Hindawi Publishing Corporation
Advances in Mathematical Physics
Volume 2015, Article ID 472818, 7 pages
http://dx.doi.org/10.1155/2015/472818

Research Article

Local System Matrix Compression for Efficient Reconstruction in Magnetic Particle Imaging

T. Knopp[1,2] and A. Weber[3]

[1] *Section for Biomedical Imaging, University Medical Center Hamburg-Eppendorf, 20246 Hamburg, Germany*
[2] *University of Technology, 21073 Hamburg, Germany*
[3] *Bruker Biospin MRI, 76275 Ettlingen, Germany*

Correspondence should be addressed to T. Knopp; t.knopp@uke.de

Received 15 October 2014; Revised 11 December 2014; Accepted 11 December 2014

Academic Editor: Xiao-Jun Yang

Magnetic particle imaging (MPI) is a quantitative method for determining the spatial distribution of magnetic nanoparticles, which can be used as tracers for cardiovascular imaging. For reconstructing a spatial map of the particle distribution, the system matrix describing the magnetic particle imaging equation has to be known. Due to the complex dynamic behavior of the magnetic particles, the system matrix is commonly measured in a calibration procedure. In order to speed up the reconstruction process, recently, a matrix compression technique has been proposed that makes use of a basis transformation in order to compress the MPI system matrix. By thresholding the resulting matrix and storing the remaining entries in compressed row storage format, only a fraction of the data has to be processed when reconstructing the particle distribution. In the present work, it is shown that the image quality of the algorithm can be considerably improved by using a local threshold for each matrix row instead of a global threshold for the entire system matrix.

1. Introduction

The quantitative imaging method magnetic particle imaging (MPI) allows resolving the spatial distribution of superparamagnetic iron oxide (SPIO) nanoparticles at high spatiotemporal resolution and high sensitivity [1–3]. Introduced in 2005, MPI experienced a rapid development leading to first preclinical results in 2009 revealing structures of the cardiovascular system of a living mouse [4].

While the MPI hardware is fast enough to acquire more than 40 frames per second for scanning volumes of about $1 \times 2 \times 2\,\mathrm{cm}^3$, only little effort has been investigated into the question of how to reconstruct the particle distribution with similar temporal performance. One major challenging part during image reconstruction is to handle the memory requirements of the system matrix of the linear system that has to be solved. Due to an insufficient understanding of the particle physics, the system matrix is usually measured in a calibration procedure and stored on permanent memory for further processing. For a 3D MPI experiment with an image size of 32^3, the size of the system matrix is about 40 GB in double precision. Although advanced workstations can be equipped today with 64 GB and more, processing such amounts of memory can be a computational bottleneck. For instance, loading a 40 GB matrix from the permanent into the main memory takes about 7 minutes assuming a disk reading rate of 100 MB/s. When increasing the image size and in turn the density of the sampling trajectory, the size of the system matrix increases quadratically such that for 64^3 image voxels more than 3 TB of main memory are required. While in current MPI reconstruction methods [5], only less than half of the full system is actually used and in turn is loaded into main memory, with better scanner hardware, the noise in the measured system matrix will be reduced so that more matrix rows have sufficient SNR and can be considered for reconstruction.

Recently, a matrix compression technique has been proposed, which allows for performing in-memory reconstruction even for large image sizes [6]. To this end, a matrix compression technique is used, which transforms the rows of the system matrix into a certain basis, yielding a sparse coefficient matrix. This matrix can then be compressed by applying a threshold and storing only the significant components. Such matrix compression techniques are well known in the literature, where commonly a wavelet transformation is used [7].

In [6], a global threshold has been used for the reduction of the system matrix. Without verification, it has been assumed that global thresholding is better than local thresholding. The contribution of the present paper is to show that the opposite is true. By applying a local threshold to each matrix row individually, the compression rate can be considerably increased while maintaining the image quality.

Besides discussing the impact of the thresholding strategy, we will show in this work that modest thresholding can even improve the image quality compared to a regular reconstruction without matrix compression. This is based on the noise reduction that results from truncation of coefficients which contain only noise.

2. Preliminaries

Magnetic particle imaging applies various static and oscillating magnetic fields for spatial encoding. Using a set of usually L receive coils, the change of the particles' magnetization progression is measured during the time interval $[0,T)$, where T is the repetition time of the experiment. The induced voltages are given by $u_l(t)$, $l = 0,\dots,L-1$. As the induced voltages $u_l(t)$ are in practice covered by excitation signals $u_l^E(t)$ directly coupling into the receive coils, the measurement signals are usually considered in frequency space, where the excitation signal can be removed by a narrow band-stop filter. The remaining frequency components

$$\hat{u}_{l,k} = \frac{1}{T}\int_0^T u_l(t)\, e^{-2\pi i k t/T} \mathrm{d}t, \quad k = 0,\dots,K-1 \tag{1}$$

contain only the signal induced by the magnetic nanoparticles. Due to finite sampling of the time signal, only K discrete Fourier coefficients are available in practice. To store the frequency components of all receive channels in a 1D vector, one can introduce the mapping $\hat{u}_j := \hat{u}_{l,k}$, if $j = lK + k$. The total number of frequency components is thus given by $M = KL$.

3. Imaging Equation

The aim of MPI is to reconstruct the particle concentration c at N discrete positions $\mathbf{r}_n$, $n = 0,\dots,N-1$, which can lay on a 1D, 2D, or 3D sampling grid. The relation between the particle concentration $c_n := c(\mathbf{r}_n)$ and the measured frequency component $\hat{u}_j$ can be expressed by a linear system of equations

$$\hat{u}_j = \sum_{n=0}^{N-1} s_{j,n} c_n, \quad j = 0,\dots,M-1, \tag{2}$$

where $s_{j,n}$ describes the contribution of particles located at position $\mathbf{r}_n$ to the measured frequency components $\hat{u}_j$. In continuous form, $s_j(\mathbf{r}_n) = s_{j,n}$ is called the MPI system function. In matrix-vector notation, (2) can be expressed as

$$\hat{\mathbf{u}} = \mathbf{S}\mathbf{c}, \tag{3}$$

where $\hat{\mathbf{u}}$ is the measurement vector, $\mathbf{c}$ is the particle concentration vector, and $\mathbf{S}$ is the system matrix. The rows of the system matrix are denoted by $\mathbf{s}_j$, $j = 0,\dots,M-1$, which we assume to be column vectors in this work.

4. Matrix Compression

As it has been discussed in [8, 9], the individual rows of the system matrix consist of wave-like functions bearing high similarity to tensor products of Chebyshev polynomials. But as the relationship is only qualitative, one cannot exploit this structure of the system matrix directly. Instead, as it has been proposed by [6], one can use matrix compression techniques to reduce the size of the system matrix, which accelerates the multiplications with $\mathbf{S}$ considerably. The basic idea has already been used for general matrices in the context of the wavelet transformation.

In the discrete setting, we consider a set of N linear independent basis vectors $\mathbf{b}_i$, $i = 0,\dots,N-1$, which set up a transformation matrix $\mathbf{B} = (\mathbf{b}_0,\dots,\mathbf{b}_{N-1})$. In order to expand the jth matrix row $\mathbf{s}_j$ into the basis vectors, one has to solve the linear system

$$\mathbf{B}\widetilde{\mathbf{s}}_j = \mathbf{s}_j, \tag{4}$$

where $\widetilde{\mathbf{s}}_j$ are the basis coefficients. If the transformation matrix $\mathbf{B}$ is orthogonal, the coefficients can be computed by

$$\widetilde{\mathbf{s}}_j = \widetilde{\mathbf{B}}\mathbf{s}_j, \tag{5}$$

where $\widetilde{\mathbf{B}} = \mathbf{B}^{\mathsf{H}}$. When considering all matrix rows $\mathbf{s}_j$, $j = 0,\dots,M-1$, (5) can be written as

$$\widetilde{\mathbf{S}}^{\mathsf{T}} = \widetilde{\mathbf{B}}\mathbf{S}^{\mathsf{T}}, \quad \text{or} \quad \widetilde{\mathbf{S}} = \mathbf{S}\widetilde{\mathbf{B}}^{\mathsf{T}}, \tag{6}$$

where $\widetilde{\mathbf{S}} = (\widetilde{\mathbf{s}}_0,\dots,\widetilde{\mathbf{s}}_{M-1})^{\mathsf{T}}$. By introducing a coefficient vector $\widetilde{\mathbf{c}}$ satisfying

$$\widetilde{\mathbf{B}}^{\mathsf{T}}\widetilde{\mathbf{c}} = \mathbf{c}, \tag{7}$$

the original linear system (3) can be reformulated as

$$\hat{\mathbf{u}} = \mathbf{S}\mathbf{c} = \mathbf{S}\widetilde{\mathbf{B}}^{\mathsf{T}}\widetilde{\mathbf{c}} = \widetilde{\mathbf{S}}\widetilde{\mathbf{c}}. \tag{8}$$

Instead of solving (3) directly, one thus can perform the following steps.

(1) Perform the basis transformation $\widetilde{\mathbf{S}} = \mathbf{S}\widetilde{\mathbf{B}}^{\mathsf{T}}$.

(2) Solve the linear system $\hat{\mathbf{u}} = \widetilde{\mathbf{S}}\widetilde{\mathbf{c}}$.

(3) Compute the solution $\mathbf{c} = \widetilde{\mathbf{B}}^{\mathsf{T}}\widetilde{\mathbf{c}}$.

While on a first sight these three steps seem to require more computational effort than solving (3), they actually can be carried out much more efficiently, which is substantiated by the following facts. First of all, the basis transformation of the system matrix has to be performed only once. In fact, this operation can be directly performed after measuring the system matrix. Furthermore, although a matrix multiplication has to be performed, for the mainly considered transformations, this multiplication can be performed in an efficient manner using fast algorithms (e.g., the fast Fourier transformation (FFT) and the fast cosine transformation (FCT)). Consequently, also the postprocessing step $\mathbf{c} = \widetilde{\mathbf{B}}^{\mathsf{T}}\widetilde{\mathbf{c}}$ can be performed very efficiently by applying a fast basis transformation.

The main reason for expanding the system matrix rows into basis functions is that the second step, the solution of the linear system $\hat{\mathbf{u}} = \widetilde{\mathbf{S}}\widetilde{\mathbf{c}}$, can be considerably accelerated by exploiting the sparse structure of the transformed matrix $\widetilde{\mathbf{S}}$. Hence, by applying a threshold and taking only few entries of $\widetilde{\mathbf{S}}$ into account, arithmetic operations involving the system matrix can be considerably accelerated. This is essential for iterative reconstruction algorithms, which, for instance, apply matrix-vector multiplications with the system matrix and its adjoint. Even more important is that the full MPI system matrix will not fit into the main memory of the reconstruction computer for practical voxel resolutions. Hence, matrix compression can enable in-memory reconstruction, which would not be possible when considering the full representation of the system matrix.

5. Selection of Matrix Rows

At this point, it should be noted that even for the conventional MPI reconstruction involving the solution of the linear system (3) usually only parts of the system matrix are considered. More precisely, the energy of the system matrix rows and in turn the energy of the measurements $\hat{\mathbf{u}}$ considerably vary in dependence of the time frequency. As $\mathbf{S}$ is usually measured in a calibration procedure, there are several matrix rows that only contain noise. Therefore, as it has been discussed in [12], for reconstruction of MPI data, only the matrix rows with sufficient SNR are taken into account. The set of row indices to be used for reconstruction can be obtained by calculating the SNR of $\mathbf{s}_j$ in dependence of frequency and applying a threshold.

Considering the sparse system matrix $\widetilde{\mathbf{S}}$, one can apply the same row thresholding as for the dense system matrix $\mathbf{S}$. Matrix rows containing only noise will not be compressed by the basis transformation and in turn these rows should not be taken into account.

6. Matrix Thresholding

Besides selecting the threshold that determines the set of used matrix rows, one has to choose for each matrix row the number of coefficients (i.e., columns) that are used for reconstruction. In this work, we propose to use a fixed number of coefficients per matrix row. This is contrary to the procedure proposed by [6], where a global threshold was applied to the matrix $\widetilde{\mathbf{S}}$. A global threshold implies that more coefficients of matrix rows with high energy are used than those of matrix rows with low energy. But as the details (i.e., the high-resolution part) of the particle distribution are encoded in matrix rows having low energy [10], using a global threshold leads to a loss of spatial resolution, which is experimentally proven in Section 8.

Note that for a fair comparison between local thresholding and global thresholding we apply the threshold only to those matrix rows that are selected by the row selection algorithm described in Section 5.

7. Materials and Methods

Experiments have been carried out using the MPI scanner published in [4], which has a bore diameter of 32 mm and is capable of imaging mice. The selection field has a gradient strength of $5.5\,\mathrm{T m^{-1}}\,\mu_0^{-1}$ while the drive fields have an amplitude of $18\,\mathrm{mT}\,\mu_0^{-1}$. A 2D Lissajous-type sampling sequence is performed using the two drive-field frequencies $f_x = 2.5\,\mathrm{MHz}/96 \approx 26041.7\,\mathrm{kHz}$ and $f_y = 2.5\,\mathrm{MHz}/99 \approx 25252.5\,\mathrm{kHz}$. These correspond to a commensurable frequency ratio of 33/32, determining the density of the sampling trajectory. The resulting drive-field cycle is $T = 1.26$ ms. By averaging 17 sequential frames, the SNR of the measured data is improved, leading to a total acquisition time of 21.5 ms per frame. The acquired MPI signals are expanded into Fourier series while considering frequencies up to 1 MHz. Hence, the total number of frequency components per receive channel is 1268. After combining the frequency components of the two receive channels, the total number of entries of the measurement vector $\hat{\mathbf{u}}$ is 2536.

The system matrix is acquired using a calibration measurement using a cubic delta sample with an edge length of 0.6 mm filled with undiluted Resovist (0.5 mol (Fe) L^{-1}). The sample is shifted to positions on a regular grid of size 68×40 covering a FOV of $20.4 \times 12.0\,\mathrm{mm}^2$. Regular measurements are acquired using a phantom consisting of 12 dots forming the letter P. The phantom is rotated by about 45° in a dynamic sequence (see Figure 1). The data has also been used in [9, 12].

For the compression of the MPI system matrix, we use the discrete cosine transform (DCT) that already yielded excellent results in [6]. Only matrix rows with a signal-to-noise ratio larger than 10 are used for reconstruction leading to a total number of 1400 rows.

Different thresholds are applied to the remaining matrix such that the amount of stored data is between 0.2% and 5% of the total number of matrix entries. Besides using a fixed number of coefficients per row, we compared this approach to a global threshold for reduction of the system matrix, which has been proposed in [6].

For image reconstruction, the iterative Kaczmarz method [13] is used considering 5 iterations and a regularization parameter, which is chosen such that the best visual result is obtained (see [12]).

All algorithms are implemented using the Julia programming language [14, 15], which allows us to formulate algorithms at a high abstraction level (similar to MATLAB) but run the programs in native C-speed.

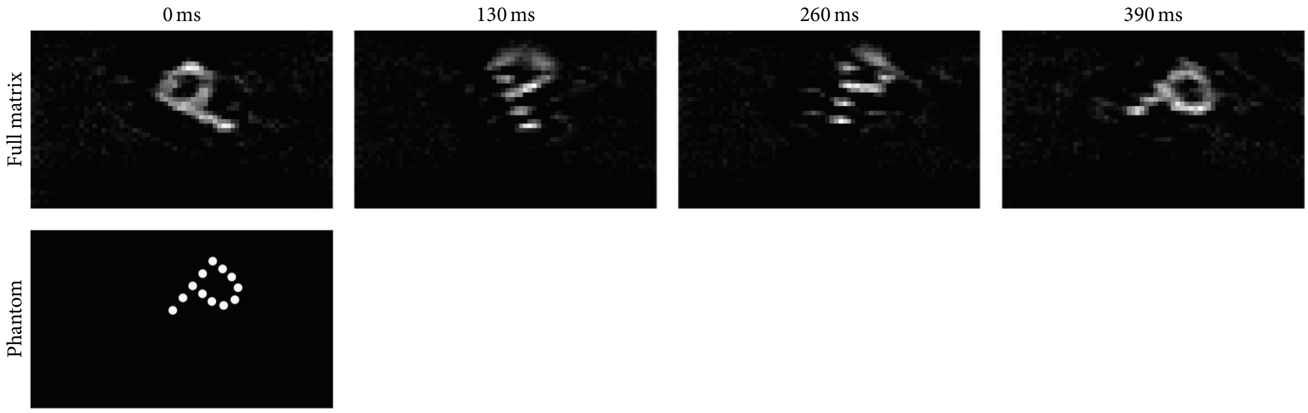


Figure 1: Reconstructed MPI data at four selected time points using the full system matrix without basis transformation. For comparison, a sketch of the used phantom is shown.

8. Results

As a reference for the proposed sparse reconstruction framework using local and global thresholding, we use the common MPI reconstruction method, which uses the dense representation of the system matrix. The reconstruction results of the phantom data are shown in Figure 1. As one can see, the vertical resolution of the images is better than the horizontal direction. This is due to the gradient strength of the selection field being 5.5 $\mathrm{Tm}^{-1}\,\mu_0^{-1}$ in vertical direction and 2.75 $\mathrm{Tm}^{-1}\,\mu_0^{-1}$ in horizontal direction.

In Figure 2, the results of the sparse reconstruction algorithm outlined in Section 6 are shown for the DCT basis transformation and local as well as global thresholding. When considering 5% of the full system matrix for reconstruction, the results using the local and the global threshold show equivalent image quality. Interestingly, compared to the reference images using the full system matrix, the images are even slightly less noisy while retaining the spatial resolution (see also Figure 3). This can be explained by the fact that the removed basis coefficients mainly contain noise. Removing these coefficients thus can potentially even improve the image quality.

When reducing the number of used basis coefficients further, the image quality degrades as the removed coefficients then not only contain noise but also describe the shape of the MPI system function. Using a global threshold, the image quality degrades when reducing the number of coefficients to 1% of the total number of coefficients so that the dots in vertical direction can hardly be resolved when using the DCT as a basis transformation. In contrast, for the local threshold, the dots in vertical direction can even be resolved when only 0.5% of the total number of basis coefficients is used.

9. Discussion

In the present work, we have investigated the sparse reconstruction framework for magnetic particle imaging that allows the processing of large MPI datasets for which the full system matrix would not fit into the main memory of the reconstruction computer.

We have shown that one can achieve better image quality when thresholding the basis transformed MPI system matrix locally on each matrix row instead of globally on the complete matrix. This can be explained by the fact that the energy of the MPI system matrix rows varies considerably. When applying a global threshold, a large number of rows fall below the threshold while from individual rows with high energy more basis coefficients are taken than actually needed for an accurate approximation. Using the proposed local thresholding strategy, from each row, the same number of coefficients is taken which makes the method invariant of the row energy. In turn, even the low energy rows of the MPI system function carrying the high spatial frequency information of the particle distribution are present in the approximated system matrix. In order to show this effect, in Figure 4 one particular matrix row is shown for different compression rates and global as well as local thresholding. At 0.2% compression rate, the entire matrix row falls under the threshold when using the global thresholding strategy, while the local thresholding strategy still reveals the wave pattern of the original matrix row.

It should be noted that one reason that the local threshold performs better than the global threshold is that the SNR of the receiver is not constant over frequency. Typically the MPI-receive chain (filter and analog receive amplifier) is implemented in such a way that frequencies that have a lower MPI signal are amplified more than those where the MPI signal is stronger. This allows one to detect more frequency components than one would capture when a constant amplification over frequency would be used.

For the acquisition of larger scanning volumes, MPI will rely on a multipatch method where small volumes are acquired using a fast 3D drive-field sequence and the small measuring field is shifted in space over larger distances using the so-called focus field [11]. Due to field inhomogeneity, the system matrices for different focus-field positions will differ slightly such that for optimal image quality no single system

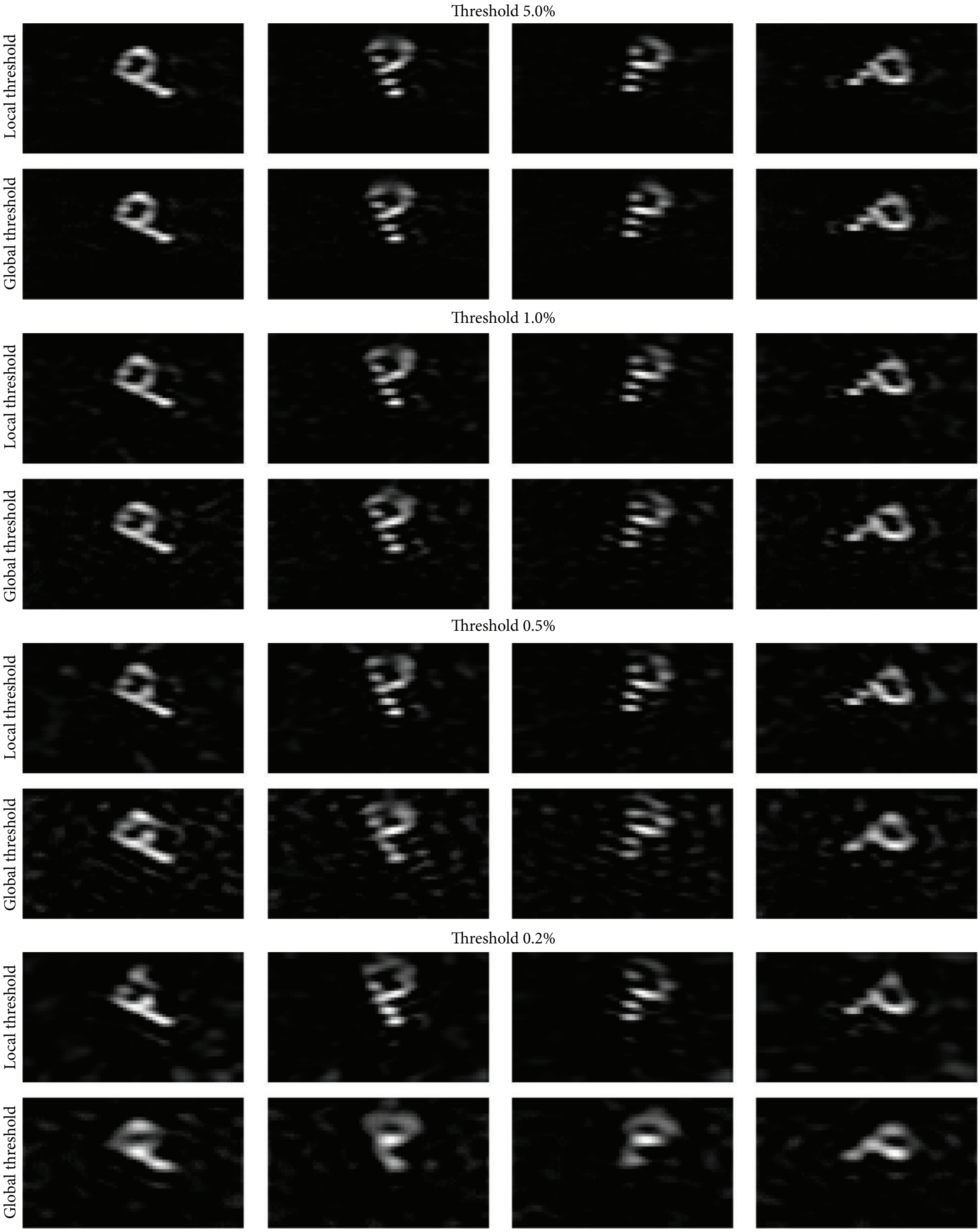


FIGURE 2: Reconstructed MPI data at four selected time points using the sparse reconstruction algorithm with a DCT basis transformation using a local and a global threshold for matrix compression. The compression rates are between 0.2% and 5%.

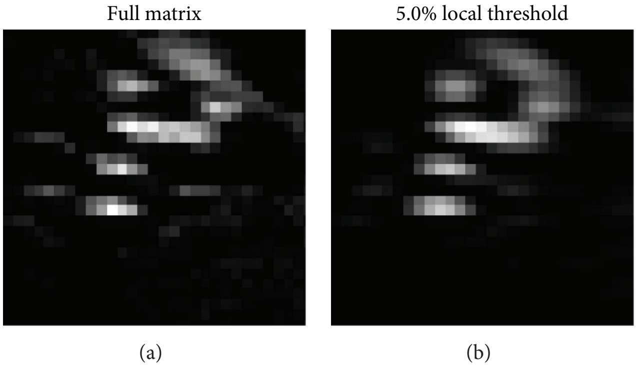


Figure 3: Reconstructed MPI data using the conventional (a) and the sparse (b) reconstruction technique considering a compression rate of 5% and local thresholding.

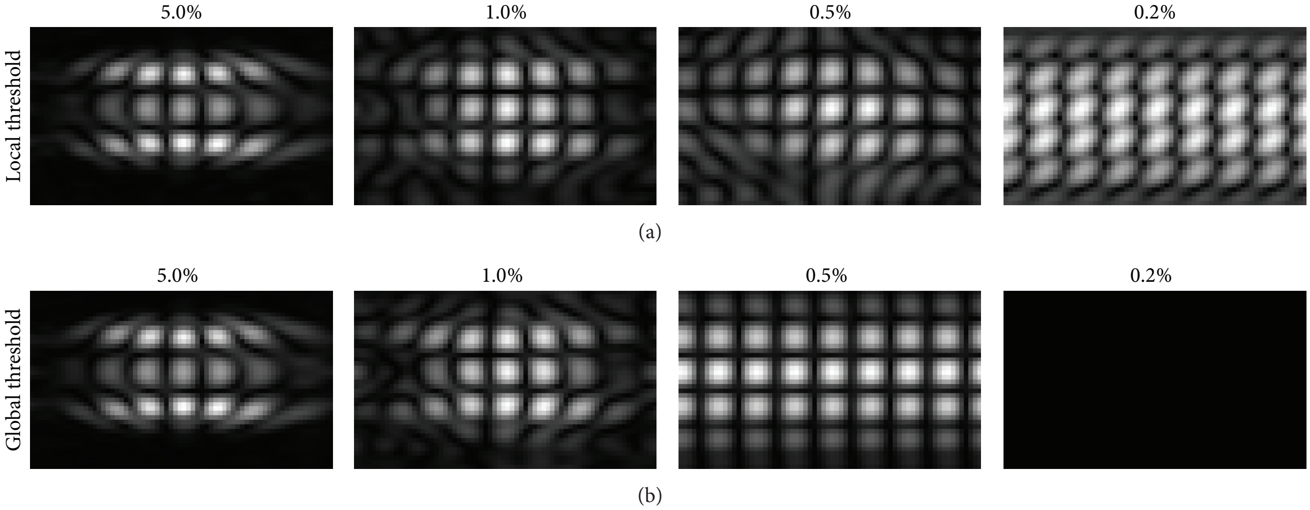


Figure 4: The 156th matrix row (about 123 kHz) of receive channel x after applying a local (a) and a global (b) threshold for different compression rates between 0.2% and 5%. Note that, for 0.2% compression rate, no coefficients are above the threshold for the global thresholding strategy.

matrix can be used for all focus-field patches. In this case, the matrix compression technique with local thresholding discussed in this paper can be applied to each system matrix individually, which will considerably improve the reconstruction performance for multipatch focus-field data.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] B. Gleich and J. Weizenecker, “Tomographic imaging using the nonlinear response of magnetic particles,” *Nature*, vol. 435, no. 7046, pp. 1214–1217, 2005.

[2] P. W. Goodwill and S. M. Conolly, “The X-space formulation of the magnetic particle imaging process: 1-D signal, resolution, bandwidth, SNR, SAR, and magnetostimulation,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 11, pp. 1851–1859, 2010.

[3] J. B. Weaver, A. M. Rauwerdink, C. R. Sullivan, and I. Baker, “Frequency distribution of the nanoparticle magnetization in the presence of a static as well as a harmonic magnetic field,” *Medical Physics*, vol. 35, no. 5, pp. 1988–1994, 2008.

[4] J. Weizenecker, B. Gleich, J. Rahmer, H. Dahnke, and J. Borgert, “Three-dimensional real-time *in vivo* magnetic particle imaging,” *Physics in Medicine and Biology*, vol. 54, no. 5, p. -L10, 2009.

[5] T. Knopp and A. Weber, “Sparse reconstruction of the magnetic particle imaging system matrix,” *IEEE Transactions on Medical Imaging*, vol. 32, no. 8, pp. 1473–1480, 2013.

[6] J. Lampe, C. Bassoy, J. Rahmer et al., “Fast reconstruction in magnetic particle imaging,” *Physics in Medicine and Biology*, vol. 57, no. 4, pp. 1113–1134, 2012.

[7] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, New York, NY, USA, 2nd edition, 1992.

[8] J. Rahmer, J. Weizenecker, B. Gleich, and J. Borgert, “Signal encoding in magnetic particle imaging: properties of the system function,” *BMC Medical Imaging*, vol. 9, article 4, 2009.

[9] T. Knopp, S. Biederer, T. F. Sattel et al., "2D model-based reconstruction for magnetic particle imaging," *Medical Physics*, vol. 37, no. 2, pp. 485–491, 2010.

[10] T. Knopp and T. Buzug, *Magnetic Particle Imaging: An Introduction to Imaging Principles and Scanner Instrumentation*, Springer, Berlin, Germany, 2012, http://www.springer.com/medicine/radiology/book/978-3-642-04198-3.

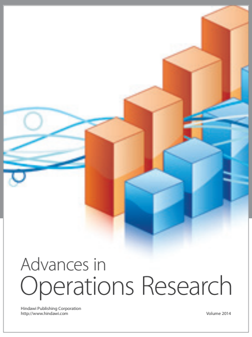
[11] B. Gleich, J. Weizenecker, H. Timminger et al., "Fast MPI demonstrator with enlarged field of view," in *Proceedings of the International Society for Magnetic Resonance in Medicine*, vol. 18, p. 218, Stockholm, Sweden, May 2010.

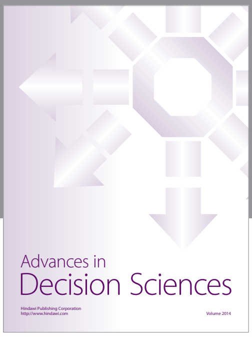
[12] T. Knopp, J. Rahmer, T. F. Sattel et al., "Weighted iterative reconstruction for magnetic particle imaging," *Physics in Medicine and Biology*, vol. 55, no. 6, pp. 1577–1589, 2010.

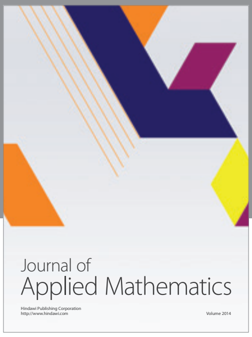
[13] S. Kaczmarz, "Angenäherte Auflösung von Systemen linearer Gleichungen," *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, vol. 35, pp. 355–357, 1937.

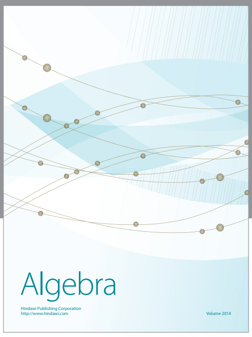
[14] J. Bezanson, S. Karpinski, V. Shah, and A. Edelman, "Julia: A fast dynamic language for technical computing," CoRR, abs/1209.5145, 2012.

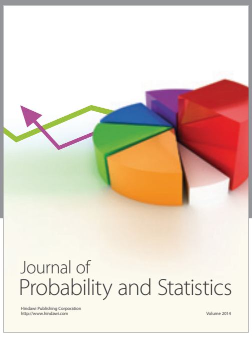
[15] J. Bezanson, J. Chen, S. Karpinski, V. Shah, and A. Edelman, "Array operators using multiple dispatch: a design methodology for array implementations in dynamic languages," in *Proceedings of the ACM SIGPLAN International Workshop on Libraries, Languages, and Compilers for Array Programming (ARRAY '14)*, p. 56, Edinburgh, UK, June 2014.


Advances in
Operations Research

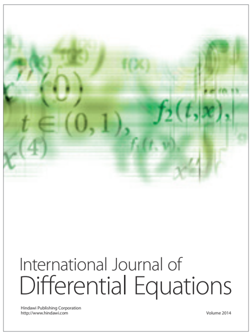

Advances in
Decision Sciences


Journal of
Applied Mathematics

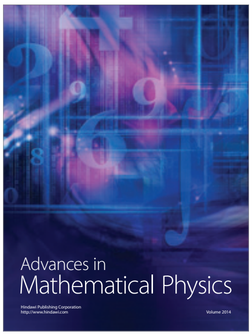

Algebra

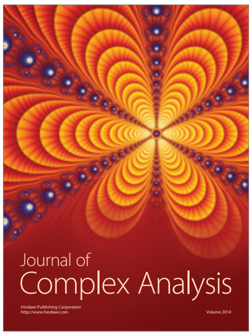

Journal of
Probability and Statistics

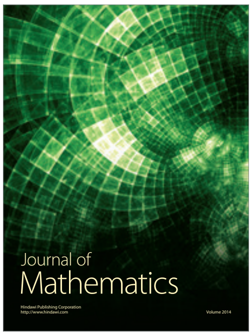
The Scientific
World Journal

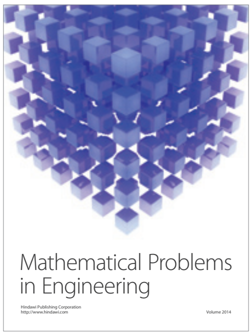

International Journal of
Differential Equations

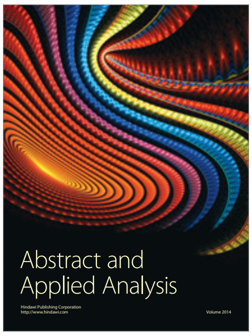
International Journal of
Combinatorics

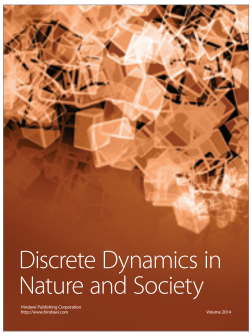

Advances in
Mathematical Physics

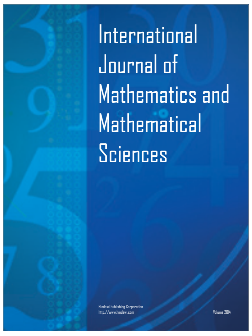

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

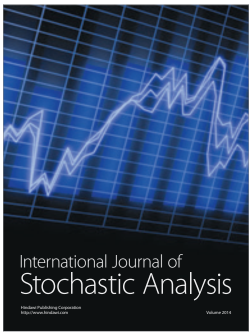

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization