Real-Time Robust Signal Space Separation for Magnetoencephalography

Chenlei Guo, Xin Li*, *Member, IEEE*, Samu Taulu, Wei Wang, and Douglas J. Weber, *Member, IEEE*

***Abstract*—In this paper, we develop a robust signal space separation (rSSS) algorithm for real-time magnetoencephalography (MEG) data processing. rSSS is based on the spatial signal space separation (SSS) method and it applies robust regression to automatically detect and remove bad MEG channels so that the results of SSS are not distorted. We extend the existing robust regression algorithm via three important new contributions: 1) a low-rank solver that efficiently performs matrix operations; 2) a subspace iteration scheme that selects bad MEG channels using low-order spherical harmonic functions; and 3) a parallel computing implementation that simultaneously runs multiple tasks to further speed up numerical computation. Our experimental results based on both simulation and measurement data demonstrate that rSSS offers superior accuracy over the traditional SSS algorithm, if the MEG data contain significant outliers. Taking advantage of the proposed fast algorithm, rSSS achieves more than 75× runtime speedup compared to a direct solver of robust regression. Even though rSSS is currently implemented with MATLAB, it already provides sufficient throughput for real-time applications.**

***Index Terms*—Magnetoencephalography (MEG), robust regression, signal space separation.**

I. Introduction

FOR decades, great efforts have been made to study human brain and find its underlying principles to learn, process, and store information from outside world. Magnetoencephalography (MEG) can noninvasively measure the magnetic fields generated by neural activities [1]. Because of its superior temporal resolution [2], [3], MEG is considered as a promising neuroimaging method that is complementary to electroencephalography (EEG) and functional magnetic resonance imaging (fMRI), and it has been applied to both neuroscience studies [4]–[7] and clinical applications [8].

MEG signals are extremely weak (typically 50–500 fT) [1], as compared to external interferences. Many techniques have been proposed to measure and process MEG signals to enhance SNR. For instance, superconducting quantum interference devices (SQUIDs) [9], [10] are used to detect weak MEG signals, and magnetically shielded rooms are designed to reduce interferences [11]–[13]. In addition to these hardware-based approaches, a number of signal processing algorithms, such as signal space projection (SSP) [14] and signal space separation (SSS) [15]–[17], have been developed to further improve SNR by "software magnetic shielding". Taking SSS as an example, it creates two subspaces corresponding to brain signals and interferences, respectively. As such, interferences can be efficiently removed from the recorded MEG data. Other than interference reduction, SSS can also be applied to coordinate alignment [16], movement compensation [16], and dc field measurement [18].

While SSS was extensively studied in the past, existing SSS algorithms [15]–[17] mainly focus on off-line data analysis. In this paper, we aim to develop a new real-time SSS tool to facilitate online processing of MEG data. Such a real-time capability is of great importance for a number of neuroscience studies, e.g., brain–computer interface (BCI) [5], [7], since it enables active feedback from human subjects during the experiment. It also allows neuroscience researchers to deploy dynamic experiment paradigms and determine optimal stimulus on the fly. Extending SSS to real time, however, is not trivial, primarily due to the following two reasons.

1) *Runtime*: A real-time SSS implementation must be sufficiently fast to maintain high throughput and small latency. The sampling frequency of MEG is typically 1 kHz or even higher. Hence, one SSS run must be finished within at least 1 ms.
2) *Robustness*: As a spatial filtering technique, SSS is sensitive to "bad channels" that are saturated or contain large nonmagnetic interferences [15]–[17]. Spurious interferences are often observed for these bad channels, and they can substantially distort SSS results. Traditionally, bad channels are detected and removed by statistically analyzing the recorded MEG data over a timing window (e.g., a few seconds). Such a channel-screening scheme was implemented in the MaxFilter software developed by Elekta Neuromag. This approach, however, does not fit the need of real-time processing, as it results in large latency that prevents us from generating the SSS results in a short time (e.g., within a few milliseconds).

The combination of these two issues renders a challenging problem of real-time SSS implementation. Simply applying existing SSS algorithms cannot achieve the goal of real-time processing. Instead, new algorithms and tools must be developed to address this open problem.

Manuscript received August 7, 2009; revised December 12, 2009; accepted January 21, 2010. Date of publication February 18, 2010; date of current version July 14, 2010. *Asterisk indicates corresponding author*.

C. Guo was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: chenlei.guo@gmail.com).

*X. Li is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: xinli@ece.cmu.edu).

S. Taulu is with Elekta Neuromag Oy, Helsinki 00530, Finland (e-mail: Samu.Taulu@neuromag.fi).

W. Wang and D. J. Weber are with the Department of Physical Medicine and Rehabilitation and the Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA 15213 USA (e-mail: wangw4@upmc.edu; djw50@pitt.edu).

Digital Object Identifier 10.1109/TBME.2010.2043358

0018-9294/$26.00 © 2010 IEEE

In this paper, we propose a new robust SSS (rSSS) algorithm that is particularly tuned for real-time applications. rSSS borrows the concept of robust regression [25] to automatically detect and remove bad MEG channels based on the residual of the spherical harmonic expansion of SSS. For instance, if a channel shows extremely large residual, it is likely to contain significant, possibly nonmagnetic, interference and, hence, a small weight should be assigned to it when calculating the spherical harmonic expansion. By applying robust regression, the weight values of all channels can be quickly determined by solving a nonlinear optimization problem.

In addition, to address the runtime issue, we extend the existing robust regression algorithm via three important new contributions: 1) a low-rank solver that performs robust regression with efficient matrix operations; 2) a subspace iteration scheme that accurately determines the weight values of all MEG channels using low-order spherical harmonic functions; and 3) a parallel computing implementation that exploits multi-core microprocessors [19], [20] to further speed up numerical computation. As will be demonstrated by the experimental results in Section V, the combination of these proposed techniques significantly reduces the runtime (by 75×) for our real-time rSSS. Even though rSSS is currently implemented with MATLAB, it is already sufficiently fast for real-time applications.

The remainder of this paper is organized as follows. In Section II, we review the important background for SSS. We propose our new rSSS algorithm in Section III and discuss several implementation issues in Section IV. The efficiency of rSSS is demonstrated by both simulation and measurement data in Section V. Finally, we conclude in Section VI.

II. Background

Consider an MEG acquisition system where all SQUID coils are located in a source-free region. In this case, the magnetic field $\mathbf{B}(\mathbf{r})$ can be represented as the gradient of a scalar potential $V(\mathbf{r})$ that is free of singularity and harmonic in the volume containing the coils [15], [16], [27]

$$\mathbf{B}(\mathbf{r}) = -\mu_0 \cdot \nabla V(\mathbf{r}) \tag{1}$$

where μ_0 stands for the permeability of free space and the symbol "·" denotes multiplication. The harmonic potential $V(\mathbf{r})$ satisfies the Laplace's equation

$$\nabla^2 V(\mathbf{r}) = 0. \tag{2}$$

In a spherical coordinate system, the solution $V(\mathbf{r})$ of (2) can be approximated as the expansion of spherical harmonic functions

$$V(\mathbf{r}) = \sum_{l=1}^{L_{\mathrm{In}}} \sum_{m=-l}^{l} \alpha_{lm} \cdot \frac{Y_{lm}(\theta,\varphi)}{r^{l+1}} + \sum_{l=1}^{L_{\mathrm{Out}}} \sum_{m=-l}^{l} \beta_{lm} \cdot r^l \cdot Y_{lm}(\theta,\varphi) \tag{3}$$

where L_{In} and L_{Out} determine the expansion order, α_{lm} and β_{lm} represent the expansion coefficients,

$$Y_{lm}(\theta,\varphi) = \sqrt{\frac{2l+1}{4\pi} \cdot \frac{(l-m)!}{(l+m)!}} \cdot P_{lm}(\cos\theta) \cdot e^{im\varphi} \tag{4}$$

is the normalized spherical harmonic function, r, θ, and φ are the spherical coordinates, $P_{lm}(\cos\theta)$ denotes the associated Legendre function, and i stands for the imaginary unit. In this paper, we use the symbol M to represent the total number of spherical harmonic functions in (3). It is easy to verify that M is equal to $L_{\mathrm{In}}^2 + 2L_{\mathrm{In}} + L_{\mathrm{Out}}^2 + 2L_{\mathrm{Out}}$, as shown in [15] and [16].

The spherical harmonic functions in (4) have complex values. In practice, $V(\mathbf{r})$ is real-valued. We can also construct a real-valued expansion for $V(\mathbf{r})$ by using cos($m\varphi$) and sin($m\varphi$) to replace exp($im\varphi$) [21]

$$Y_{lm}(\theta,\varphi) = \begin{cases} \sqrt{\dfrac{2l+1}{4\pi} \cdot \dfrac{(l-m)!}{(l+m)!}} \cdot P_{lm}(\cos\theta) \cdot \cos(m\varphi), \\ \qquad (m \geq 0) \\ \sqrt{\dfrac{2l+1}{4\pi} \cdot \dfrac{(l-m)!}{(l+m)!}} \cdot P_{lm}(\cos\theta) \cdot \sin(m\varphi), \\ \qquad (m < 0). \end{cases} \tag{5}$$

In what follows, we will always use the real-valued spherical harmonic functions to illustrate the mathematical formulations of SSS and rSSS.

Given N MEG channels, we represent the measured signal vector $\boldsymbol{\psi} \in R^{N\times 1}$ corresponding to each of the terms in (3). Denote the signal vectors corresponding to $Y_{lm}(\theta,\varphi)/r^{l+1}$ and $r^l Y_{lm}(\theta,\varphi)$ as $\mathbf{a}_{lm} \in R^{N\times 1}$ and $\mathbf{b}_{lm} \in R^{N\times 1}$, respectively. We have

$$\boldsymbol{\psi} = \sum_{l=1}^{L_{\mathrm{In}}} \sum_{m=-l}^{l} \alpha_{lm} \cdot \mathbf{a}_{lm} + \sum_{l=1}^{L_{\mathrm{Out}}} \sum_{m=-l}^{l} \beta_{lm} \cdot \mathbf{b}_{lm}. \tag{6}$$

Equation (6) can be rewritten as the matrix form

$$\boldsymbol{\psi} = \mathbf{S} \cdot \mathbf{x} = [\,\mathbf{S}_{\mathrm{In}} \quad \mathbf{S}_{\mathrm{Out}}\,] \cdot \begin{bmatrix} \mathbf{x}_{\mathrm{In}} \\ \mathbf{x}_{\mathrm{Out}} \end{bmatrix} \tag{7}$$

where $\mathbf{S}_{\mathrm{In}}$ and $\mathbf{S}_{\mathrm{Out}}$ contain the corresponding vectors $\mathbf{a}_{lm}$ and $\mathbf{b}_{lm}$, respectively, and $\mathbf{x}_{\mathrm{In}}$ and $\mathbf{x}_{\mathrm{Out}}$ contain the corresponding coefficients α_{lm} and β_{lm}, respectively.

Equation (7) contains N linear equations for M problem unknowns, i.e., $\mathbf{S} \in R^{N\times M}$, $\mathbf{x} \in R^{M\times 1}$, and $\boldsymbol{\psi} \in R^{N\times 1}$. For a typical SSS implementation, the expansion order is set to $L_{\mathrm{In}} = 8$ and $L_{\mathrm{Out}} = 4$ in (3), resulting in $M = 104$ spherical harmonic functions in total [15], [16]. On the other hand, the Elekta Neuromag system contains $N = 306$ MEG channels and, hence, (7) is overdetermined. SSS finds the least squares solution $\mathbf{x}$ of (7) by minimizing the objective function $g(\mathbf{x})$

$$g(\mathbf{x}) = \frac{1}{2} \cdot \|\boldsymbol{\varepsilon}\|_2^2 = \frac{1}{2} \cdot \sum_{n=1}^{N} \varepsilon_n^2 \tag{8}$$

where the residual $\boldsymbol{\varepsilon} \in R^{N \times 1}$ is equal to

$$\boldsymbol{\varepsilon} = \boldsymbol{\psi} - \mathbf{S} \cdot \mathbf{x} \tag{9}$$

$\|\boldsymbol{\varepsilon}\|_2$ denotes the L_2-norm of the vector $\boldsymbol{\varepsilon}$, and ε_n is the nth element of $\boldsymbol{\varepsilon}$. It has been shown that $\mathbf{S}_{\mathrm{In}}\mathbf{x}_{\mathrm{In}}$ corresponds to brain signals and $\mathbf{S}_{\mathrm{Out}}\mathbf{x}_{\mathrm{Out}}$ represents interferences. Once the least squares solution $\mathbf{x}$ is solved by QR decomposition [26], the external interferences can be efficiently removed by leaving out the component $\mathbf{S}_{\mathrm{Out}}\mathbf{x}_{\mathrm{Out}}$. More details of SSS can be found in [15] and [16].

The traditional SSS algorithm relies on least squares regression. Its results can be substantially distorted by the bad channels that are saturated or contain large nonmagnetic interferences. To address this issue, the MaxFilter software developed by Elekta Neuromag implemented a channel screening routine to detect and remove bad channels by statistically analyzing the recorded MEG data over a timing window (e.g., a few seconds). Such an approach, however, is not applicable to real-time applications, as it results in large latency that prevents us from generating the SSS results in a short time (e.g., within a few milliseconds). It, in turn, motivates us to propose a new rSSS algorithm to detect bad channels in real time.

III. Robust Signal Space Separation

A. Mathematical Formulation

The traditional SSS uses least squares regression to minimize the objective function $g(\mathbf{x})$ in (8) where $g(\mathbf{x})$ quadratically increases with the residual $\{|\varepsilon_n|, n = 1, 2, \ldots, N\}$ for all channels. If the nth channel has an extremely large residual $|\varepsilon_n|$, it can significantly impact $g(\mathbf{x})$ and the least squares solution $\mathbf{x}$ of (7) will be particularly tuned to reduce $|\varepsilon_n|$.

In practice, however, a large residual $|\varepsilon_n|$ typically implies that the nth channel is an outlier. For example, it can be saturated or distorted by large interferences from SQUID coils and/or electronics. In this case, simply minimizing the residual $|\varepsilon_n|$ does not lead to the solution of interest. Instead, we should ignore $|\varepsilon_n|$ so that the interferences in the nth channel do not distort the SSS result. Since (7) is overdetermined, a meaningful solution $\mathbf{x}$ can still be found, even if a few bad channels are removed. In other words, the over-determined equation (7) contains redundant information and, hence, the MEG signals of the bad channels can be predicted from the other channels using SSS.

The proposed rSSS algorithm borrows the robust regression idea from statistics [25] to redefine the objection function $g(\mathbf{x})$ in (8) so that a large residual at the outlier is not overpenalized. In particular, the objective function $g(\mathbf{x})$ is changed to

$$g(\mathbf{x}) = \sum_{n=1}^{N} \rho(\varepsilon_n) \tag{10}$$

where $\rho(\varepsilon_n)$ defines the error penalty of the nth channel. There are many possible ways to define $\rho(\varepsilon_n)$, as long as it satisfies the following properties [25]: 1) $\rho(\varepsilon_n) \geq 0$; 2) $\rho(\varepsilon_n = 0) = 0$; 3) $\rho(\varepsilon_n) = \rho(-\varepsilon_n)$; and 4) $\rho(\varepsilon_n) \geq \rho(\varepsilon_m)$ for any $|\varepsilon_n| \geq |\varepsilon_m|$.

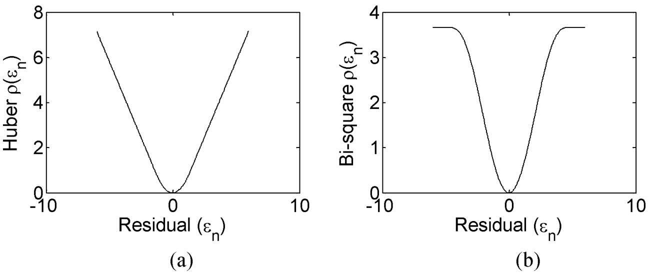

Fig. 1. Two objective function examples for robust regression. (a) Huber function in (11) where $k = 1.34$. (b) Bisquare function in (12) where $k = 4.69$.

For least-squares regression, we simply have $\rho(\varepsilon_n) = 0.5 \cdot \varepsilon_n^2$. For robust regression, the Huber function

$$\rho(\varepsilon_n) = \begin{cases} \dfrac{1}{2} \cdot \varepsilon_n^2 & (|\varepsilon_n| \leq k) \\[2ex] k \cdot |\varepsilon_n| - \dfrac{1}{2} \cdot \varepsilon_n^2, & (|\varepsilon_n| > k) \end{cases} \tag{11}$$

and the bisquare function

$$\rho(\varepsilon_n) = \begin{cases} \left\{ \dfrac{1}{6} \cdot \left\{ 1 - \left[1 - \left(\dfrac{\varepsilon_n}{k} \right)^2 \right]^3 \right\} \right., & (|\varepsilon_n| \leq k) \\[2ex] \dfrac{k^2}{6}, & (|\varepsilon_n| > k) \end{cases} \tag{12}$$

are two important objective functions that have been widely used in the literature. In (11) and (12), k is a constant that can be either manually specified by a user or statistically determined by the data [25]. Fig. 1(a) and (b) plots the Huber function ($k = 1.34$) and the bisquare function ($k = 4.69$), respectively. Note that for an increasingly large residual $|\varepsilon_n|$, both the Huber function and the bisquare function increase much more slowly than the quadratic function that is used for least squares regression. In other words, a large residual is much less penalized than that in least-squares regression. This is the reason why robust regression is not sensitive to outliers. If a proper objection function $\rho(\varepsilon_n)$ is used, the result solved by robust regression can be almost independent of outliers. In Section IV, we will construct a unique objective function $\rho(\varepsilon_n)$ for rSSS so that the proposed signal space separation is both robust (i.e., insensitive to outliers) and efficient (i.e., low computational complexity).

Once the objective function (10) is determined, the solution $\mathbf{x}$ of the over-determined equation (7) can be solved by minimizing (10). Unlike least squares regression where $\mathbf{x}$ can be efficiently found by QR decomposition [26], no closed-form solution exists for robust regression. It must be treated as a nonlinear optimization problem and numerically solved by an iterative algorithm.

B. Iterative Solver

In this section, we describe an iteratively reweighted least squares (IRLS) algorithm that was previously developed for robust regression [22]. We show the details of this algorithm, since the mathematical formulations will be reused in Section IV-A to derive our own fast solver for rSSS.

Given the linear equation $\boldsymbol{\psi} = \mathbf{S} \cdot \mathbf{x}$ in (7), we first rewrite the nth equation as

$$\psi_n = \mathbf{S}_n \cdot \mathbf{x}, \qquad (n = 1, 2, \ldots, N) \tag{13}$$

where ψ_n is the nth element of $\boldsymbol{\psi}$, $\mathbf{S}_n \in R^{1\times M}$ denotes the nth row of $\mathbf{S}$, and N is the total number of MEG channels. Substituting (13) into (10) yields

$$g(\mathbf{x}) = \sum_{n=1}^{N} \rho(\psi_n - \mathbf{S}_n \cdot \mathbf{x}). \tag{14}$$

If $g(\mathbf{x})$ is continuously differentiable and $\mathbf{x}$ is the optimal solution that minimizes (14), the gradient $\partial g/\partial \mathbf{x}$ should be zero, implying

$$\begin{aligned} \frac{\partial g}{\partial \mathbf{x}} &= \frac{\partial}{\partial \mathbf{x}} \sum_{n=1}^{N} \rho(\psi_n - \mathbf{S}_n \cdot \mathbf{x}) \\ &= -\sum_{n=1}^{N} \left[\varphi(\psi_n - \mathbf{S}_n \cdot \mathbf{x}) \cdot \mathbf{S}_n^T\right] = \mathbf{0} \end{aligned} \tag{15}$$

where $\varphi(\varepsilon_n) = \mathrm{d}\rho(\varepsilon_n)/\mathrm{d}\varepsilon_n$ is the derivative of $\rho(\varepsilon_n)$. Define the weight $w(\varepsilon_n) = \varphi(\varepsilon_n)/\varepsilon_n$ and (15) can be rewritten as

$$\sum_{n=1}^{N} \left[w(\varepsilon_n) \cdot (\psi_n - \mathbf{S}_n \cdot \mathbf{x}) \cdot \mathbf{S}_n^T\right] = \mathbf{0} \tag{16}$$

or equivalently

$$\sum_{n=1}^{N} \left[\mathbf{S}_n^T \cdot w(\varepsilon_n) \cdot \mathbf{S}_n\right] \cdot \mathbf{x} = \sum_{n=1}^{N} \left[\mathbf{S}_n^T \cdot w(\varepsilon_n) \cdot \psi_n\right]. \tag{17}$$

Equation (17) can be further represented as the matrix form

$$\left(\mathbf{S}^T \cdot \mathbf{W} \cdot \mathbf{S}\right) \cdot \mathbf{x} = \mathbf{S}^T \cdot \mathbf{W} \cdot \boldsymbol{\psi} \tag{18}$$

where $\mathbf{W} = \mathrm{diag}[w(\varepsilon_1), w(\varepsilon_2), \ldots, w(\varepsilon_N)] \in R^{N\times N}$ is a diagonal matrix.

For least squares regression, the weight $\mathbf{W}$ is simply an identity matrix, since its diagonal elements are all equal to one: $\varphi(\varepsilon_n) = \mathrm{d}\rho(\varepsilon_n)/\mathrm{d}\varepsilon_n = \varepsilon_n$ and $w(\varepsilon_n) = \varphi(\varepsilon_n)/\varepsilon_n = 1$. In this case, (18) becomes $(\mathbf{S}^T \cdot \mathbf{S}) \cdot \mathbf{x} = \mathbf{S}^T \cdot \boldsymbol{\psi}$, and its solution $\mathbf{x} = (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot (\mathbf{S}^T \cdot \boldsymbol{\psi})$ is exactly the least squares solution of the over-determined equation (7). Note that least squares regression treats all channels equally, as the same weight $\{w(\varepsilon_n) = 1, n = 1, 2, \ldots, N\}$ is assigned to all of them.

Unlike least squares regression, the weight $\mathbf{W}$ strongly depends on the residual $\boldsymbol{\varepsilon}$ for robust regression. For example, the weight $w(\varepsilon_n)$ is

$$w(\varepsilon_n) = \begin{cases} 1, & (|\varepsilon_n| \le k) \\ k/|\varepsilon_n|, & (|\varepsilon_n| > k) \end{cases} \tag{19}$$

for the Huber function and it is

$$w(\varepsilon_n) = \begin{cases} [1 - (\varepsilon_n/k)^2]^2, & (|\varepsilon_n| \le k) \\ 0, & (|\varepsilon_n| > k) \end{cases} \tag{20}$$

for the bisquare function. Fig. 2(a) and (b) plot the weight $w(\varepsilon_n)$ for the Huber function ($k = 1.34$) and the bisquare function ($k = 4.69$), respectively. Studying Fig. 2, we can make two important observations. First, $w(\varepsilon_n)$ is between zero and one. This is always true, as long as the objection/weight function is properly scaled. Second, $w(\varepsilon_n)$ decreases as $|\varepsilon_n|$ increases. It implies that if the nth channel has a large residual $|\varepsilon_n|$, a small weight $w(\varepsilon_n)$ will be assigned to it. In other words, this channel is considered as an outlier and its residual will not be overpenalized during robust regression. In the extreme case, if $|\varepsilon_n|$ exceeds the value k for the bisquare function in (20), $w(\varepsilon_n)$ becomes exactly zero, implying that the nth channel is completely removed. From this point of view, the weight matrix $\mathbf{W}$ in (18) plays an important role in detecting and removing bad MEG channels for the proposed rSSS algorithm.

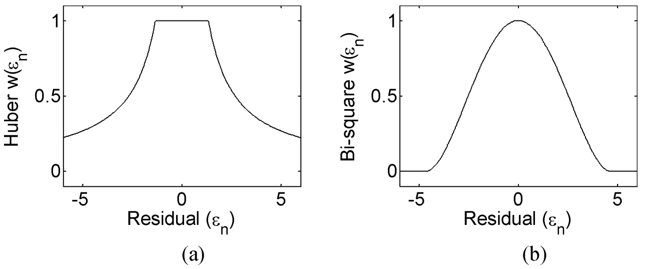

Fig. 2. Two weight function examples for robust regression. (a) Huber function in (19) where $k = 1.34$. (b) Bisquare function in (20) where $k = 4.69$.

It should also be noted that since $\mathbf{W}$ is a nonlinear function of $\boldsymbol{\varepsilon} = \boldsymbol{\psi} - \mathbf{S} \cdot \mathbf{x}$, (18) is not simply a linear equation of $\mathbf{x}$. Hence, no closed-form solution exists. To solve (18), we start from $\mathbf{W} = \mathbf{I}$, where $\mathbf{I}$ stands for the identity matrix. Given this initial value of $\mathbf{W}$, we solve (18) to get an estimate of $\mathbf{x}$. Next, we update $\mathbf{W}$ based on the residual $\boldsymbol{\varepsilon} = \boldsymbol{\psi} - \mathbf{S} \cdot \mathbf{x}$, and then calculate $\mathbf{x}$ by (18) again. These two steps are repeatedly applied to update $\mathbf{W}$ and $\mathbf{x}$, until convergence is reached. Since this iterative algorithm repeatedly solves (18) that can be conceptually considered as weighted least-squares regression, it is referred to as the IRLS method in [22]. Algorithm 1 summarizes the major steps of the proposed rSSS algorithm, where IRLS is used to solve the robust regression problem.

Algorithm 1: Robust Signal Space Separation

1) Follow the traditional SSS method to formulate the over-determined equation (7).
2) Select an appropriate objective function $\rho(\varepsilon_n)$ and determine the corresponding weight function $w(\varepsilon_n)$.
3) Find the least-squares solution of (7): $\mathbf{x}^{(0)} = (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot (\mathbf{S}^T \cdot \boldsymbol{\psi})$. Set the iteration index $p = 1$.
4) Calculate the residual: $\boldsymbol{\varepsilon}^{(p)} = \boldsymbol{\psi} - \mathbf{S} \cdot \mathbf{x}^{(p-1)}$ and update the weight matrix $\mathbf{W}^{(p)}$.
5) Substitute $\mathbf{W}^{(p)}$ into (18) and solve the linear equation: $\mathbf{x}^{(p)} = [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}]^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}]$.
6) If $\|\mathbf{x}^{(p)} - \mathbf{x}^{(p-1)}\|_2 < \delta$, where δ is a user-defined tolerance, stop iteration. Otherwise, $p = p + 1$ and go to Step 4.

The computational cost of Algorithm 1 is dominated by Step 5, i.e., solving the linear equation: $\mathbf{x}^{(p)} = [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}]^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}]$. Given $\mathbf{S} \in R^{N\times M}$ and $\mathbf{W}^{(p)} \in R^{N\times N}$, $[\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}]$ is an $M \times M$ matrix. In a typical implementation of SSS, M is equal to 104, as shown in Section II. In other words, within each iteration of Algorithm 1, we must solve a linear equation with $M = 104$ unknowns. Even though the matrix $\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}$ is positive-definite [22], [25] and, hence, the linear equation can be solved by Cholesky decomposition [26], the computational cost is proportional to $O(M^3)$. In the following section, we will propose several novel techniques to reduce the computational cost so that the proposed rSSS algorithm is of practical utility for real-time applications.

IV. Implementation Issues

The proposed rSSS approach is made practically feasible for real-time applications by applying several novel techniques to reduce computational time. In this section, we discuss the details of these methods.

A. Low-Rank Solver

The proposed low-rank solver aims to incrementally update the matrix $\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}$ in Step 5 of Algorithm 1 so that the linear equation: $\mathbf{x}^{(p)} = [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}]^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}]$ can be solved with low computational cost. While the weight matrix $\mathbf{W}^{(p)}$ is changed in each iteration of Algorithm 1, most MEG channels should be good in practice and their weight values should be close to one. In other words, there are only a few outliers for which the corresponding diagonal elements in $\mathbf{W}^{(p)}$ deviate from one. Our proposed low-rank solver is built upon this observation. It consists of two key components: 1) a unique weight function $w(\varepsilon_n)$ that appropriately distinguishes good and bad channels; and 2) a low-rank update scheme that efficiently solves the linear equation: $\mathbf{x}^{(p)} = [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}]^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}]$.

1) Weight Function Construction: One important component of the proposed low-rank solver is to construct an appropriate weight function $w(\varepsilon_n)$ so that $w(\varepsilon_n)$ is equal to one for good channels (i.e., small residual $|\varepsilon_n|$) and zero for bad channels (i.e., large residual $|\varepsilon_n|$). The Huber function in Fig. 2(a) assigns $w(\varepsilon_n) = 1$ for good channels, but its weight does not quickly decay to zero for bad channels. Hence, a bad channel may not be completely removed. On the other hand, the bisquare function in Fig. 2(b) assigns $w(\varepsilon_n) = 0$ for bad channels, but it only yields $w(\varepsilon_n) = 1$ for $\varepsilon_n = 0$. In practice, even a good channel will not exactly have zero residual, primarily due to random physical interferences and/or approximation errors of the spherical harmonic expansion. Therefore, if the bisquare function is used, all diagonal elements of the weight matrix $\mathbf{W}^{(p)}$ must be updated in each iteration. It would be impossible to efficiently apply a low-rank update in this case.

Given these observations, we propose to combine the Huber function and the bisquare function to construct a modified bisquare weight function

$$w(\varepsilon_n) = \begin{cases} 1, & (|\varepsilon_n| \le k_1) \\ \left[1 - \left(\dfrac{\varepsilon_n - k_1}{k_2 - k_1}\right)^2\right]^2, & (k_1 < |\varepsilon_n| \le k_2) \\ 0, & (|\varepsilon_n| > k_2) \end{cases} \tag{21}$$

where k_1 and k_2 are two constants. In our implementation, the values of k_1 and k_2 are fixed: $k_1 = 1.72$ and $k_2 = 4.69$, respectively. Since the residual $\boldsymbol{\varepsilon} = \boldsymbol{\psi} - \mathbf{S} \cdot \mathbf{x}$ may vary in different cases, we always normalize the linear equation $\boldsymbol{\psi} = \mathbf{S} \cdot \mathbf{x}$ by a constant λ. Namely, we divide both sides of the equation by λ. The value of λ is equal to the standard deviation of the residual $\boldsymbol{\varepsilon} = \boldsymbol{\psi} - \mathbf{S} \cdot \mathbf{x}$ over all channels calculated from the least-squares solution. In other words, the normalized residual $(\boldsymbol{\psi} - \mathbf{S} \cdot \mathbf{x})/\lambda$ has unit variance so that we can use the same k_1 and k_2 in different cases.

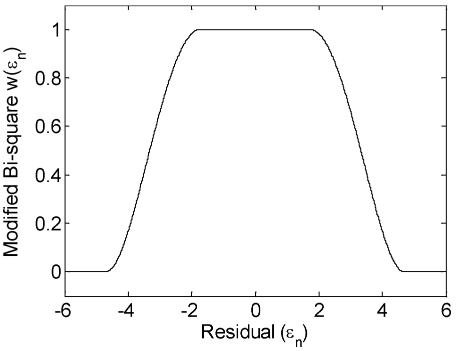

Fig. 3. The modified bi-square weight function in (21) where $k_1 = 1.72$ and $k_2 = 4.69$ results in $w(\varepsilon_n) = 1$ for good channels and $w(\varepsilon_n) = 0$ for bad channels.

Fig. 3 plots the modified bi-square weight function ($k_1 = 1.72$ and $k_2 = 4.69$, respectively). Comparing Figs. 2 and 3, we would find that the modified bisquare function results in $w(\varepsilon_n) = 1$ for good channels (similar to the Huber function) and $w(\varepsilon_n) = 0$ for bad channels (similar to the bisquare function). It, in turn, facilitates us to both accurately remove bad channels and efficiently apply low-rank update for matrix operations. Next, we will discuss the low-rank update algorithm in detail.

2) Low-Rank Update: To formulate the low-rank update for $\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}$ in Step 5 of Algorithm 1, we rewrite the matrix $\mathbf{W}^{(p)}$ as

$$\mathbf{W}^{(p)} = \mathbf{I} + [\mathbf{W}^{(p)} - \mathbf{I}] \tag{22}$$

where $\mathbf{I}$ stands for the identity matrix. As previously discussed, if the modified bisquare function in (21) is used to calculate the weight, we expect that most diagonal elements of the matrix $\mathbf{W}^{(p)}$ are equal to one. Hence, the matrix $\mathbf{W}^{(p)} - \mathbf{I}$ is low rank, i.e., it only contains a few nonzero elements on the diagonal. To explicitly represent this low-rank structure, we define the set $\{Q_j, \mathbf{W}^{(p)}_{Qj,Qj} \neq 1\}$. Namely, it contains the indexes $\{Q_j\}$ at which the (Q_j, Q_j)th element of $\mathbf{W}^{(p)} - \mathbf{I}$ is nonzero. Define:

$$\mathbf{V}^{(p)} = \begin{bmatrix} \mathbf{S}_{Q1} \\ \mathbf{S}_{Q2} \\ \vdots \end{bmatrix} \quad \boldsymbol{\Delta}^{(p)} = \begin{bmatrix} \mathbf{W}^{(p)}_{Q1,Q1} - 1 & & \\ & \mathbf{W}^{(p)}_{Q2,Q2} - 1 & \\ & & \ddots \end{bmatrix} \tag{23}$$

where $\mathbf{S}_{Qj} \in R^{1\times M}$ denotes the Q_jth row of the matrix $\mathbf{S}$ in (7). In (23), the matrix $\boldsymbol{\Delta}^{(p)}$ is much smaller than $\mathbf{W}^{(p)} - \mathbf{I}$, as it removes all zero diagonal elements of $\mathbf{W}^{(p)} - \mathbf{I}$. In our MEG experiments, the dimension of $\boldsymbol{\Delta}^{(p)}$ is typically around 5–20. There are typically less than five channels that are obviously bad (e.g., saturated). Their weight values are close to zero. Other than these bad channels, we often observe 5–15 additional channels with weight close to, but not exactly equal to, one. These channels contain noise, but they still carry useful information about the magnetic field.

Substituting (22) and (23) into $\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}$ yields

$$\begin{aligned}\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S} &= \mathbf{S}^T \cdot \mathbf{S} + \mathbf{S}^T \cdot [\mathbf{W}^{(p)} - \mathbf{I}] \cdot \mathbf{S} \\ &= \mathbf{S}^T \cdot \mathbf{S} + \mathbf{V}^{(p)T} \cdot \boldsymbol{\Delta}^{(p)} \cdot \mathbf{V}^{(p)} \end{aligned} \tag{24}$$

where $\mathbf{V}^{(p)T} \cdot \boldsymbol{\Delta}^{(p)} \cdot \mathbf{V}^{(p)}$ is a low-rank matrix. According to the Sherman–Morrison–Woodbury formula [26], the inverse of $\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}$ can be exactly represented as

$$\begin{aligned}&[\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}]^{-1} = [\mathbf{S}^T \cdot \mathbf{S}]^{-1} - [\mathbf{S}^T \cdot \mathbf{S}]^{-1} \cdot \mathbf{V}^{(p)T} \\ &\cdot \{[\boldsymbol{\Delta}^{(p)}]^{-1} + \mathbf{V}^{(p)} \cdot [\mathbf{S}^T \cdot \mathbf{S}]^{-1} \cdot \mathbf{V}^{(p)T}\}^{-1} \cdot \mathbf{V}^{(p)} \cdot [\mathbf{S}^T \cdot \mathbf{S}]^{-1}.\end{aligned} \tag{25}$$

Therefore, the solution $\mathbf{x}^{(p)}$ in Step 5 of Algorithm 1 can be written as

$$\begin{aligned}\mathbf{x}^{(p)} &= (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}] - (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot \mathbf{V}^{(p)T} \\ &\cdot \{[\boldsymbol{\Delta}^{(p)}]^{-1} + \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot \mathbf{V}^{(p)T}\}^{-1} \\ &\cdot \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\Psi}].\end{aligned} \tag{26}$$

Even though (26) looks much more complicated than its original form: $\mathbf{x}^{(p)} = [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}]^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}]$, it is computationally efficient due to two reasons. First, the matrix $\mathbf{S}^T \cdot \mathbf{S}$ is independent of the recorded MEG signals. It is uniquely determined by spherical harmonic functions. As long as the head position is fixed, $\mathbf{S}^T \cdot \mathbf{S}$ is fixed. In other words, if we do not consider real-time head movement, $\mathbf{S}^T \cdot \mathbf{S}$ does not vary over time. Hence, the inverse $(\mathbf{S}^T \cdot \mathbf{S})^{-1}$ can be precomputed, when the head position is measured at the beginning of an MEG experiment. Once $(\mathbf{S}^T \cdot \mathbf{S})^{-1}$ is known, its multiplication with other matrices and vectors in (26) can be easily calculated without solving any linear equation in real time.

Second, to calculate $\mathbf{x}$ in (26), we need to solve the linear equation: $\{[\boldsymbol{\Delta}^{(p)}]^{-1} + \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot \mathbf{V}^{(p)T}\}^{-1} \cdot \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}]$. The matrix $\boldsymbol{\Delta}^{(p)}$ is diagonal and, hence, its inverse is easy to calculate. As $\boldsymbol{\Delta}^{(p)}$ and $\mathbf{V}^{(p)}$ vary from iteration to iteration, the matrix $[\boldsymbol{\Delta}^{(p)}]^{-1} + \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot \mathbf{V}^{(p)T}$ must be repeatedly factorized at each iteration step to solve the linear equation: $\{[\boldsymbol{\Delta}^{(p)}]^{-1} + \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot \mathbf{V}^{(p)T}\}^{-1} \cdot \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}]$. However, the size of $[\boldsymbol{\Delta}^{(p)}]^{-1} + \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot \mathbf{V}^{(p)T}$ is small—it equals the number of nonzero diagonal elements of $\mathbf{W}^{(p)} - \mathbf{I}$, which is typically around 5–20 in our MEG experiments. In addition, the matrix $[\boldsymbol{\Delta}^{(p)}]^{-1} + \mathbf{V}^{(p)} \cdot (\mathbf{S}^T \cdot \mathbf{S})^{-1} \cdot \mathbf{V}^{(p)T}$ is symmetric and, hence, it can be efficiently factorized by the $\mathrm{LDL}^{\mathrm{T}}$ factorization [26].

Because of these reasons, computing (26) is much cheaper than directly solving the original linear equation: $\mathbf{x}^{(p)} = [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \mathbf{S}]^{-1} \cdot [\mathbf{S}^T \cdot \mathbf{W}^{(p)} \cdot \boldsymbol{\psi}]$. As will be demonstrated by the experimental examples in Section V, the proposed low-rank solver achieves significant runtime speedup (more than 20×) over a direct solver. Hence, it is a key technology to facilitate real-time rSSS, where latency and throughput are of great importance.

B. *Subspace Iteration*

Subspace iteration is another technique that further reduces the computational cost of rSSS. It is motivated by the fact that the dominant energy of MEG signals can be captured by the low-order spherical harmonic functions in (3). Therefore, we can use low-order spherical harmonic functions only to calculate the weight matrix $\mathbf{W}$ and identify bad channels.

The proposed subspace iteration method consists of two steps. First, the matrix $\mathbf{S}$ in (7) is constructed by using low-order harmonic functions. For example, if we set $L_{\mathrm{In}} = 5$ and $L_{\mathrm{Out}} = 4$ for the expansion (3), the matrix $\mathbf{S}$ in (7) will contain $M = 59$ harmonic functions. Note that the number of harmonic functions (i.e., M) is substantially reduced, compared to the traditional case $L_{\mathrm{In}} = 8$, $L_{\mathrm{Out}} = 4$, and $M = 104$. Since (7) contains less unknown coefficients, solving the over-determined linear equation using robust regression becomes less expensive. Next, once the weight matrix $\mathbf{W}$ is calculated from the low-order spherical harmonic functions, high-order harmonic functions are added to the matrix $\mathbf{S}$ and the weighted least squares problem in (18) is solved. The computation in this second step is cheap, since no additional iteration is applied to recalculate the weight matrix $\mathbf{W}$.

The aforementioned method runs the iterations of rSSS (i.e., Algorithm 1) in the subspace spanned by low-order spherical harmonic functions. Hence, it is referred to as subspace iteration in this paper. The runtime speedup achieved by subspace iteration will be demonstrated by our experimental examples in Section V.

C. *Parallel Computing*

A modern microprocessor typically consists of several or even hundreds of CPU cores [19], [20]. These cores can simultaneously process different input data or even run different programs with completely different instructions. To take advantage of this parallel computing capability, we implement the proposed rSSS algorithm in MATLAB with multiple parallel tasks. In particular, since rSSS is a spatial filtering technique, it handles MEG signals at different times independently. We map the recorded MEG signals to different CPU cores and run rSSS separately, as shown in Fig. 4. Although such a parallel computing strategy does not reduce latency, it helps to increase throughput. More discussions on latency and throughput will be presented in Section V-C, along with the experimental results.

V. Experimental Results

In this section, we demonstrate the efficiency of the proposed rSSS algorithm using both simulation and measurement data. For both SSS and rSSS, we set $L_{\mathrm{In}} = 8$ and $L_{\mathrm{Out}} = 4$ for the spherical harmonic expansion in (3). In addition, a set of low-order spherical harmonic functions with $L_{\mathrm{In}} = 5$ and $L_{\mathrm{Out}} = 4$ are used for the subspace iteration scheme proposed in Section IV-B.

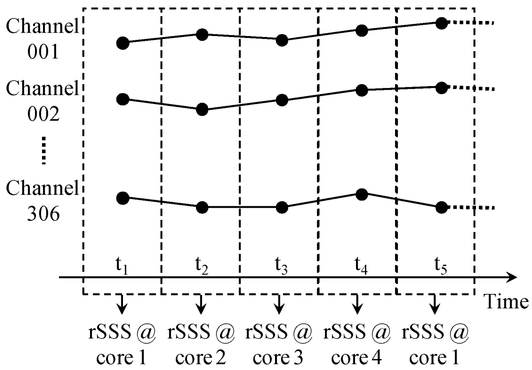

Fig. 4. An example of the proposed parallel rSSS implementation, where MEG signals are mapped to four different CPU cores to improve throughput.

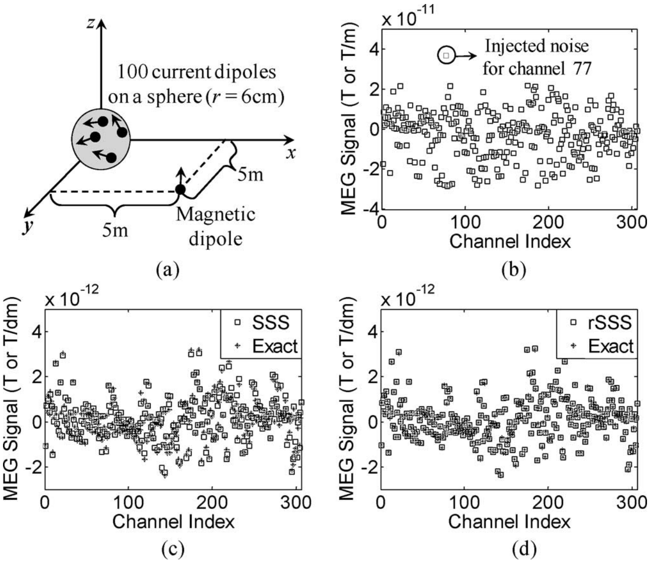

Fig. 5. SSS and rSSS results for simulation data. (a) One-hundred current dipoles and one magnetic dipole used in the simulation example. (b) Simulated MEG signals of all channels for the Elekta Neuromag 306-channel MEG system. (c) Exact and SSS-predicted MEG signals generated by the current dipoles (relative error = 19.29%). (d) Exact and rSSS-predicted MEG signals generated by the current dipoles (relative error = 2.79%).

A. Simulation Data

Fig. 5(a) shows our simulation setup where 100 current dipoles are distributed on the surface of a sphere with the radius $r = 6$ cm. These current dipoles are used to model the neural activities inside the brain. In addition, a magnetic dipole is placed far away from the origin in order to model the external interference. Similar simulation setup was used to evaluate the efficiency of SSS in the past [15], [16], [23]. For this simulation example, we know the exact magnetic field, which enables us to fully compare the accuracy of the traditional SSS and the proposed rSSS.

Given the aforementioned setup for current and magnetic dipoles, we calculate the MEG signals of all channels for the Elekta Neuromag 306-channel MEG system. This MEG system contains 204 gradiometers and 102 magnetometers. To mimic the effect of bad channels, we inject an interference 5×10^{-11} T/m at channel 77 (a randomly selected gradiometer). Fig. 5(b) plots the MEG signals (including the injected interference) for all channels.

Next, we run two signal space separation algorithms to estimate the MEG signals generated by the current dipoles: 1) the traditional SSS based on least-squares regression; and 2) the proposed rSSS using robust regression. Fig. 5(c) shows the exact solution and the SSS result. The relative error of SSS is 19.29%, where the error is defined as

$$\|\boldsymbol{\Psi}_{\rm In} - \tilde{\boldsymbol{\Psi}}_{\rm In}\|_2/\|\tilde{\boldsymbol{\Psi}}_{\rm In}\|_2. \tag{27}$$

In (27), $\boldsymbol{\psi}_{\rm In} \in R^{N\times 1}$ and $\tilde{\boldsymbol{\psi}}_{\rm In} \in R^{N\times 1}$ represent the estimated solution and the exact solution, respectively, and N is the total number of MEG channels. Note that the traditional SSS is not able to accurately recover the MEG signals in this example. Most importantly, even though we only inject interference to a single channel, the interference propagates to many other channels after SSS. In other words, if the bad channel is not properly identified and removed, the spatial filtering done by SSS can yield wrong results for a large number of MEG channels.

On the other hand, Fig. 5(d) shows the exact MEG signals and the rSSS result. Studying Fig. 5(d), one would find that the proposed rSSS algorithm accurately predicts the MEG signals at all channels. The relative error is 2.79%. We further verify that rSSS automatically assigns $w(\varepsilon_n) = 0$ to channel 77 and $w(\varepsilon_n) = 1$ to all other channels. It, in turn, implies that the bad channel is successfully detected and removed in this example.

B. Measurement Data

In our experiment, a healthy human subject sits in a chair with his head inside the MEG helmet. The subject is asked to perform center-to-out movement with his right wrist. There are four different movement directions in total: left, right, up and down. Target images are generated by a computer and projected to a non-magnetic screen in front of the subject to prompt the movement direction. The sequence of movement targets is selected randomly. One movement trial takes 2.2 seconds to complete. The MEG signals of 480 independent trials are recorded with 1 kHz sampling frequency using an Elekta Neuromag 306-channel MEG system. At the beginning of the experiment, head position indicator (HPI) signals are measured to find the head position and then the origin of spherical harmonic expansion is determined.

We select a single movement trial (out of 480 trials in total) for which the recorded MEG signals contain spurious interferences. Such a movement trial possibly contains bad channels and, hence, is a good example for us to study the efficiency of the proposed rSSS method. For the purpose of testing and comparison, both the traditional SSS and the proposed rSSS are used to estimate the MEG signals generated by the neural actives inside the brain. Fig. 6(a) plots the MEG signals recovered by both SSS and rSSS at $t = 2.16$ s. Here, we show the results at this particular time point, since SSS and rSSS yield substantially different solutions. The relative difference between these two solutions is 53.96% where the difference is defined as

$$\|\boldsymbol{\Psi}_{\rm In\_SSS}(t) - \boldsymbol{\Psi}_{\rm In\_rSSS}(t)\|_2/\|\boldsymbol{\Psi}_{\rm In\_rSSS}(t)\|_2. \tag{28}$$

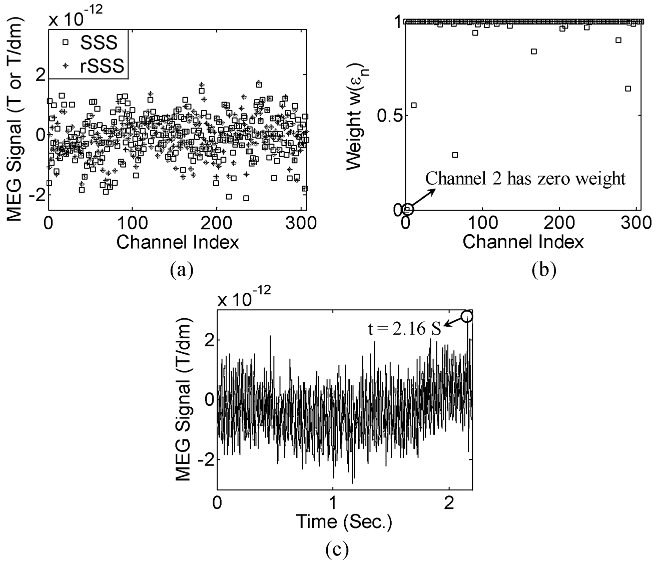

Fig. 6. SSS and rSSS results for measurement data. (a) MEG signals recovered by SSS and rSSS at $t = 2.16$ s (relative difference = 53.96%). (b) Weight $w(\varepsilon_n)$ of all channels calculated by rSSS at $t = 2.16$ s. (c) Raw MEG signals recorded by channel 2 (gradiometer) before applying SSS or rSSS.

In (28), $\psi_{\mathrm{In\_SSS}} \in R^{N\times 1}$ and $\psi_{\mathrm{In\_rSSS}} \in R^{N\times 1}$ represent the solutions calculated by SSS and rSSS, respectively, and N is the total number of MEG channels. In our experiment, we observe that the relative difference between SSS and rSSS solutions is typically around or below 10%. Hence, the large difference at $t = 2.16$ s implies the existence of malfunctioning MEG channels and it provides a good example for us to compare SSS and rSSS.

Fig. 6(b) plots the weight $w(\varepsilon_n)$ of all channels calculated by rSSS at $t = 2.16$ s. Studying Fig. 6(b), we would notice that the weight $w(\varepsilon_n)$ of channel 2 (gradiometer) is exactly 0, implying that channel 2 is considered as a bad channel at $t = 2.16$ s. In addition to channel 2, there are several other channels that are "marginally" bad, i.e., the corresponding weight values are between zero and one. The combination of all these bad channels leads to the large difference between SSS and rSSS results, as shown in Fig. 6(a). Fig. 6(c) plots the raw MEG signals recorded by channel 2, before any spatial filtering (e.g., SSS or rSSS) is applied. As labeled in Fig. 6(c), there is a sharp spike (i.e., large spurious interference) at $t = 2.16$ s. This observation matches the rSSS result. Namely, channel 2 is bad and should be ignored at $t = 2.16$ s.

To further demonstrate the efficiency of rSSS, Fig. 7(a) shows the raw MEG signals recorded by channel 86 (gradiometer) and Fig. 7(b) plots the corresponding weight $w(\varepsilon_n)$ calculated by rSSS for the same channel. Since rSSS is applied at each time point, the weight $w(\varepsilon_n)$ is plotted as a function of time in Fig. 7(b). Two important observations can be made from Fig. 7(a) and (b). First, the recorded MEG signals contain spurious interferences that occur at different times, as shown by the sharp spikes in Fig. 7(a). We believe that the interferences are caused by the electronics of MEG sensors. It appears from time to time, implying that a channel is not constantly bad. Hence, we cannot identify good or bad channels at the beginning of an experiment; instead, we must dynamically select bad channels on the fly. Second, the proposed rSSS successfully captures the bad channels in this example. Comparing Fig. 7(a) and (b), we would notice that the weight $w(\varepsilon_n)$ of channel 86 is automatically set to zero, once the spurious interferences occur. It demonstrates that the proposed rSSS algorithm using robust regression is an efficient technique to accurately and dynamically detect bad channels.

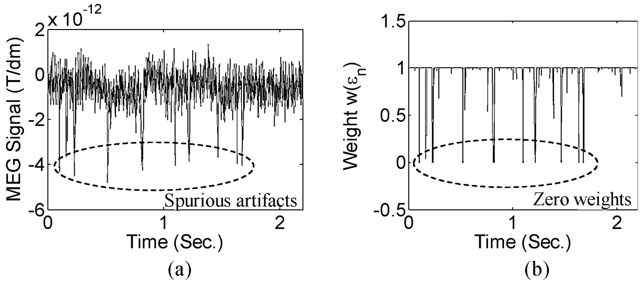

Fig. 7. MEG signals and rSSS results of channel 86 (gradiometer). (a) Raw MEG signals recorded by channel 86 before applying SSS or rSSS. (b) Weight $w(\varepsilon_n)$ of channel 86 calculated by rSSS.

TABLE I
COMPUTATIONAL TIME OF ONE rSSS RUN FOR DIFFERENT IMPLEMENTATIONS

Implementation	Runtime (ms)
Direct solver	72.09
Low-rank solver	3.50
Low-rank solver + Subspace iteration	2.40
Low-rank solver + Subspace iteration + Parallel computing	0.95

C. Computational Time

To test the runtime for rSSS and demonstrate the efficiency of the proposed fast algorithms, we implement four different versions of rSSS in MATLAB as listed in the following.

1) Direct solver: A direct Cholesky decomposition is used to solve the linear equation in Step 5 of Algorithm 1.
2) Low-rank solver: The low-rank solver proposed in Section IV-A is used to solve the linear equation in Step 5 of Algorithm 1.
3) Low-rank solver + Subspace iteration: Both the low-rank solver (Section IV-A) and the subspace iteration scheme (Section IV-B) are used to speed up numerical computation.
4) Low-rank solver + Subspace iteration + Parallel computing: All these three techniques discussed in Section IV are integrated to achieve the smallest runtime. Four parallel tasks are managed by MATLAB to increase throughput.

We run these four implementations on a Linux server with four dual-core AMD 2.8 GHz Opteron processors. Table I shows the computational time of one rSSS run for these four implementations. Note that more than 75× runtime speedup is achieved by the proposed fast algorithms compared to a direct solver. Even though the rSSS algorithm is currently implemented with MATLAB, one rSSS run can be finished within 0.95 ms. If the MEG sampling frequency is 1 kHz, i.e., one new signal vector per 1 ms, the throughput of our MATLAB implementation already meets the requirement of real-time processing.

On the other hand, since we run four parallel tasks simultaneously, the average latency of rSSS is around 0.95 × 4 = 3.8 ms. In a real-time MEG acquisition system, we also need to consider the latency for data collection, data communication, and other signal processing steps. Hence, the total latency will be larger than 3.8 ms. In addition, the variation of latency (i.e., jitter) is also of great importance for many timing-critical applications [24]. All these open questions related to latency will be further studied in our future research.

VI. Conclusion

In this paper, we develop a new rSSS algorithm for real-time interference reduction of MEG data. In particular, we propose to incorporate robust regression with SSS in order to efficiently detect and remove bad MEG channels in real time. Three different but complementary techniques, i.e., 1) a low-rank solver; 2) a subspace iteration scheme; and 3) a parallel computing implementation, are developed to reduce the computational cost of rSSS.

Our experimental results based on both simulation and measurement data demonstrate that rSSS successfully detects and removes bad MEG channels so that the results of SSS are not distorted. In addition, the proposed fast algorithms achieve significant (i.e., more than 75×) runtime speedup over a direct solver. Even though rSSS is currently implemented with MATLAB, it already provides sufficient throughput for real-time MEG signal processing.

While the proposed rSSS method shows promising results, there remain a number of open questions that should be further studied in the future as listed in the following.

1) The proposed rSSS algorithm is efficient, if and only if a limited number of MEG channels are simultaneously bad. To successfully apply rSSS, the number of good MEG channels must be greater than the number of spherical harmonic functions $M = L_{\mathrm{In}}^2 + 2L_{\mathrm{In}} + L_{\mathrm{Out}}^2 + 2L_{\mathrm{Out}}$. As a result, redundant information exists and we can accurately predict the MEG signals by removing bad channels. In our experiments, there are 306 MEG channels in total and the number of spherical harmonic functions is $M = 104$ ($L_{\mathrm{In}} = 8$ and $L_{\mathrm{Out}} = 4$, respectively). We typically observe that less than 5 channels are obviously bad (e.g., saturated) and 5–15 additional channels contain large noise. In this case, the number of good MEG channels is much greater than the number of spherical harmonic functions and, hence, rSSS has been demonstrated with great efficiency. It should be noted, however, that rSSS may fail to work, if a large number of bad channels exist (e.g., most SQUID coils are broken).

2) There are a number of practical cases where the sources of external interferences are close to the SQUID coils for MEG recording. Such sources may consist of stimulators, magnetized EEG electrodes, etc. Similar to the traditional SSS algorithm, the proposed rSSS method cannot accurately separate brain signals from external interferences in such cases. The spatiotemporal SSS method (tSSS) was developed to address this important limitation [17]. tSSS requires to analyze MEG data over a timing window and, hence, it cannot be directly applied to real-time signal processing. An important problem for future research is how to integrate rSSS with tSSS for real-time applications.

3) The traditional SSS method can be used to correct the distortion caused by head movement during MEG recording [16]. Such movement compensation may not be necessary for cooperative healthy subjects. However, it is extremely important for some special patient groups (e.g., small children), if head movement is unavoidable. The proposed rSSS (in particular, the low-rank solver discussed in Section IV-A) relies on the assumption that head position is fixed. Otherwise, the matrix $(\mathbf{S}^T \cdot \mathbf{S})^{-1}$ in (26) cannot be precomputed and, hence, the rSSS problem cannot be efficiently solved. As an important component of our future research, we will further study efficient real-time algorithms to handle movement compensation problems.

4) In our current implementation, the order of harmonic expansion is fixed (i.e., $L_{\mathrm{In}} = 8$ and $L_{\mathrm{Out}} = 4$). When the subspace iteration method is applied to bad channel detection in Section IV-B, L_{In} is further decreased to $L_{\mathrm{In}} = 5$ to reduce computational cost. Ideally, the expansion order L_{In} and L_{Out} should be adaptively learned, since their optimal values can vary from case to case. This adaptive learning problem will be explored in our future research.

5) The proposed rSSS algorithm is implemented with MATLAB, which enables us to quickly build a prototype tool to validate the algorithm. However, MATLAB is not the best way to implement real-time applications. As an important follow up work, we will focus on the development of a real-time MEG acquisition and processing system based on rSSS. Towards this goal, we will explore various implementation options (both hardware and software) for timing-critical applications, where latency and jitter are of great importance.

Acknowledgment

The authors would like to thank all anonymous reviewers for their excellent comments that help to improve the quality of this paper.

References

[1] M. Hämäläinen, R. Hari, R. Ilmoniemi, J. Knuutila, and O. Lounasmaa, "Magnetoencephalography—Theory, instrumentation, and applications to noninvasive studies of the working human brain," *Rev. Modern Phys.*, vol. 65, no. 2, pp. 413–497, Apr. 1993.

[2] S. Baillet, J. Mosher, and R. Leahy, "Electromagnetic brain mapping," *IEEE Signal Process. Mag.*, vol. 18, no. 6, pp. 14–30, Nov. 2001.

[3] B. He and Z. Liu, "Multimodal functional neuroimaging: Integrating functional MRI and EEG/MEG," *IEEE Rev. Biomed. Eng.*, vol. 1, pp. 23–40, 2008.

[4] N. Hoogenboom, J. Schoffelen, R. Oostenveld, L. Parkes, and P. Fries, "Localizing human visual gamma-band activity in frequency, time and space," *NeuroImage*, vol. 29, pp. 764–773, 2006.

[5] J. Mellinger, G. Schalk, C. Braun, H. Preissl, W. Rosenstiel, N. Birbaumer, and A. Kübler, "An MEG-based brain-computer interface (BCI)," *NeuroImage*, vol. 36, pp. 581–593, 2007.

[6] M. Guimaraes, D. Wong, E. Uy, L. Grosenick, and P. Suppes, "Single-trial classification of MEG recordings," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 3, pp. 436–443, Mar. 2007.

[7] S. Waldert, H. Preissl, E. Demandt, C. Braun, N. Birbaumer, A. Aertsen, and C. Mehring, "Hand movement direction decoded from MEG and EEG," *J. Neuroscience*, vol. 28, no. 4, pp. 1000–1008, Jan. 2008.

[8] F. Lopes da Silva, "The impact of EEG/MEG signal processing and modeling in the diagnostic and management of epilepsy," *IEEE Rev. Biomed. Eng.*, vol. 1, pp. 143–156, Dec. 2008.

[9] J. Clarke and A. Barginski, *The SQUID Handbook: Fundamentals and Technology of SQUIDs and SQUID Systems*. Hoboken, NJ: Wiley, 2004.

[10] J. Clarke and A. Barginski, *The SQUID Handbook: Applications of SQUIDs and SQUID Systems*. Hoboken, NJ: Wiley, 2006.

[11] V. Kelhä, J. Pukki, R. Peltonen, A. Penttinen, R. Ilmoniemi, and J. Heino, "Design, construction, and performance of a large-volume magnetic shield," *IEEE Trans. Magn.*, vol. 18, no. 1, pp. 260–270, Jan. 1982.

[12] G. Kajiwara, K. Harakawa, and H. Ogata, "High-performance magnetically shielded room," *IEEE Trans. Magn.*, vol. 32, no. 4, pp. 2582–2585, Jul. 1996.

[13] J. Zimmerman, "SQUID instruments and shielding for low-level magnetic measurements," *J. Appl. Phys.*, vol. 48, no. 2, pp. 702–710, Feb. 1977.

[14] M. Uusitalo and R. Ilmoniemi, "Signal-space projection method for separating MEG or EEG into components," *Med. Biol. Eng. Comput.*, vol. 35, no. 2, pp. 135–140, 1997.

[15] S. Taulu and M. Kajola, "Presentation of electromagnetic multichannel data: The signal space separation method," *J. Appl. Phys.*, vol. 97, Jun. 2005.

[16] S. Taulu, J. Simola, and M. Kajola, "Applications of the signal space separation method," *IEEE Trans. Signal Process.*, vol. 53, no. 9, pp. 3359–3372, Sep. 2005.

[17] S. Taulu and J. Simola, "Spatiotemporal signal space separation method for rejecting nearby interference in MEG measurements," *Phys. Med. Biol.*, vol. 51, no. 7, pp. 1759–1768, 2006.

[18] S. Taulu, J. Simola, and M. Kajola, "MEG recordings of DC fields using the signal space separation method (SSS)," *Neurol. Clin. Neurophysiol.*, vol. 35, Nov. 2004.

[19] J. Manferdelli, N. Govindaraju, and C. Crall, "Challenges and opportunities in many-core computing," *Proc. IEEE*, vol. 96, no. 5, pp. 808–815, May 2008.

[20] J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, and J. Phillips, "GPU computing," *Proc. IEEE*, vol. 96, no. 5, pp. 879–899, May 2008.

[21] J. Wikswo and K. Swinney, "A comparison of scalar multipole expansions," *J. Appl. Phys.*, vol. 56, no. 11, pp. 3039–3049, Dec. 1984.

[22] D. O'Leary, "Robust regression computation using iteratively reweighted least squares," *SIAM J. Matrix Anal. Appl.*, vol. 11, no. 3, pp. 466–480, Jul. 1990.

[23] T. Song, K. Gaa, L. Cui, L. Feffer, R. Lee, and M. Huang, "Evaluation of signal space separation via simulation," *Med. Biol. Eng. Comput.*, vol. 46, no. 9, pp. 923–932, Sep. 2008.

[24] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw, "BCI2000: a general-purpose brain-computer interface (BCI) system," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004.

[25] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. Hoboken, NJ: Wiley, 1987.

[26] G. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.

[27] J. Jackson, *Classical Electrodynamics*. Hoboken, NJ: Wiley, 1999.

Chenlei Guo received the B.S. and M.S. degrees in electronic engineering from Fudan University, Shanghai, China, in 2005 and 2008, respectively.

He is currently at Microsoft Corporation, Redmond, WA, USA. His research interest includes modeling biologically plausible computational visual attention, object detection/recognition, and brain–computer interface.

Xin Li (S'01–M'06) received the B.S. and M.S. degrees in electronics engineering from Fudan University, Shanghai, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2005.

In 2005, he cofounded Xigmix Inc. to commercialize his Ph.D. research, and was the Chief Technical Officer until the company was acquired in 2007. He is currently an Assistant Research Professor in the Department of Electrical and Computer Engineering, Carnegie Mellon University. Since 2009, he has the Assistant Director for FCRP Focus Research Center for Circuit and System Solutions (C2 S2), a national consortium of 13 research universities (Carnegie Mellon University, Massachusetts Institute of Technology, Stanford University, University of California, Berkeley, University of Illinois at Urbana-Champaign, University of Michigan, Columbia University, University of California, Los Angeles, among others) chartered by the U.S. semiconductor industry and the U.S. Department of Defense to work on next-generation IC design challenges. His research interests include IC design and neural signal processing.

Dr. Li was in the Technical Program Committee of International Conference on Computer-Aided Design during 2008 and 2009, the Technical Program Committee of International Conference on Very-large-scale integration Design in 2009, the Technical Program Committee of International Conference on Image Theory and Applications in 2009, and the IEEE Outstanding Young Author Award Selection Committee in 2006. He received the Best Session Award from Semiconductor Research Corporation Student Symposium in 2006, the Best Paper Nomination from Design Automatic Conference in 2006, and the IEEE/ACM William J. McCalla ICCAD Best Paper Award in 2004. He also received the Inventor Recognition Awards from Focus Center Research Program in 2006, 2007, and 2009.

Samu Taulu received the M.S. degree in technical physics, in 2000, and the Ph.D. degree in biomedical engineering and computational science, in 2008, both from Helsinki University of Technology (HUT), Espoo, Finland.

He was at Brain Research Unit, Low Temperature Laborator, HUT. He is currently a Senior Researcher at Elekta Neuromag Oy, Helsinki, Finland, where he is involved in mathematical method development and project management for magnetoencephalography (MEG) applications and software. He is engaged in research as a Scientist in collaboration with the customers and other collaborators of Elekta Neuromag Oy, and also delivers lectures on applications of electromagnetic measurements at HUT (called Aalto University since January 2010). The method developed by him and his colleagues at Elekta Neuromag Oy includes the signal space separation method, which was initiated in 2001 by the challenges of signal processing of the infant MEG measurements that were carried out in the BioMag Laboratory, Helsinki University Central Hospital, Finland. His research interests include signal processing of biomagnetic multichannel data, and modeling of MEG fields with applications in neural current estimation and instrumentation design.

Dr. Taulu has been a member of the International Society for Advanced Clinical MEG, since 2009. He received the Excellence Award at the 2005 MEG Applications Conference.

Wei Wang received the M.Sc. degree in biomedical engineering from the University of Tennessee Health Science Center, Memphis, TN, in 2002, the M.D. degree from Peking University Health Science Center (formerly Beijing Medical University), Beijing, China, in 1999, and the Ph.D. degree in Biomedical Engineering from Washington University, St. Louis, MO, in 2006.

He was a Senior Scientist at St. Jude Medical, Inc., Sylmar, CA. He is currently an Assistant Professor in the Department of Physical Medicine and Rehabilitation and also involved in the Department of Bioengineering, University of Pittsburgh, Pittsburgh, PA. He is also a Project Leader in the National Science Foundation Quality of Life Technology Engineering Research Center, a joint entity between University of Pittsburgh and Carnegie Mellon University. His research interests include neural engineering, motor neuroprosthetics, brain–computer interface, rehabilitation of movement disorders, and motor system neurophysiology.

Douglas J. Weber (M'94) received the B.S. degree in biomedical engineering from the Milwaukee School of Engineering, Milwaukee, WI, in 1994, and the M.S. and Ph.D. degrees in bioengineering from Arizona State University, Tempe, in 2000 and 2001, respectively.

He was a Postdoctoral Fellow (2001–2003) and an Assistant Professor (2003–2005) in the Center for Neuroscience, University of Alberta, Alberta, Canada. He is currently an Assistant Professor in the Department of Physical Medicine and Rehabilitation, University of Pittsburgh, Pittsburgh, PA. He is also a Faculty Member in the Department of Bioengineering and the Center for the Neural Basis of Cognition, University of Pittsburgh. His research interests include neural engineering, including studies of motor learning and control of walking and reaching with particular emphasis on applications to rehabilitation technologies and practice, functional electrical stimulation, activity-based neuromotor rehabilitation, neural coding, and neural control of prosthetic devices. The research in his laboratory has been concerned with development of somatosensory neural interfaces to record from or stimulate primary afferent neurons in cats and humans, and brain machine interface studies with magnetoencephalography and electrocorticography in humans. His research is supported by the grants from the National Institute of Biomedical Imaging and Bioengineering, the National Institute of Neurological Diseases and Stroke, and the U.S. Army's Telemedicine and Advanced Technology Research Center.

Dr. Weber has been a member of IEEE Engineering in Medicine and Biology Society, since 1994 and a member of the Society for Neuroscience, since 1995.