Accelerated High-Resolution EEG Source Imaging*

Jing Qin[1], Tianyu Wu[2], Ying Li[3], Wotao Yin[4], Stanley Osher[5], and Wentai Liu[6]

***Abstract*— Electroencephalography (EEG) signal has been playing a crucial role in clinical diagnosis and treatment of neurological diseases. However, it is very challenging to efficiently reconstruct the high-resolution brain image from very few scalp EEG measurements due to high ill-posedness. Recently some efforts have been devoted to developing EEG source reconstruction methods using various forms of regularization, including the ℓ_1-norm, the total variation (TV), as well as the fractional-order TV. However, since high-dimensional data are very large, these methods are difficult to implement. In this paper, we propose accelerated methods for EEG source imaging based on the TV regularization and its variants. Since the gradient/fractional-order gradient operators have coordinate friendly structures, we apply the Chambolle-Pock and ARock algorithms, along with diagonal preconditioning. In our algorithms, the coordinates of primal and dual variables are updated in an asynchronously parallel fashion. A variety of experiments show that the proposed algorithms have more rapid convergence than the state-of-the-art methods and have the potential to achieve the real-time temporal resolution.**

I. INTRODUCTION

Electroencephalography (EEG) has been playing a crucial role in detecting abnormal brain activities which are responsible for various neurological diseases, e.g., epilepsy and brain tumors. Different from Single Photon Emission Computed Tomography (SPECT), Positron Emission Tomography (PET) and functional Magnetic Resonance Imaging (fMRI), EEG can record fast neural activity over milliseconds. As long as scalp measurements are available, EEG source imaging estimates brain sources of high temporal resolution by solving the corresponding *inverse problem*. Under the assumption that the location and orientation of dipoles are fixed, the EEG inverse problem is converted to estimating amplitudes of distributed sources on the cortical surface. In this case, there are much more unknowns than the given measurements, so the problem is under-determined. To address the ill-posedness of the EEG inverse problem, a large number of methods based on various regularization techniques have been proposed [1], [2], [3]. Recently, variational-based approaches [4], [5], [6], [7], [8] have improved localization performance concerning accuracy and focalization error. However, computation efficiency is still a bottleneck and hinders many methods from being applied in practice [3], [9], [6]. It is of high necessity to develop fast, high-resolution source imaging methods which can not only localize sources but also estimate the spatial extent of the sources.

In this paper, we develop a novel framework for accelerating variational-based high-resolution EEG source imaging. The algorithm is motivated by our recent work but is significantly improved. Our recent reconstruction model [8] for EEG sources uses the regularization of graph fractional-order total variation (gFOTV). gFOTV has a great potential to achieve the state-of-the-art performance regarding accuracy, spatial resolution as well as localization accuracy. However, since each of its iterations solves a least squares subproblem, the computation intensity of gFOTV grows very quickly as the number of voxels increases. Inspired by the numerical performance of the asynchronous coordinate updating scheme [10], we exploit the coordinate-friendly structure of the gFOTV operator and develop a very fast algorithm to solve it. It is important to note that the proposed algorithm can be easily extended toward other regularization models that involve the composition of the ℓ_1-norm and a coordinate-friendly operator, such as those in the sparse source imaging (SSI) [11] and the spatial Laplacian in [9].

The rest of the paper is organized as follows. Section II introduces the EEG inverse problem and briefly reviews the inverse EEG imaging models regularized by total variation and its variants. Section III describes the proposed acceleration approach, as well as its numerical algorithm, in detail. The numerical performance obtained on synthetic data is reported in Section IV. A brief conclusion is drawn in Section V.

II. REGULARIZED EEG INVERSE PROBLEM

The EEG inverse problem is to reconstruct spatial distributions of currents in the form of a 3D brain image from a very limited number of electrode measurements. To simplify the discussion, we adopt the two widely used assumptions: 1) the dipole sources with fixed locations and directions are distributed on the cortex surface, which is discretized as a triangular mesh containing N voxels; 2) each dipole is normally aligned perpendicular to the cortex surface. Thus far, we only need to estimate the amplitudes of dipole sources. Assume that there are M electrodes placed on the scalp. Let $b \in \mathbb{R}^M$ be the electrical potential on the scalp measured by the electrodes, and $u \in \mathbb{R}^N$ be the neural current density at each dipole location. Then the potential b and the neural current density u satisfy the equation

$$b = Au + n$$

*This work is supported in part by the California Capital Equity LLC, the Keck foundation, NSF DMS-1317602, NSF ECCS-1462397 and ONR N000141612157.

[1]Jing Qin is with the Department of Mathematical Sciences, Montana State University, Bozeman, MT 59717, USA jing.qin@montana.edu

[2]Tianyu Wu, Wotao Yin and Stanley Osher are with the Department of Mathematics, University of California, Los Angeles, CA 90095, USA {wuty11, wotaoyin, sjo}@math.ucla.edu

[3]Ying Li and Wentai Liu are with the Department of Bioengineering, University of California, Los Angeles, CA 90095, USA {yingli, wentai}@ucla.edu

where $n \in \mathbb{R}^M$ is the noise. Here $A \in \mathbb{R}^{M\times N}$ is called the lead field matrix, whose (i,j)-th entry represents the electrical potential measured by the i-th electrode due to an unit dipole source at the j-th voxel. The matrix A is obtained by constructing a head model followed by solving the Maxwell's equations under quasi-static conditions. In practice, we have $M \ll N$ which implies there will be infinitely many solutions to the under-determined linear system $b = Au$. In order to obtain the unique solution, various regularization techniques have been applied to impose geometric characterizations on the solution. In this paper, we unify the total variation regularization that was used in VB-SCCD and SVB-SCCD [4], and one of its generalizations–the graph fractional-order TV [8]. We consider the following variational model to reconstruct the current density distribution on the cortical surface

$$\min_u \frac{1}{2}\|Au-b\|_2^2 + \lambda\|D_\alpha u\|_1, \tag{1}$$

where $\lambda > 0$ is a tuning parameter used to balance the first data fidelity term and the second regularization term. Here $\|\cdot\|_2$ is the Euclidean norm and $\|\mathbf{x}\|_1$ returns the sum of absolute values of all components in the vector $\mathbf{x}$. The fractional-order difference matrix D_α is defined as follows. Let $d(v_i,v_j)$ be the number of voxels on the shortest path connecting the voxels v_i and v_j, which is either in or close to a geodesic passing through v_i and v_j. Given a path $p = (v_{i=m_0}, v_{m_1}, \ldots, v_{m_K})$ where the shortest distance between the voxels v_{m_0} and v_{m_j} is j voxels, the fractional-order derivative along p is defined to be

$$(D_p^\alpha u)_i := D_p^\alpha u(v_i) = \sum_{v\in p} w_\alpha(d(v_0,v))u(v) = \sum_{j=0}^{K} w_\alpha(j)u(v_{m_j}).$$

Then the discretized fractional-order TV of u is defined as follows:

$$TV_\alpha(u) = \|D_\alpha u\|_1 = \sum_{i=1}^{N}\sum_{p\in\mathscr{P}(i;K)} |(D_p^\alpha u)_i|, \tag{2}$$

where $\mathscr{P}(i;K)$ is the set of all paths starting from the i-th voxel with length K voxels. Here D_α is a matrix of the size $L\times N$ with $L = 3N\cdot 2^{K-1}$. In particular, if $\alpha = 1$, then $K = 1$ and $D_\alpha u$ reduces to the finite difference approximation of gradient, which corresponds to the TV regularized model. As suggested in [8], it is sufficient to choose $K = 4$ in practice. Note that the proposed algorithms can be easily extended when we replace D_α in (1) by another sparsifying matrix. For example, if D_α is replaced by the identity matrix, then (1) becomes the ℓ_1-norm regularized model [3], [11].

III. FAST NUMERICAL ALGORITHMS

In our previous work [8], we applied the alternating direction methods of multipliers (ADMM) to derive an algorithm, where one subproblem involves solving $Mx = b$ for an $N\times N$ matrix M. It is time-consuming even though we used the Cholesky factorization; moreover, the computation time will increase superlinearly when the number of voxels N increases. Motivated by the desire to design a scalable algorithm, we resort to the Chambolle-Pock algorithm [12]. The Chambolle-Pock algorithm is one of the first primal-dual algorithms which solves the problem $\min_x\ f(x)+g(Ax)$ without inverting any matrices.

By defining

$$B = \begin{pmatrix} A \\ D_\alpha \end{pmatrix},\ f(p,q) = \frac{1}{2}\|p-b\|_2^2 + \lambda\|q\|_1 \text{for } \begin{pmatrix} p \\ q \end{pmatrix} \in \mathbb{R}^{M+L},$$

Problem (1) can be rewritten as $\min_u\ f(Bu)$. Then we can apply the Chambolle-Pock algorithm with diagonal preconditioning [13] to solve it. By introducing two dual variables $s \in \mathbb{R}^L, t \in \mathbb{R}^M$, we obtain the following algorithm

$$\begin{cases} u^{k+1} = u^k - \Sigma(D_\alpha^T s^k + A^T t^k) \\ s^{k+1} = \mathrm{Proj}_{\|\cdot\|_\infty\le\lambda}(s^k + \Gamma_1 D_\alpha(u^k - 2\Sigma(D_\alpha^T s^k + A^T t^k))) \\ t^{k+1} = (I+\Gamma_2)^{-1}(t^k - \Gamma_2 b + \Gamma_2 A^T(2u^{k+1} - u^k)). \end{cases} \tag{3}$$

Here $\Sigma, \Gamma_1, \Gamma_2$ are diagonal matrices controlling the step sizes, which are defined by

$$\begin{cases} \Sigma_{ii} = (\sum_{j=1}^{L} |(D_\alpha)_{ji}| + \sum_{j=1}^{M} |A_{ji}|)^{-1} \\ (\Gamma_1)_{ii} = (\sum_{j=1}^{N} |(D_\alpha)_{ij}|)^{-1} \\ (\Gamma_2)_{ii} = (\sum_{j=1}^{N} |A_{ij}|)^{-1}. \end{cases}$$

The projection of $\mathbf{x} \in \mathbb{R}^L$ is defined componentwise as

$$\left[\mathrm{Proj}_{\|\cdot\|_\infty\le\lambda}(\mathbf{x})\right]_i = \min(\lambda, \max(-\lambda, x_i)).$$

It has been shown that the above choices make (3) converge fast.

By letting $z^k := (u^k, s^k, t^k)^T$, (3) can be rewritten as

$$z^{k+1} = Tz^k.$$

As shown in [10], it is computationally advantageous to update z^k in the (block) coordinate fashion. Specifically, at each iteration, we randomly update one coordinate of z in (3), say $z_i^{k+1} = (Tz^k)_i$, and keep the rest coordinates unchanged, i.e., $z_j^{k+1} = z_j^k$ for $j \neq i$. Furthermore, as shown in [10, Section 4.2], we can plug the u update in (3) to the t update, and get a similar yet new algorithm:

$$\begin{cases} u^{k+1} = u^k - \Sigma(D_\alpha^T s^k + A^T t^k) \\ s^{k+1} = \mathrm{Proj}_{\|\cdot\|_\infty\le\lambda}(s^k + \Gamma_1 D_\alpha(u^k - 2\Sigma(D_\alpha^T s^k + A^T t^k))) \\ t^{k+1} = (I+\Gamma_2)^{-1}(t^k - \Gamma_2 b + \Gamma_2 A^T(u^k - 2\Sigma(D_\alpha^T s^k + A^T t^k))), \end{cases} \tag{4}$$

which is more suitable for coordinate update. It has been shown that (4) is *coordinate-friendly*, i.e., updating one coordinate of z is much cheaper than updating z entirely, and the aggregate cost of updating all coordinates is similar to updating the whole z. Moreover, one of the major advantages of coordinate update is that it allows larger step sizes than its full update counterpart. During the implementation of the coordinate update of (4), all the step-size matrices $\Sigma, \Gamma_1, \Gamma_2$ are multiplied by a scaling factor $s \ge 1$, which empirically leads to faster convergence.

Furthermore, to speed up convergence on computers and clusters equipped with multiple cores, we apply the method

in [10], [14] and get an asynchronous parallel update version of (4), which is described in Algorithm 1, With a properly

Algorithm 1: Async-parallel update of (4)

Input : $z^0 \in \mathbb{R}^{N+L+M}$, $K > 0$, $0 < \rho < 1$, $(q_1, \ldots, q_{N+L+M})$ where $q_i > 0$, $\sum_{i=1}^{N+L+M} q_i = 1$

Set the global iteration counter $k = 0$;

while $k < K$, *every agent asynchronously* **do**

- Select $1 \leq i_k \leq N+L+M$ with $\text{Prob}(i_k = i) = q_i$;
- Compute $\hat{z}_{i_k}^{k+1}$ according to (4);
- Update $z_{i_k}^{k+1} = (1-\rho) z_{i_k}^k + \rho \hat{z}_{i_k}^{k+1}$;
- Update the global counter $k \leftarrow k+1$;

chosen factor $\rho < 1$, Algorithm 1 will yield a solution to Problem (1). We empirically choose $\rho = 0.5$ in all experiments for stable performance.

IV. EXPERIMENTAL RESULTS

In this section, we demonstrate the performance of the proposed algorithms by testing them on various simulation data sets. In particular, We compare the randomized coordinate update version of (4) and its multi-core version with the ADMM-based algorithm [8] and the CVX toolbox `http://cvxr.com/cvx/`. The toolbox has been used in the state-of-the-art EEG methods [9], [6]. We apply the fractional-order TV regularized model (2) with $\alpha = 1.6$, which consistently yields superior performance compared to other related models in terms of accuracy [8]. The current parameter setting works universally for all our test data. It is worth noting that as the number of voxels increases, the quality of brain images reconstructed from this model can be further enhanced.

A. Simulation Protocol

Our simulation uses the same sources as those in [8], which we describe as follows. First, a center is seeded and then its neighbors are gradually recruited to form a patch. The current density is the strongest in the center, and then gradually decays as it goes far from the center following the shape of a Gaussian distribution. To represent sources at different locations, we randomly select three sources located in different lobes of the cortex surface. To simulate the noise, we impose random independent and identically distributed (i.i.d.) Gaussian noise to each voxel and then add Gaussian noise at a signal-to-noise ratio (SNR) of 20dB to each channel. The simulated measurements are normalized to fall between 10 μV to 100 μV, as suggested in [15]. Experimental results show that the computation time only relies on the number of voxels, the number of electrodes and the regularization parameter λ, rather than the configuration of noise and source. Here we choose the number of electrodes as 68, 103 and 346, and the number of voxels as 10240, 16384 and 40960.

m	n	CVX	ADMM	$p=1$	$p=16$
68	10240	52.78	67.36	40.30	4.90
68	16384	95.00	237.80	74.46	8.46
68	40960	311.71	2015.79	242.42	25.45
103	10240	62.37	64.21	33.00	3.84
103	16384	108.16	175.31	56.19	6.35
103	40960	393.23	1639.51	194.48	19.77
346	10240	370.63	51.02	45.14	5.21
346	16384	476.16	135.12	66.45	7.59
346	40960	2720.28	1200.10	177.15	19.70

TABLE I: Computation time in seconds. Columns 3-6 list the computation times of single-threaded CVX, ADMM and the algorithm (1) with one thread and 16 threads, respectively.

B. Computing Platform

All numerical experiments are performed in a machine with an Intel® Xeon® CPU E5-2650 v4 @ 2.2GHz and 64GB RAM in double precision. The CPU has 12 physical cores and each core has 2 logical processors. The ADMM and CVX are called in Matlab 2016a. The async-parallel coordinate update algorithm is written in C++ based on the toolbox [16]. We compare only the computing time of the three algorithms. The core optimization algorithm of CVX is written in C and called in Matlab through MEX; the ADMM subproblems are solved using basic linear algebra operations in Matlab, which are highly optimized. Therefore the comparison is considered fair despite the difference of platforms.

C. Performance Comparison

To make a fair comparison, we let each algorithm run until it achieves the same objective function value, which implies the same accuracy and focalization degree. The scaling factor s for the step size matrices in (1) is tuned between 5 and 11 to achieve the best performance. In addition, we fix the regularization parameter $\lambda = 20$ for all tests, as it works well consistently. Let p be the number of threads in our algorithm.

Fig. 1 shows all the brain images reconstructed from various data sets by randomized coordinate update of Algorithm (1) running one thread. As the triangular mesh gets finer, the reconstructed sources become more focused. The increase of electrodes also helps shrink the extent of sources and thereby localize sources more accurately. In Columns 3-6 of Table I, we compare the computation times of single-threaded CVX, ADMM and coordinate update of (1) running one thread and 16 threads. Here we manually change the precision in CVX to achieve the same accuracy. In ADMM, the algorithm terminates when it reaches either 1000 iterations or the tolerance of 10^{-3} for the relative error between two consecutive results. In our method, we tune the scaling factor s and the number of epoches for each data set, and then fix them for both single-threaded and multi-threaded versions of Algorithm (1). One can see that for large-scale data sets, e.g., the case with 346 electrodes and 40960 voxels, coordinate update algorithm (1) shows superior performance over ADMM and CVX in terms of computation time.

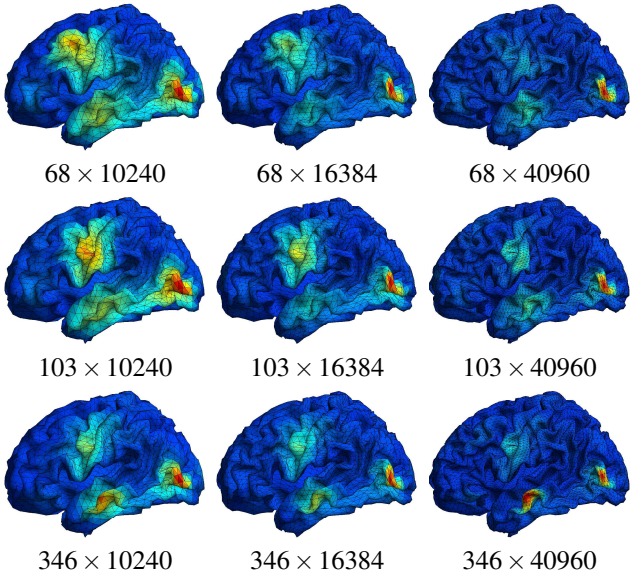

Fig. 1: Reconstructed brain images for various data sets (No. of electrodes × No. of voxels).

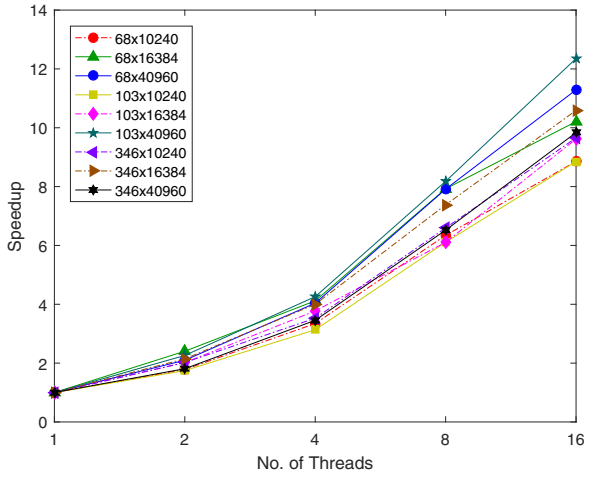

Fig. 2: Multi-threading speedup of computation.

Furthermore, to study the speedup behavior of Algorithm (1), we use various numbers of threads $p = 1, 2, 4, 8, 16$ and fix the number of epoches as 5000 and the scaling factor $s = 6$. We define the speedup ratio by

$$\frac{\text{running time using 1 thread}}{\text{running time using } p \text{ threads}},$$

which measures the reduction of running time due to the growth of threads. In Fig 2, we plot the speedup ratio against the number of threads. One can see that the async-parallel coordinate update algorithm achieves an (almost) linear speedup as the number of threads grows. The above comparisons show that multi-threading significantly shortens the computation time: for the 346×40960 test, it is reduced from over twenty minutes (ADMM or CVX) to less than twenty seconds (see Table I).

V. CONCLUSIONS

In this paper, we propose a fast and high-resolution EEG source imaging method, which significantly accelerates the numerical solutions of TV and gFOTV regularized EEG reconstruction methods. Specifically, by utilizing the coordinate friendly structure of the gradient and the fractional-order gradient operators, we derive the proposed algorithm by applying the primal-dual method and diagonal preconditioning technique. Numerical experiments show that the proposed method running multiple threads on a multi-core system exhibits superior performance in terms of both computation time and solution accuracy over the state-of-the-art methods. The proposed approach can be generalized to accelerate other regularized models involving the ℓ_1-norm. It also has a great potential to achieve real-time temporal resolution, which can potentially bring tremendous convenience and broad influence to clinical applications.

REFERENCES

[1] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa. Magnetoencephalography–theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics*, 65(2):413, 1993.

[2] R. D. Pascual-Marqui. Standardized low-resolution brain electromagnetic tomography (sLORETA): technical details. *Methods Find Exp Clin Pharmacol*, 24(Suppl D):5–12, 2002.

[3] K. Uutela, M. Hämäläinen, and E. Somersalo. Visualization of magnetoencephalographic data using minimum current estimates. *NeuroImage*, 10(2):173–180, 1999.

[4] L. Ding. Reconstructing cortical current density by exploring sparseness in the transform domain. *Physics in Medicine and Biology*, 54(9):2683, 2009.

[5] H. Becker, L. Albera, P. Comon, R. Gribonval, and I. Merlet. Fast, variation-based methods for the analysis of extended brain sources. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European*, pages 41–45. IEEE, 2014.

[6] A. Sohrabpour, Y. Lu, G. Worrell, and B. He. Imaging brain source extent from EEG/MEG by means of an iteratively reweighted edge sparsity minimization (IRES) strategy. *NeuroImage*, 2016.

[7] Ying Li, Jing Qin, Yue-Loong Hsin, Stanley Osher, and Wentai Liu. s-smooth: Sparsity and smoothness enhanced eeg brain tomography. *Frontiers in Neuroscience*, 10:543, 2016.

[8] Y. Li, J. Qin, S. Osher, and W. Liu. Graph Fractional-Order Total Variation EEG Source Reconstruction. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 101–104, Orlando, Florida, August 2016.

[9] W. Chang, A. Nummenmaa, J. Hsieh, and F. Lin. Spatially sparse source cluster modeling by compressive neuromagnetic tomography. *NeuroImage*, 53(1):146–160, 2010.

[10] Z. Peng, T. Wu, Y. Xu, M. Yan, and W. Yin. Coordinate friendly structures, algorithms and applications. *Annals of Mathematical Sciences and Applications*, 1:57–119, 2016.

[11] L. Ding and B. He. Sparse source imaging in electroencephalography with accurate field modeling. *Human brain mapping*, 29(9):1053–1067, 2008.

[12] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[13] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *2011 International Conference on Computer Vision*, pages 1762–1769. IEEE, 2011.

[14] Z. Peng, Y. Xu, M. Yan, and W. Yin. ARock: an algorithmic framework for asynchronous parallel coordinate updates. *SIAM Journal on Scientific Computing*, 38(5):A2851CA2879, 2016.

[15] H. Aurlien, I. O. Gjerde, J. H. Aarseth, G. Eldøen, B. Karlsen, H. Skeidsvoll, and N. E. Gilhus. EEG background activity described by a large computerized database. *Clinical Neurophysiology*, 115(3):665–673, 2004.

[16] B. Edmunds, Z. Peng, and W. Yin. TMAC: A Toolbox of Modern Async-Parallel, Coordinate, Splitting, and Stochastic Methods. *arXiv preprint arXiv:1606.04551*, 2016.