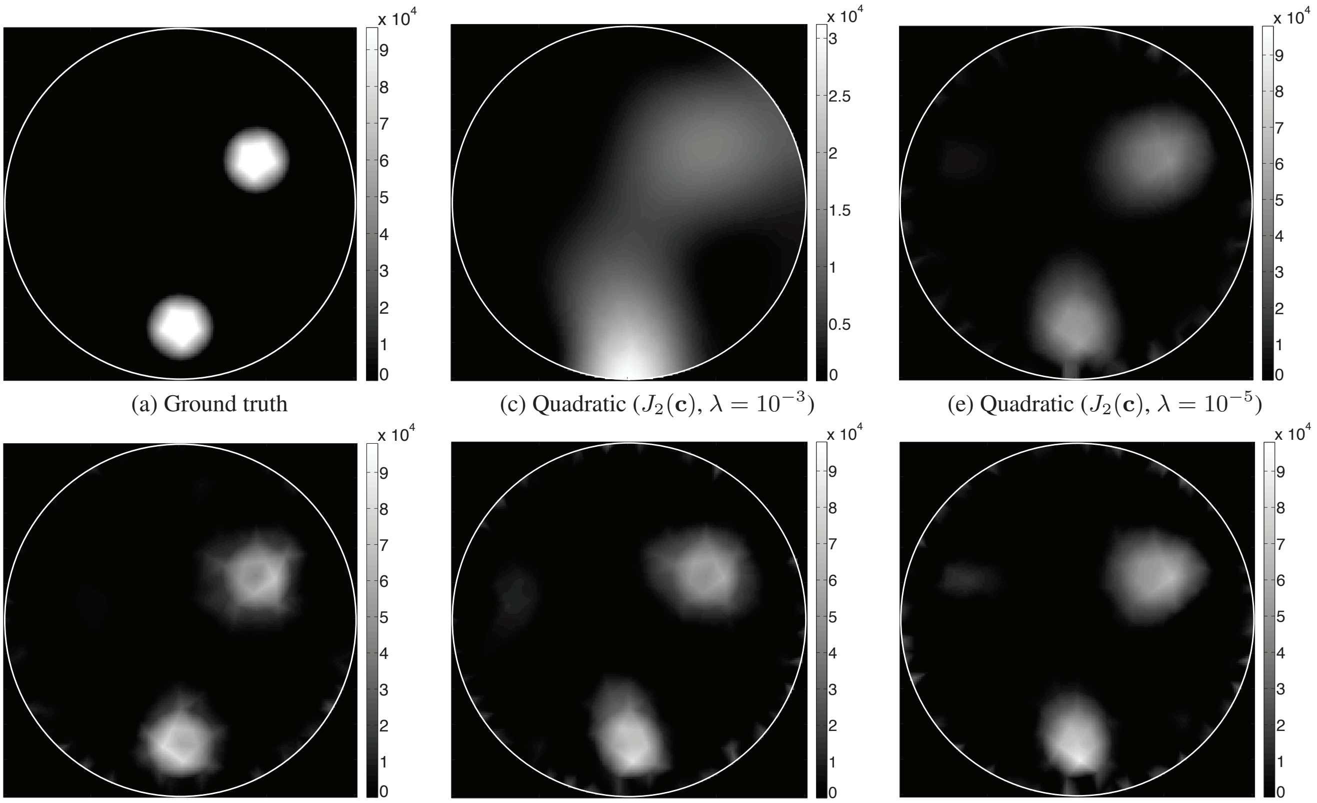
9. REFERENCES

[1] A Ishimaru, *Wave propagation and Scattering in Random Media*, New York: Academic, 1978.

[2] S R Arridge, “Optical tomography in medical imaging,” *Inverse Problems*, vol. 15, no. 2, pp. R41–R93, 1999.

[3] P G Ciarlet and J L Lions, *Handbook of numerical analysis*, Elsevier Science, 1990.

[4] T K Moon and W C Stirling, *Mathematical methods and algorithms for signal processing*, Prentice Hall, 2000.



(a) Ground truth

(c) Quadratic ($J_2(\mathbf{c})$, $\lambda = 10^{-3}$)

(e) Quadratic ($J_2(\mathbf{c})$, $\lambda = 10^{-5}$)

(b) Sparse ($I_1(\mathbf{c})$, $\lambda = 10^{-5}$)

(d) TV ($J_1(\mathbf{c})$, $\lambda = 10^{-3}$)

(f) TV ($J_1(\mathbf{c})$, $\lambda = 10^{-5}$)

Fig. 1. Reconstruction of two fluorescent sources