Published in final edited form as:

J Neurosci Methods. 2013 September 15; 218(2): 184–195. doi:10.1016/j.jneumeth.2013.04.015.

High-throughput optogenetic functional magnetic resonance imaging with parallel computations

Zhongnan Fang¹ and Jin Hyung Lee^{1,2,3,*}

¹Department of Electrical Engineering, University of California, Los Angeles, CA 90095, USA

²Department of Neurology and Neurological Sciences, Department of Bioengineering, Department of Neurosurgery, Stanford University, CA 94305

³Department of Psychiatry and Biobehavioral Sciences, Department of Radiology, Neuroscience, and Biomedical Engineering Interdepartmental Program, University of California, Los Angeles, CA 90095, USA

Abstract

Optogenetic functional magnetic resonance imaging (ofMRI) technology enables cell-type specific, temporally precise neuronal control and accurate, *in vivo* readout of resulting activity across the whole brain. With the ability to precisely control excitation and inhibition parameters, and to accurately record the resulting activity, there is an increased need for a high-throughput method to bring the ofMRI studies to their full potential. In this paper, an advanced system that can allow real-time fMRI with interactive control and analysis in a fraction of the MRI acquisition repetition time (TR) is proposed. With such high processing speed, sufficient time will be available for integration of future developments that can further enhance ofMRI data quality or better streamline the study. We designed and implemented a highly optimized, massively parallel system using graphics processing unit (GPU)s which achieves reconstruction, motion correction, and analysis of 3D volume data in approximately 12.80 ms. As a result, with a 750 ms TR and 4 interleaved fMRI acquisition, we can now conduct sliding window reconstruction, motion correction, analysis and display in approximately 1.7% of the TR. Therefore, a significant amount of time can now be allocated to integrating advanced but computationally intensive methods that can enable higher image quality and better analysis results all within a TR. Utilizing the proposed high-throughput imaging platform with sliding window reconstruction, we were also able to observe the much-debated initial dips in our ofMRI data. Combined with methods to further improve SNR, the proposed system will enable efficient real-time, interactive, high-throughput ofMRI studies.

Keywords

optogenetics; ofMRI; parallel; GPU; motion correction; initial dip

© 2013 Elsevier B.V. All rights reserved.

*Corresponding Author: Jin Hyung Lee, PhD, ljinhy@stanford.edu, 1201 Welch Road, #P206, Stanford, CA 94305.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

1. Introduction

The ofMRI technology enables systematic probing of brain circuitry with cell-type specific and temporally accurate control, while *in vivo* monitoring of whole brain network response (Lee, 2012, 2011; Lee et al., 2010). The ability to control neurons with high specificity combined with accurate readout reflecting neural activity location and temporal firing patterns, provides an unprecedented opportunity to understand the whole brain neural network function. However, the increased degree of freedom in control (Fig. 1) and accurate readout calls for a high-throughput method that can accelerate discoveries using ofMRI. To enable such a process with high fidelity and to provide potential for future integration of more advanced methods to further improve ofMRI image quality, and to more efficiently streamline ofMRI studies, we propose a GPU based parallel high-speed system that enables data reconstruction, motion correction, and analysis for a 3D volume in approximately 12.80 ms. With such high speed, the remaining time within a MRI acquisition repetition time (TR) can be used to integrate techniques such as iterative reconstruction (Fessler, 2007) for higher image quality, automatic segmentation (Lee et al., 2008b), anatomy/atlas registration, and brain connectivity analysis. Moreover, the high processing speed will increase the robustness of the studies, which can help the system recover swiftly from possible operating system scheduling and network delays.

Since Cox et al (Cox et al., 1995) first published a real-time fMRI (rtfMRI) cumulative correlation analysis method in 1995, many different aspects of rtfMRI has been explored, e.g. real-time analysis (Bagarinao et al., 2003; Esposito et al., 2003; Gembris et al., 2000), real-time motion correction (Cox and Jesmanowicz, 1999), and real-time applications such as brain machine interface and clinical diagnosis (Caria et al., 2011; Cohen, 2001; deCharms, 2008; Lee et al., 2009; Voyvodic, 1999; Weiskopf et al., 2004). Most of these widely used rtfMRI techniques are designed to reconstruct and analyze fMRI images after a complete 3D volume acquisition with a relatively long and the response time requirement. However, considering future integration of advanced but usually computationally intensive techniques for the ofMRI studies to improve image quality and efficiency, we sought to further increase processing speed.

Real-time motion correction is also a critical part of a high-throughput interactive fMRI system. Because motion correction is usually an iterative process, most of the current algorithms are designed for offline processing (e.g. AIR, FSL and SPM (Friston et al., 1995; Jenkinson et al., 2002; Woods et al., 1992)). AFNI (Cox and Jesmanowicz, 1999) offers real-time motion correction at an approximately 51.31 ms/volume speed. We seek to achieve even higher speed motion correction in order to optimize for future integration with computationally intensive processing.

In recent years, the GPU, which is rapidly evolving for massively parallel computations and devotes more of its transistors to computation than CPUs do, is showing increasing potential for high-throughput rtfMRI systems. Many remarkable speedups by GPUs are reported (Ansorge et al., 2009; Eklund et al., 2010; Huang et al., 2011; Ruijters et al., 2008; Shams et al., 2010; Stone et al., 2008). Encouraged by these successful results, we designed and optimized a series of new parallel algorithms for the GPU platform.

With the proposed system, ofMRI studies can potentially be conducted with high efficiency: optogenetic modulation parameters such as stimulation frequency, wavelength, power and pulse width can be controlled based on live and accurate feedback of each stimulation's impact across the whole brain. Tested on averaged high SNR phantom and ofMRI datasets, robust performance with high speed and accuracy was achieved on our system. With the sliding window reconstruction (Nayak et al., 2004; Riederer et al., 1988) in our proposed system, higher effective temporal resolution is achieved and the much-debated initial dips are also observed in our ofMRI data. Although initial dips can also be observed through offline sliding window reconstruction, the ability to perform this computationally intensive calculation in real-time clearly demonstrates the potential of our high-throughput system for future integration of advanced techniques to further improve ofMRI image quality and study throughput. In the following sections, detailed methods, analysis, results, and discussion are presented.

2. Methods

2.1. System Architecture

We implemented the proposed high-throughput ofMRI algorithms on a custom-built parallel workstation that is connected to a 7 T Bruker Biospec MRI scanner (Fig. 2). Three levels of simultaneous processing are designed on the high-throughput ofMRI workstation CPU threads. In level one, the raw data is received through a reliable Ethernet transmission protocol and preprocessed by a baseline correction method. In the second level, the pre-processed raw data is real-time reconstructed, motion corrected, and analyzed through the proposed GPU-based massively parallel algorithms. In this level, we design the CPU thread as a master process that allocates parallel computing tasks to the slave GPU. In level three, the processed real-time images are displayed with analysis results.

2.2. Image Acquisition Pulse Sequences

We utilized two types of MRI acquisition pulse sequences for our real-time scanning: a gradient recalled echo (GRE) multi-slice, interleaved-spiral readout sequence and a balanced steady state free precession (b-SSFP) three-dimensional, interleaved stack-of-spiral readout sequence (Lee et al., 2008a; Lee et al., 2003; Miller et al., 2006; Scheffler and Lehnhardt, 2003). Spiral readouts were chosen to achieve large volume coverage with high sampling speed. In addition, the interleaved acquisition provides the advantage of enabling high temporal resolution sliding window reconstruction (Meyer et al., 1992).

The GRE sequence was designed to have a TR and an echo time (TE) of $TR/TE = 750/12$ ms and a 30° flip angle. 23 four-interleaf coronal slices (11.5 mm slice direction coverage) were obtained every 3 s. The b-SSFP sequence was designed to have $TR/TE = 9.375/1.5$ ms and a flip angle of 30° . 32 ten-interleaf slices (16 mm slice direction coverage) were acquired every 3 s. The in-plane field of view (FOV) was 35×35 mm² with 0.5 mm slice thickness for both sequences. The GRE BOLD and b-SSFP data were both reconstructed into $128 \times 128 \times 23$ matrix-size images with 0.273 mm reconstructed voxel size in both x and y directions.

2.3. Reconstruction

2.3.1. Parallel Gridding—In contrast to the data-driven conventional gridding method (Jackson et al., 1991) (Fig. 3a) that conducts convolutions sample by sample, our memory conflict free gridding algorithm (Fig. 3b) calculates each Cartesian grid value in parallel using the following formula:

$$M_{grid}(m, n) = \sum_i (M_{spiral}(u_i, v_i) \cdot (w(u_i, v_i) \cdot C(m - u_i, n - v_i))) \quad (2.1)$$

subject to

$$\sqrt{(u - u_i)^2 + (v - v_i)^2} < 1.5 \quad (2.2)$$

When the reconstruction function is called, thousands of GPU threads are allocated and each thread is assigned to a k-space Cartesian grid location: (m, n) . Each thread searches through the spiral samples $M_{spiral}(u_i, v_i)$ for points that are within its convolution kernel coverage range. Eligible samples are then separately multiplied with a weighted kernel function $w(u, v)$ centered at (u_i, v_i) and gridding kernel function $C(u, v)$ centered at (m, n) , i.e. $M_{spiral}(u_i, v_i) \cdot (w(u_i, v_i) \cdot C(m - u_i, n - v_i))$. In the final step, each GPU thread will sum the re-sampled data of all eligible samples. We chose the gridding kernel to be $C(u, v) = 1 - \sqrt{u^2 + v^2}/r$ with a radius r equals to 1.5 times the grid spacing, which gives a good trade-off between speed and image quality.

As pointed out by Sorensen et al (Sorensen et al., 2008), searching through spiral samples puts a heavy overhead on the calculations. We addressed this issue by presorting the spiral samples into groups for each Cartesian grid row. In our case, because we chose kernel radius r to be 1.5 times the grid spacing, we sorted spiral samples $(M_{spiral}(u_i, v_i))$ with the row coordinates satisfying $|u_i - k| < 1.5$ into group k . We did not sort the samples for each grid because that would require a large lookup memory. In our GPU implementation, the presorted spiral sample groups are stored in the cached GPU texture memory for fast retrieval.

After gridding, the images are Fast Fourier Transformed (FFT) by the GPU NVIDIA library “CUFFT” and then rearranged by a custom-written GPU function “FFTSHIFT” that centers the image.

2.3.2. Sliding Window Reconstruction—To improve temporal resolution exploiting our fast processing speed, we designed and implemented a real-time sliding window reconstruction strategy (Fig. 3c). In a conventional reconstruction scheme, new fMRI images are reconstructed only after all interleaves are acquired for each slice. However, with the sliding window reconstruction method, the reconstruction window slides every TR (750 ms in our GRE acquisition), reconstructing images after the acquisition of each interleaf. Since our reconstruction time is far less than the TR (750 ms), we can perform even faster updates for acquisitions with smaller TR such as in the b-SSFP acquisition case, where TRs

are typically much shorter (9.375 ms in our case). For the b-SSFP sequence, we implemented updates every 10 TRs (i.e. 93.75 ms).

2.4. Motion Correction

We proposed a new motion correction method that enables fast, robust motion correction through mathematical optimization that reduces the complexity, and through GPU parallel platform design that utilized hardware features to speed up the computational speed.

2.4.1. Reducing the Complexity—We selected the Least Squared Error (LSE) as the cost function for the proposed motion correction method. LSE is reliable for intra-modal image registration, such as the continuous fMRI time-series, and is simple to calculate and enables fast computation.

We extended the 2D affine transform inverse Gauss-Newton method that was first derived by Baker et al (Baker and Matthews, 2001, 2004; Lucas and Kanade, 1981) to a 3D 6-DOF transform motion correction. It has been shown that the inverse Gauss-Newton method is able to achieve calculation cost of $O(nN)$ instead of $O(n^2N)$ that is required by regular gradient-based methods. Here, n is the degree of freedom and N is the image size. For regular gradient-based methods, the Hessian matrix H is a function of the transform matrix at each iteration. As will be seen in the following derivation, the matrix H in our proposed inverse Gauss-Newton method (Eqn. 2.6) is a constant independent of the iteration number and therefore can be calculated in advance. In our method, the main remaining computation is the calculation of p (Eqn. 2.5). The difference between the Baker's method and our proposed method is that we derived a transform matrix for 3D 6-DOF instead of a 2D affine transform.

At each iteration of the inverse Gauss-Newton method, p was determined by minimizing the cost function

$$f(\Delta p) = \sum_x [T(w(x, \Delta p)) - I(w(x, p_k))]^2, \quad (2.3)$$

where $T(x)$ is the template image and $I(x)$ is the image that is being corrected (source image). $x = [x_1 \ x_2 \ x_3]$ is the coordinate of each voxel, and $w(x, p) = [R_p x + t_p]$ is a function that transforms the voxel coordinates using 6-DOF rigid body parameter vector $p = [t_x \ t_y \ t_z \ \theta_x \ \theta_y \ \theta_z]^T$. The rotation matrix is $R_p = R_{\theta_x} R_{\theta_y} R_{\theta_z}$, and the translation vector is $t_p = [t_x \ t_y \ t_z]^T$.

Linear approximation of the cost function at $p = 0$ gives

$$f(\Delta p) = \sum_x [T(w(x, 0)) + \left(\frac{\partial w}{\partial p} \Big|_{p=0} \nabla T(w(x, 0)) \right)^T \Delta p - I(w(x, p_k))]^2, \quad (2.4)$$

$w(x, 0) = x$, $\nabla T(w(x, 0)) = \nabla T(x)$. Gradient $\nabla T(x)$ can be easily calculated using the Sobel filter. For simplicity, we will denote $w/p|_{p=0}$ as w/p . The minimum of this second order expression can be solved by Gauss-Newton algorithm, which gives

$$\Delta p = H^{-1} \sum_x \left[\frac{\partial w}{\partial p} \nabla T(x) \right] [I(w, p_k) - T(x)], \quad (2.5)$$

and

$$H = \sum_x \left(\frac{\partial w}{\partial p} \nabla T(x) \right) \left(\frac{\partial w}{\partial p} \nabla T(x) \right)^T. \quad (2.6)$$

H is the Gauss-Newton approximation of the Hessian matrix. We also show in Appendix A that w/p has a simple expression when $p = 0$:

$$\frac{\partial w}{\partial p} \Big|_{p=0} = \begin{bmatrix} 0 & x_3 & -x_2 & 1 & 0 & 0 \\ -x_3 & 0 & x_1 & 0 & 1 & 0 \\ x_2 & -x_1 & 0 & 0 & 0 & 1 \end{bmatrix}^T. \quad (2.7)$$

After the p is determined, w is *inversely* updated by setting

$$w(x, p_{k+1}) = [R_{\Delta p_1, \Delta p_2, \Delta p_3}^{-1} R_{p_k} x + (t_{p_k} - t_{\Delta p_4, \Delta p_5, \Delta p_6})]. \quad (2.8)$$

Compared to the regular Gauss-Newton method that is often used in other motion correction software packages, this inverse Gauss-Newton method is much more efficient. Each update of p depends only on the Hessian matrix H and $(w/p) \nabla T(x)$, both of which can be pre-calculated from the template image. Therefore, the calculation during iterations can be extremely fast.

2.4.2. Speedup through Parallel Computation on GPU—Fig. 4 demonstrates the implementation of our 3D inverse Gauss-Newton motion correction on a GPU. Many GPU hardware features are utilized for high computational efficiency. For example, both the template image and the source image are stored in the GPU texture memory, which is a cache space for both fast data retrieval and hardware tri-linear interpolation. Although this hardware tri-linear interpolation is not as accurate as other more sophisticated algorithms like spline or Fourier interpolation, it is much faster and produces reasonable results for real-time applications. Six $(w/p) \nabla T$ matrices are also transferred to the GPU aligned off-chip memory beforehand. This aligned memory space is equipped with ultra-high bandwidth and achieves a 22.6 GB/s transfer rate in our implementation. We also implemented a parallel summation algorithm (Kirk and Hwu, 2010) to calculate the most time consuming step with $O(nN)$ complexity, i.e. the computation of the 6×1 vector

$$\sum_x [(\partial w / \partial p) \nabla T(x)] [I(w(x, p_k)) - T(x)]$$

After numerous tests with our ofMRI datasets, we find that 5 iterations start to yield visible convergence, and 10 to 20 iterations give accurate results using our proposed motion correction algorithm. For both of our proposed method and gradient-based methods, convergence to the global minimum is not guaranteed. Therefore, we employed a multi-

resolution strategy that is also used in FSL FLIRT (Jenkinson et al., 2002; Jenkinson and Smith, 2001) to improve the robustness. When optimization is performed on down-sampled images, many local minima are eliminated, increasing the probability of convergence to the global minimum. Our fMRI data tests show that a down-sampling ratio of 2 improves both the optimization speed and robustness.

2.5. Statistical Analysis

We also designed a parallel Fourier domain analysis method (Engel et al., 1997; Lee, 2012, 2011; Lee et al., 2010) for real-time fMRI analysis. We used the Fourier domain analysis method which is comparable with other methods such as correlation coefficient (Cox et al., 1995) and general linear model (Bagarinao et al., 2003). It has the advantage of also providing the phase information, which is useful for inferring response time of different brain regions.

The original Fourier domain analysis can be written as

$$\rho_{xyz,k} = \sqrt{\frac{2 \times f_{xyz,k}^2}{\sum_{n=1}^{N-1} f_{xyz,n}^2}}, \quad (2.9)$$

where $f_{xyz,n}$ is the Fourier coefficient of the (x, y, z) voxel at the n th frequency. This coherence value measures the significance of the k^{th} coefficient across all the different frequencies. k also corresponds to the number of stimulation periods repeated in the fMRI experiment. For this method, each voxel's time-series need to be Fourier transformed, which is very time consuming and undesirable for real-time applications.

Instead, we propose an alternative approach that calculates the sum of Fourier coefficients in the time domain avoiding redundant FFT calculations. Based on the energy conservation

principle ($\sum_{n=0}^{N-1} f_{xyz,n}^2 = N \sum_{n=0}^{N-1} I_{xyz,n}^2$), the coherence value can be alternatively calculated as

$$\rho_{xyz,k} = \sqrt{\frac{2 \times \left(\sum_{n=0}^{N-1} I_{xyz,n} e^{-j2\pi nk/N} \right)^2}{N \sum_{n=0}^{N-1} I_{xyz,n}^2}}, \quad (2.10)$$

where $I_{xyz,n}$ is the intensity of the voxel (x, y, z) at the n^{th} time point. When performing the analysis, each GPU thread is assigned a single voxel time-series and the coherence analysis for each voxel time-series are conducted in parallel. The Fourier transform coefficient $e^{-j2\pi nk/N}$ can also be calculated in advance and cached in the on-chip GPU constant memory for the highest performance.

In our implementation, users are able to choose the analysis window length to be any multiples of stimulation periods. Sliding window analysis and accumulative analysis of all the acquired data are also supported.

2.6. ofMRI Experiments and Computer Resources

During the ofMRI scans, the pyramidal neurons in the motor cortex, hippocampus, and thalamus were optically stimulated. Activations were detected at both the stimulation site and a downstream remote areas (Lee et al., 2010). Each fMRI experiment consisted of a 30 s baseline acquisition period, followed by six periods of 20 s of optical stimulation and 40 s of rest as described in an earlier publication (Lee et al., 2010).

Our proposed high-throughput system was implemented on a computer with an Intel Core i5 750 quadcore 2.66 GHz CPU, Nvidia Geforce GTX 480 CUDA GPU (1.5 GB GPU memory) and 8 GB DDR3 1333 MHz memory. The software was written in C++ with NVIDIA CUDA extension on an Ubuntu Linux 11.10 operating system.

3. Results

3.1. Overall System Speed

We measured the speed of the proposed real-time system for each processing module in comparison to an optimized single-thread CPU program (C language implementation). All measurements were repeated 5 times and the results were averaged. The GPU timing also includes the data-transfer overhead. As can be seen from the comparison results in Table 1, the proposed GPU based parallel algorithms are much faster than the serial algorithms in all aspects, as expected.

The slowest processing step among all processing modules is gridding, which takes 6.46 ms. Combined with GPU Fourier transform (CUFFT 0.86 ms) and frequency shift function (FFTSHIFT 0.053 ms), the GPU reconstruction time is 7.37 ms. In contrast, CPU serial data-driven reconstruction takes 306.35 ms, which is approximately 42 times slower than the proposed parallel algorithm. The parallel inverse Gauss-Newton motion correction takes 5.39 ms. As will be shown in the next section, the proposed motion correction is approximately 10, 49, 71 times faster than widely used motion corrections in AFNI, FSL, and SPM, respectively (Table 2). For the fMRI analysis, we measured the processing time for 4 different sliding window sizes (i.e. 60, 80, 100 and 120 time frames). On average, the analysis rate per frame is 36.20 μ s with the GPU and 6400 μ s with the CPU, representing approximately 177-fold speedup using the proposed parallel algorithm. Overall, the total GPU processing time per frame is 12.80 ms while the CPU processing time per frame is 312.75 ms.

Note that we are comparing the speed between our proposed parallel algorithms and single-threaded serial algorithms. Admittedly, the performance of CPU serial algorithms can be further improved by utilizing CPU multithreading on computer clusters. However, considering the high level of parallelism afforded by GPUs and their exponentially increasing computation power, our scalable parallel algorithms is a more efficient choice in the long run.

3.2. Accuracy and Speed of the proposed Motion Correction

Since the motion correction is an iterative process, its speed and accuracy are critical issues for real-time applications, particularly for our high-throughput system. Here we tested the speed and accuracy of our proposed parallel inverse Gauss-Newton method in comparison with the widely used software packages including AFNI, SPM and FSL.

3.2.1. Motion Correction Test Criteria and Parameters—We used both real and phantom datasets to test motion correction accuracy and speed. A phantom dataset was created by extracting one frame of an actual fMRI image. The surrounding tissue outside of the brain was removed. To simulate neural activity, two 6 period 20 s on 40 s off 120-time-point activations were designed: one locates in the cortex, and one locates in the thalamus (Fig. 5a, b). The signal modulation was designed to be 1% at the center and fading towards the edge following a Gaussian function (FWHM = 2 mm, volume = 14.79 mm³ for the cortex and FWHM = 1 mm, volume = 1.16 mm³ for the thalamus). Rigid body motion parameters were then generated through MATLAB using a Gaussian random walk process with zero mean and standard deviation of $\sigma_{tx} = 0.273$, $\sigma_{ty} = 0.273$, $\sigma_{tz} = 0.0546$, $\sigma_{\theta_x} = 0.2$, $\sigma_{\theta_y} = 0.2$, $\sigma_{\theta_z} = 0.2$ (Fig. 5c, d). The maximum translation in x and y direction were set to be ± 2.184 mm (± 8 voxels). The maximum z translation was designed to be ± 0.273 mm (± 1 voxel). The maximum rotation in x, y and z direction were designed to be ± 1 degree. This motion was then added to the dataset with the spline interpolation using MATLAB. Gaussian noise was also added to the phantom in the last step to simulate 80 dB SNR.

The real datasets with motion were offline reconstructed from a 128×128×23 matrix-size, 120-time-frame GRE-BOLD datasets that were scanned in real-time. There were two main activation regions shown in these real datasets: the motor cortex and the thalamus.

Other than the interpolation method, all software that we used for comparison were set to operate with default parameters recommended by the authors of the software (Oakes et al., 2005). For fair comparison, we set all algorithms to use tri-linear interpolation to match the proposed inverse Gauss-Newton method. However, since tri-linear interpolation was not supported by AFNI, AFNI was set to use cubic interpolation.

3.2.2. Phantom Test

Speed performance: We first compared the speed of the proposed motion correction algorithm with AFNI, SPM and FSL using the phantom dataset. We repeated the test 5 times for each algorithm and averaged the result. As shown in table 2, our proposed parallel inverse Gauss-Newton algorithm is fastest, requiring only 5.39 ms for a 3D 128×128×23 matrix-size image. AFNI, which is widely used for rtfMRI, shows the second best result with 51.31 ms, approximately 10 times slower than the proposed parallel inverse Gauss-Newton method. Designed for offline processing, FSL and SPM are much slower than the proposed parallel inverse Gauss-Newton algorithm and AFNI. C++ based FSL takes 266.76 ms and MATLAB based SPM takes 383.39 ms.

Precision: We next evaluated the precision using the phantom dataset. The precision was evaluated by using a root mean squared error estimation function proposed by Mark Jenkinson (Jenkinson, 1999)

$$E_{RMS} = \sqrt{\frac{1}{5} R^2 \text{Trace}(R_d^T R_d) + t_d^T t_d}, \quad (3.1)$$

where $R_d = R_0 - R_e$ and $t_d = t_0 - t_e$ are the differences between real and estimated rotation and translation matrix. R is the radius of the brain. Since rat brain was used in the phantom dataset, we assumed $R=10$ mm. Because the transform matrices given by AFNI, FSL and SPM all use different coordinates and units, the matrices were converted to have the same format using the method proposed by Oakes et al (Oakes et al., 2005).

We first generated 10 datasets with distinct random motion. These datasets were then corrected by each method and the corresponding RMS errors were calculated. In Fig. 6, we show the remaining averaged RMS errors from the 10 datasets after motion correction using each method. The proposed parallel inverse Gauss-Newton motion correction algorithm shows the lowest error rate. The proposed parallel inverse Gauss Newton method shows 30% higher accuracy compared to the second best algorithm, AFNI.

Volume Size and Analysis Coherence Value after Correction: The detected activate volume size in fMRI images tend to decrease with increased motion. Therefore, volume size can serve as a precision measure of the motion correction algorithms. We averaged the activated volume of 10 different phantom datasets for each motion correction algorithm separately. As shown in Fig. 7a, our proposed parallel inverse Gauss-Newton algorithm corrected images achieves the highest active volume detection in both the cortex and the thalamus (2.19 mm^3 and 0.35 mm^3), which are approximately 2.5 times larger than the second largest volumes detected by AFNI and SPM motion corrected images.

In Fig. 7b, we also measured the average coherence value for a region of interest (ROI) with commonly active voxels using all methods. The threshold value used for coherence is 0.35. Our algorithm achieves the highest coherence in both the cortex (0.45) and the thalamus (0.42). The second highest coherence values are achieved by AFNI in cortex (0.34) and SPM in thalamus (0.32), both of which are approximately 1.3 times smaller than the proposed algorithm.

3.2.3. Real ofMRI Data Test—Similar to the phantom dataset, we compared the average activate volume sizes for the real ofMRI dataset (Fig. 7c) of 8 repeated scans that were acquired during the same experiment. The proposed parallel inverse Gauss-Newton algorithm results in the largest active volume detection in both the cortex (19.20 mm^3) and the thalamus (0.81 mm^3). FSL gives the second largest active volume: 12.61 mm^3 in the cortex and 0.33 mm^3 in the thalamus. The active volumes detected with AFNI and SPM are approximately the same.

We also compared the average coherence value for ROI selecting commonly active voxels. The threshold value was also 0.35. As shown in Fig. 7d, the proposed parallel inverse Gauss-Newton algorithm gives the highest coherence volume in both the cortex (0.55) and

the thalamus (0.39). FSL and AFNI show the second and third highest coherence (0.43 and 0.42) in the cortex and the same coherence (0.34) in the thalamus. SPM shows the lowest coherence values of among all algorithms (0.40 in the cortex and 0.33 in the thalamus).

The activation map of the real ofMRI dataset before and after the parallel inverse Gauss-Newton motion correction is shown in Fig. 7e. The activation is overlaid onto a T_2 anatomy image. After the parallel inverse Gauss-Newton motion correction, which takes only 5.39 ms on average, the image shows much larger active volume compared to the original uncorrected dataset. The high coherence region (marked yellow) also increased dramatically after the correction.

3.3. Real-time high temporal resolution imaging

With the proposed advanced parallel algorithms, our system is able to acquire, reconstruct, motion correct, and analyze the data in approximately 12.80 ms. Combined with the sliding window reconstruction strategy, the system also enables effectively higher temporal resolution imaging. Despite only partial updates, the resulting time-series from sliding window reconstruction shows much improved temporal dynamics (see results from GRE-BOLD fMRI in Fig. 8, 9). Interestingly, the utility of sliding window reconstruction was illustrated in our experiments by the ability to robustly observe the much-debated initial dips in our ofMRI data. The initial dips were not visible with the 3 s temporal-resolution fully updated reconstruction, while the 750 ms temporal resolution sliding window reconstruction shows clear initial dips in many of the reconstructed voxels.

In Fig. 8a, we show 3 s and 750 ms temporal resolution time-series reconstructed by conventional fully-updated reconstruction and sliding window reconstruction respectively. This ofMRI image is acquired by a 750 ms TR, 4-interleaf, spiral BOLD sequence and the time-series are averaged from dip-detected voxels in the cortex and thalamus. The initial dip is detected by determining whether the first 6 time points give a convex shape. In this dataset, initial dips are detected with 31 mm^3 (849 voxels) in the cortex and 2.87 mm^3 (77 voxels) in the thalamus. In Fig. 8b, six cycles of the time-series averaged into a single epoch are shown with the error bars. In the cortex, the 750 ms time-series clearly shows more temporal details with a potentially small initial dip (Fig. 8b, left). In the thalamus, a very obvious initial dip is observed with the high temporal resolution reconstruction, which was not visible with the 3 s reconstruction (Fig. 8b, right). This clearly demonstrates how the sliding window reconstruction improves the temporal dynamics of HRFs.

Initial dips are often observed with the sliding window reconstruction in our ofMRI data. In Fig. 9, we show three different ofMRI datasets that show the initial dips. The time-series are averaged from dip-detected voxels in the cortex, hippocampus, and thalamus (see blue active voxels in Fig. 9). As can be seen, initial dips are not present in all active regions. Example single epoch traces taken from these three datasets clearly demonstrate how the sliding window reconstructed higher effective temporal resolution traces. Despite the partial (25%) update of the image, the sliding window approach enables the capturing of important temporal dynamics that would have been lost otherwise.

Although the effectively higher temporal resolution can also be achieved by offline sliding window reconstruction, the ability to perform this computationally intensive process in real-time clearly demonstrates the high efficiency and potential of the proposed system for future integration of other advanced, computationally intensive methods to achieve higher image quality and increase the ofMRI study throughput.

4. Discussions and Conclusion

In this paper, we address the need for a high-throughput ofMRI system with a GPU based massively-parallel, real-time system. We carefully designed a novel real-time imaging system, with optimized parallel algorithms for fast data acquisition, reconstruction, and analysis, which enabled real-time inverse Gauss-Newton motion correction, Fourier domain analysis, and sliding window reconstruction. The overall processing speed of the proposed high-throughput system is 12.80 ms per 3D volume, which is much faster than our conventional GRE-BOLD fMRI TR (750 ms in our acquisition), and less than 10 times the b-SSFP fMRI TR (93.75 ms in our acquisition).. In addition to enabling robust real-time processing, such high speed allows sufficient time for advanced techniques including iterative reconstruction, automatic segmentation and brain connectivity analysis to be integrated for real-time processing in the future. This will further improve spatial-temporal resolution and streamline ofMRI studies.

However, with the computation burden eliminated for the real-time ofMRI system, image SNR can become a limiting factor in enabling real-time interactive imaging. Without high enough image SNR, we often rely on multiple-scan averaging that prevents the ofMRI from achieving interactive-feedback-based parameter selection. Possible solutions include utilizing multiple coil arrays (Ohliger and Sodickson, 2006) and cryo-cooled coil acquisition (<http://www.bruker-biospin.com/mricp-overview.html>), which can easily provide SNR improvement by a factor of 2 or more. Use of higher-field MRI systems, such as 9.4 T, 11 T, 17 T can also improve the SNR by approximately 1.3, 1.6 and 2.4 times, respectively. By combining the alternative receive coils and higher-field systems, the image SNR can be improved by a factor of over 2~5, which is equivalent to the SNR achieved by averaging 4 to 25 scans. Considering we typically average 4–8 images to obtain sufficient SNR, achieving sufficient SNR is not expected to be a problem. Furthermore, higher field systems will also give higher BOLD contrast and relax the SNR constraints. With the improvement in the SNR and BOLD contrast, the proposed system will enable interactive feedback based experiments that will considerably accelerate discoveries using ofMRI.

Acknowledgments

This work is supported by the NIH/NIBIB R00 Award (4R00EB008738), Okawa Foundation Research Grant Award, NIH Director's New Innovator Award (1DP2OD007265), the NSF CAREER Award (1056008), and the Alfred P. Sloan Foundation Research Fellowship. J.H.L. would like to acknowledge Karl Deisseroth for providing the DNA plasmids used for the optogenetic experiments. The authors thank all the Lee Lab members for their contribution in helping with the ofMRI experiments and their help editing the manuscript.

REFERENCES

- Aksoy M, Skare S, Holdsworth S, Bammer R. Effects of motion and b-matrix correction for high resolution DTI with short-axis PROPELLER-EPI. *NMR Biomed.* 2010; 23:794–802. [PubMed: 20222149]
- Ances BM. Coupling of changes in cerebral blood flow with neural activity: what must initially dip must come back up. *J Cereb Blood Flow Metab.* 2004; 24:1–6. [PubMed: 14688611]
- Ansorge, AE.; Sawiak, SJ.; Williams, GB. Exceptionally Fast non-linear 3D Image Registration using GPUs; 2009 IEEE Nuclear Science Symposium Conference Record; 2009. p. 4088–4094.
- Bagarinao E, Matsuo K, Nakai T, Sato S. Estimation of general linear model coefficients for real-time application. *NeuroImage.* 2003; 19:422–429. [PubMed: 12814591]
- Baker S, Matthews I. Equivalence and Efficiency of Image Alignment Algorithms. *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition.* 2001; 1:1090–1097.
- Baker S, Matthews I. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision.* 2004; 56:221–255.
- Bhagalia R, Kim B. Spin saturation artifact correction using slice-to-volume registration motion estimates for fMRI time series. *Med Phys.* 2008; 35:424–434. [PubMed: 18383662]
- Caria A, Sitaram R, Birbaumer N. Real-Time fMRI: A Tool for Local Brain Regulation. *Neuroscientist.* 2011
- Cohen MS. Real-time functional magnetic resonance imaging. *Methods.* 2001; 25:201–220. [PubMed: 11812206]
- Cox RW, Jesmanowicz A. Real-time 3D image registration for functional MRI. *Magnet Reson Med.* 1999; 42:1014–1018.
- Cox RW, Jesmanowicz A, Hyde JS. Real-Time Functional Magnetic Resonance Imaging. *Magnet Reson Med.* 1995; 33:230–236.
- deCharms RC. Applications of real-time fMRI. *Nat Rev Neurosci.* 2008; 9:720–729. [PubMed: 18714327]
- Duong TQ, Kim DS, Ugurbil K, Kim SG. Localized cerebral blood flow response at submillimeter columnar resolution. *Proc Natl Acad Sci U S A.* 2001; 98:10904–10909. [PubMed: 11526212]
- Eklund, A.; Andersson, M.; Knutsson, H. Acoustics Speech and Signal Processing (ICASSP). 2010 IEEE International Conference; Dallas, TX: 2010. Phase based volume registration using CUDA; p. 658–661.
- Engel SA, Glover GH, Wandell BA. Retinotopic organization in human visual cortex and the spatial precision of functional MRI. *Cereb Cortex.* 1997; 7:181–192. [PubMed: 9087826]
- Esposito F, Seifritz E, Formisano E, Morrone R, Scarabino T, Tedeschi G, Cirillo S, Goebel R, Di Salle F. Real-time independent component analysis of fMRI time-series. *Neuroimage.* 2003; 20:2209–2224. [PubMed: 14683723]
- Fessler JA. On NUFFT-based gridding for non-Cartesian MRI. *J Magn Reson.* 2007; 188:191–195. [PubMed: 17689121]
- Friston KJ, Ashburner J, Frith CD, Poline JB, Heather JD. Spatial normalization and registration of images. *Hum Brain Mapp.* 1995; 3
- Gembris D, Taylor JG, Schor S, Frings W, Suter D, Posse S. Functional Magnetic Resonance Imaging in Real Time (FIRE): Sliding-Window Correlation Analysis and Reference-Vector Optimization. *Magnet Reson Med.* 2000; 43:259–268.
- Huang TY, Tang YW, Ju SY. Accelerating image registration of MRI by GPU-based parallel computation. *Magn Reson Imaging.* 2011; 29:712–716. [PubMed: 21531103]
- Jackson JJ, Meyer CH, Nishimura DG. Selection of a Convolution Function for Fourier Inversion Using Gridding. *IEEE Transactions on Medical Imaging.* 1991; 10:6.
- Jenkinson M. Measuring Transformation Error by RMS Deviation: FMRIB Technical Report. 1999:4.
- Jenkinson M, Bannister P, Brady M, Smith S. Improved optimization for the robust and accurate linear registration and motion correction of brain images. *Neuroimage.* 2002; 17:825–841. [PubMed: 12377157]

- Jenkinson M, Smith S. A global optimisation method for robust affine registration of brain images. *Med Image Anal.* 2001; 5:143–156. [PubMed: 11516708]
- Johnson KO, Robison RK, Pipe JG. Rigid body motion compensation for spiral projection imaging. *IEEE Trans Med Imaging.* 2011; 30:655–665. [PubMed: 21062678]
- Kim K, Habas PA, Rousseau F, Glenn OA, Barkovich AJ, Studholme C. Intersection based motion correction of multislice MRI for 3-D in utero fetal brain image formation. *IEEE Trans Med Imaging.* 2010; 29:146–158. [PubMed: 19744911]
- Kirk, DB.; Hwu, WW. Programming massively parallel processors: A hands-on approach. Morgan Kaufmann; 2010.
- Kochunov P, Lancaster JL, Glahn DC, Purdy D, Laird AR, Gao F, Fox P. Retrospective motion correction protocol for high-resolution anatomical MRI. *Hum Brain Mapp.* 2006; 27:957–962. [PubMed: 16628607]
- Lee J-H, Ryu J, Jolesz FA, Cho Z-H, Yoo S-S. Brain-machine interface via real-time fMRI: Preliminary study on thought-controlled robotic arm. *Neurosci Lett.* 2009; 450:1–6. [PubMed: 19026717]
- Lee JH. Informing brain connectivity with optogenetic functional magnetic resonance imaging. *Neuroimage.* 2012
- Lee JH. Tracing activity across the whole brain neural network with optogenetic functional magnetic resonance imaging. *Front Neuroinform.* 2011; 5:21. [PubMed: 22046160]
- Lee JH, Dumoulin SO, Saritas EU, Glover GH, Wandell BA, Nishimura DG, Pauly JM. Full-brain coverage and high-resolution imaging capabilities of passband b-SSFP fMRI at 3T. *Magn Reson Med.* 2008a; 59:1099–1110. [PubMed: 18421687]
- Lee JH, Durand R, Gradinaru V, Zhang F, Goshen I, Kim D-S, Fenno LE, Ramakrishnan C, Deisseroth K. Global and local fMRI signals driven by neurons defined optogenetically by type and wiring. *Nature.* 2010; 10:788–792. [PubMed: 20473285]
- Lee JH, Hargreaves BA, Hu BS, Nishimura DG. Fast 3D imaging using variable-density spiral trajectories with applications to limb perfusion. *Magn. Reson. Med.* 2003; 50:1276–1285. [PubMed: 14648576]
- Lee JH, O'Leary HM, Park H, Jolesz FA, Yoo SS. Atlas-based multichannel monitoring of functional MRI signals in real-time: automated approach. *Hum Brain Mapp.* 2008b; 29:157–166. [PubMed: 17370340]
- Li B, Freeman RD. High-resolution neurometabolic coupling in the lateral geniculate nucleus. *J Neurosci.* 2007; 27:10223–10229. [PubMed: 17881528]
- Lindquist MA, Zhang CH, Glover G, Shepp L. Rapid three-dimensional functional magnetic resonance imaging of the initial negative BOLD response. *J Magn Reson.* 2008; 191:100–111. [PubMed: 18207441]
- Lucas BD, Kanade T. An Iterative Image Registration Technique with an Application to Stereo Vision. DARPA Image Understanding Workshop. 1981:121–130.
- Meyer CH, Hu BS, Nishimura DG, Macovski A. Fast spiral coronary artery imaging. *Magn Reson Med.* 1992; 28:202–213. [PubMed: 1461123]
- Miller KL, Smith SM, Jezzard P, Pauly JM. High-resolution FMRI at 1.5T using balanced SSFP. *Magn Reson Med.* 2006; 55:161–170. [PubMed: 16345040]
- Nayak KS, Cunningham CH, Santos JM, Pauly JM. Real-time cardiac MRI at 3 tesla. *Magnet Reson Med.* 2004; 51:655–660.
- Nielsen T, Bornert P. Iterative motion compensated reconstruction for parallel imaging using an orbital navigator. *Magn Reson Med.* 2011; 66:1339–1345. [PubMed: 21469188]
- Oakes TR, Johnstone T, Ores Walsh KS, Greischar LL, Alexander AL, Fox AS, Davidson RJ. Comparison of fMRI motion correction software tools. *Neuroimage.* 2005; 28:529–543. [PubMed: 16099178]
- Ohliger MA, Sodickson DK. An introduction to coil array design for parallel MRI. *NMR Biomed.* 2006; 19:300–315. [PubMed: 16705631]
- Riederer SJ, Tasciyan T, Farzaneh F, Lee JN, Wright RC, Herfkens RJ. MR fluoroscopy: technical feasibility. *Magn Reson Med.* 1988; 8:1–15. [PubMed: 3173063]

- Ruijters D, Romeny BMtH, Suetens P. Efficient GPU-Based Texture Interpolation using Uniform B-Splines. *Journal of Graphics, GPU, & Game Tools*. 2008; 13:61–69.
- Scheffler K, Lehnhardt S. Principles and Applications of Balanced SSFP Techniques. *European Radiology*. 2003; 13:2409–2418. [PubMed: 12928954]
- Shams R, Sadeghi P, Kennedy RA, Hartley RI. A Survey of Medical Image Registration on Multicore and the GPU. *Ieee Signal Proc Mag*. 2010; 27:50–60.
- Sorensen TS, Schaeffter T, Noe KO, Hansen MS. Accelerating the nonequispaced fast Fourier transform on commodity graphics hardware. *IEEE Trans Med Imaging*. 2008; 27:538–547. [PubMed: 18390350]
- Stone SS, Haldar JP, Tsao SC, Hwu W-mW, Liang Z-P, Sutton BP. Accelerating Advanced MRI reconstructions on GPUs. *Journal of Parallel and Distributed Computing*. 2008; 68:1307–1318. [PubMed: 21796230]
- Voyvodic JT. Real-Time fMRI Paradigm Control, Physiology, and Behavior Combined with Near Real-Time Statistical Analysis. *NeuroImage*. 1999; 10:91–106. [PubMed: 10417244]
- Weiskopf N, Mathiak K, Bock SW, Scharnowski F, Veit R, Grodd W, Goebel R, Birbaumer N. Principles of Brain-Computer Interface (BCI) Based on Real-Time Functional Magnetic Resonance Imaging (fMRI). *IEEE Transactions on Biomedical Engineering*. 2004; 51:966–970. [PubMed: 15188865]
- Woods RP, Cherry SR, Mazziotta JC. Rapid automated algorithm for aligning and reslicing PET images. *Journal of Computer Assisted Tomography*. 1992; 16:620–633. [PubMed: 1629424]

Appendix A. Derivation of the derivative of the transformation function

For 6 degrees of freedom rigid body registration, there are 6 parameters: 3 for rotations $\theta_x, \theta_y, \theta_z$ (p_1, p_2, p_3) and 3 for translations: t_x, t_y, t_z (p_4, p_5, p_6).

$$\begin{aligned}
 w &= [R_p x + t_p] = [R_{\theta_x} R_{\theta_y} R_{\theta_z} x \\
 &+ t_p] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{pmatrix} \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{pmatrix} \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \\
 &+ \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \\
 &= \begin{pmatrix} x_1 \cos\theta_y \cos\theta_z - x_2 \cos\theta_y \sin\theta_z + x_3 \sin\theta_y + t_x \\ x_1 (\sin\theta_x \sin\theta_y \cos\theta_z + \cos\theta_x \sin\theta_z) + x_2 (-\sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z) - x_3 \sin\theta_x \cos\theta_y + t_y \\ x_1 (-\cos\theta_x \sin\theta_y \cos\theta_z + \sin\theta_x \sin\theta_z) + x_2 (\cos\theta_x \sin\theta_y \sin\theta_z + \sin\theta_x \cos\theta_z) + x_3 \cos\theta_x \cos\theta_y + t_z \end{pmatrix}
 \end{aligned}$$

The gradient of w is given by

$$\frac{\partial w}{\partial p} = \left[\frac{\partial w^T}{\partial \theta_x} \frac{\partial w^T}{\partial \theta_y} \frac{\partial w^T}{\partial \theta_z} \frac{\partial w^T}{\partial t_x} \frac{\partial w^T}{\partial t_y} \frac{\partial w^T}{\partial t_z} \right]^T,$$

where

$$\frac{\partial w^T}{\partial \theta_x} = \begin{pmatrix} 0 \\ x_1 (\cos\theta_x \sin\theta_y \cos\theta_z - \sin\theta_x \sin\theta_z) + x_2 (-\cos\theta_x \sin\theta_y \sin\theta_z - \sin\theta_x \cos\theta_z) - x_3 \cos\theta_x \cos\theta_y \\ x_1 (\sin\theta_x \sin\theta_y \cos\theta_z + \cos\theta_x \sin\theta_z) + x_2 (-\sin\theta_x \sin\theta_y \sin\theta_z + \cos\theta_x \cos\theta_z) - x_3 \sin\theta_x \cos\theta_y \end{pmatrix},$$

$$\frac{\partial w^T}{\partial \theta_y} = \begin{pmatrix} -x_1 \sin \theta_y \cos \theta_z - x_2 \sin \theta_y \sin \theta_z + x_3 \cos \theta_y \\ x_1 \sin \theta_x \cos \theta_y \sin \theta_z - x_2 \sin \theta_x \cos \theta_y \sin \theta_z + \sin \theta_x \sin \theta_y \\ -x_1 \cos \theta_x \cos \theta_y \cos \theta_z + x_2 \cos \theta_x \cos \theta_y \sin \theta_z - x_3 \cos \theta_x \sin \theta_y \end{pmatrix},$$

$$\frac{\partial w^T}{\partial \theta_z} = \begin{pmatrix} -x_1 \cos \theta_y \sin \theta_z - x_2 \cos \theta_y \cos \theta_z \\ x_1 (-\sin \theta_x \sin \theta_y \sin \theta_z + \cos \theta_x \cos \theta_z) + x_2 (-\sin \theta_x \sin \theta_y \cos \theta_z - \cos \theta_x \sin \theta_z) \\ x_1 (\cos \theta_x \sin \theta_y \sin \theta_z + \sin \theta_x \cos \theta_z) + x_2 (\cos \theta_x \sin \theta_y \cos \theta_z - \sin \theta_x \sin \theta_z) \end{pmatrix},$$

$$\frac{\partial w^T}{\partial t_x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

$$\frac{\partial w^T}{\partial t_y} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

$$\frac{\partial w^T}{\partial t_z} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

When $p = 0$, we get a very simple expression

$$\frac{\partial w}{\partial p}|_{p=0} = \begin{bmatrix} 0 & x_3 & -x_2 & 1 & 0 & 0 \\ -x_3 & 0 & x_1 & 0 & 1 & 0 \\ x_2 & -x_1 & 0 & 0 & 0 & 1 \end{bmatrix}^T.$$

Highlights

- ☐ A system enabling fast, high-throughput of MRI studies is proposed.
- ☐ Highly optimized and massively parallel algorithms process 3D images in 12.80 ms.
- ☐ Enabled future real-time integration of advanced recon and automatic registration.
- ☐ Higher effective temporal resolution is demonstrated by detection of initial dips.
- ☐ Proposed faster and more accurate parallel motion correction algorithm.

Figure 1. ofMRI studies offer high degree of freedom in neural control

With ofMRI, neural population can be very specifically controlled based on their cell type, location, and temporal firing pattern. They can be specifically excited or inhibited while whole brain responses can be observed with spatio-temporal accuracy. Therefore, there is an important need to have intelligent selection of control parameters through real-time feedback, which will accelerate scientific discovery in ofMRI studies.

Figure 2. The proposed high-throughput ofMRI system achieves massive parallelization through CPU and GPU multi-threading

There are three levels of processes running concurrently on the CPU. In level one, CPU thread 1 communicates with the scanner client software and receives the raw data through the Ethernet. It also pre-processes the received raw data including baseline correction and interleaved reordering. In the second level, GPU based parallel image reconstruction, motion correction, and data analyses are performed. In this level, CPU thread 2 functions as a master and controls the slave GPU performing massive parallel processing. In level three, CPU thread 3 displays the real-time data analysis result with the real-time reconstructed images. It also takes input from the user and responds to the input.

Figure 3. High temporal resolution is achieved by the proposed high-throughput ofMRI system with the parallel gridding method and the sliding window reconstruction

a. The serial data-driven conventional gridding method iterates through spiral samples. At each iteration, a spiral sample is interpolated onto Cartesian grids within its corresponding kernel window range. This approach is difficult to parallelize because memory conflicts can occur when values are added to the same grid point. **b.** In our proposed grid-driven parallel gridding approach, each Cartesian grid point is assigned a thread to interpolate the values simultaneously. Each Cartesian grid searches through spiral samples that lie within its kernel window and accumulates interpolated values. To speed up the search, samples are presorted, and cached on the GPU constant memory. **c.** Each GRE-BOLD image is reconstructed from 4 interleaves and each interleaf contains equal number of samples. Conventional fMRI waits for all 4 interleaves to be updated before reconstruction. Utilizing our proposed high-speed system, we can reconstruct and update the images after each interleaf acquisition within a fraction of the TR, which gives effectively higher temporal resolution.

Figure 4. Proposed parallel inverse Gauss-Newton algorithm for 3D 6-DOF motion corrections
 Because the Hessian matrix is constant for the inverse Gauss-Newton algorithm, the calculation is more efficient compared to the original Gauss-Newton algorithm, without compromising accuracy. In step 1, template image gradient ∇T in x, y and z directions are calculated using the Sobel operators. In step 2, both the source image I and the template image T are transferred to the cached GPU texture memory. In step 3, the template gradient ∇T is then multiplied with the Jacobian matrix $\frac{\partial w}{\partial p}$ and the resulting matrices $(\frac{\partial w}{\partial p_k}) \nabla T$ are transferred onto the GPU aligned global memory. In step 4, a 6×6 Hessian matrix H is calculated from the $(\frac{\partial w}{\partial p_k}) \nabla T$ matrices on the CPU. Pre-processing steps 1–4 are implemented on the CPU because the preprocessing only takes a small percentage of overall registration time compared to the iterative optimization process. In step 5 and 6, the difference of the potentially motion corrupted image I and the template image T is multiplied with $(\frac{\partial w}{\partial p_k}) \nabla T$ matrices on the GPU. In step 7, products from step 6 are then parallel summed on the GPU with the parallel summation algorithm. In step 8, the summed 6×1 vector from the GPU is returned to the CPU and the update parameter p is calculated by solving a linear equation with the LU decomposition. In the final step 9, the new warp

function $w(x, p_{k+1}) = [R_{\Delta p_1, \Delta p_2, \Delta p_3}^{-1} R_p x + (t_p - t_{\Delta p_4, \Delta p_5, \Delta p_6})]$ is determined and is sent to the GPU for a new iteration that starts from step 5.

Figure 5. Phantom datasets were designed to test the accuracy of the proposed parallel inverse Gauss-Newton motion correction algorithm

a. Two sites were chosen to have activation for the phantom dataset: one at the cortex and the other at the thalamus. The color of the activation reflects the percentage of signal modulation we added into the time-series. **b.** A periodic hemodynamic function was designed for the 120-frame time-series by using the SPM tools. **c.** The range of the translation parameters used for our tests are shown. The z direction translation ranges were chosen to be smaller than other directions because large z direction translation is unlikely to occur with the stereotaxic cradle we use. **d.** The range of the rotation parameters used for our tests is shown.

Figure 6. Accuracy test shows that the proposed parallel inverse Gauss-Newton method has higher precision compared to widely used packages

a. The average mean RMS errors of 10 different 120-frame time-series with different motion parameters are shown. Our proposed parallel inverse Gauss-Newton method gives the lowest error rate for the majority of the time frames and datasets. **b.** The mean of the RMS errors calculated across the time frames are shown with the corresponding standard deviation (STD). Our proposed parallel Inverse Gauss-Newton method improves the precision by 30% compared to the second best estimation using AFNI.

Figure 7. Quantitative measurement of the activation volume and average coherence demonstrates that the proposed parallel inverse Gauss-Newton method outperforms widely used packages

a. The activation volumes in the cortex and the thalamus after motion correction for the phantom data are shown. Compared to the second best algorithm, our method has approximately 2 times larger volume in both active regions. **b.** Our method also shows higher average coherence value in commonly active regions by a factor of approximately 1.3. **c.** With the real ofMRI data, our method also outperforms others with approximately 1.5 times larger activation volume in the cortex and 2.5 times larger activation volume in the thalamus. **d.** Our method also shows approximately 1.2 times higher coherence value than other algorithms. **e.** Activation map before and after the parallel inverse Gauss-Newton motion correction is shown. As can be seen, both the activation volume and coherence value increased after the proposed motion correction.

Figure 8. Average time-series with sliding window reconstruction reveals faster temporal dynamics not captured with conventional reconstruction

a. Effectively higher temporal resolution is achieved using the sliding window reconstruction. **b.** Averaged single cycle of sliding window reconstructed time-series demonstrates unveils the initial dip (Ances, 2004; Duong et al., 2001; Li and Freeman, 2007; Lindquist et al., 2008) characteristics that were not visible with fully-updated reconstruction. Although the effectively higher temporal resolution can also be achieved by offline sliding window reconstruction, the ability to perform this computationally intensive process in real-time clearly demonstrates the high efficiency and potential of the proposed system for future integration of other advanced methods to achieve higher image quality and increase the ofMRI study throughput. The blue line indicates the optical stimulation period. The time scales on the plots indicate the times at which the images were reconstructed. Therefore, for 3 s temporal resolution reconstructions, there is one point at the end of 3 s while for 750 ms temporal resolution reconstruction four points are reconstructed by the 3 s time.

Figure 9. Initial dips are detected in fMRI datasets with sliding window reconstruction
 Three example datasets are shown where pyramidal neurons in the **a.** motor cortex, **b.** hippocampus, and **c.** thalamus are stimulated. The regions with orange color show active brain regions without the initial dips. Regions with initial dips within the active regions are marked blue. Initial dips are detected in all three datasets. The temporal traces were averaged from initial-dip-detected regions. White triangles indicate viral injection and stimulation site. Only the sliding-window-reconstructed traces with effectively higher temporal resolution show the initial dips.

Table 1
The proposed GPU-based, parallel, high-throughput ofMRI system shows large improvement in speed compared to the serial CPU-based counterpart.

The test results are based on $128 \times 128 \times 23$ matrix-size ofMRI images. For the GPU algorithm timing, the transfer time between CPU and GPU are included. A serial CPU-based method was timed in comparison to our parallel algorithms. The CPU algorithm consists of only the reconstruction (306.35 ms) and data analysis (6.40 ms). The CPU-based Fourier domain analysis time is linearly dependent on the analysis window length. The overall processing time for the CPU-based algorithm is 312.75 ms without the motion correction. In comparison, our parallel algorithm is able to reconstruct and analyze a single 3D image in 7.37 ms and 36.20 μ s, respectively. The GPU FFT runs in 0.86 ± 0.27 ms and the FFTSHIFT takes 0.053 ± 0.012 ms. Note, the higher variance in run time measurement is due to the precision error seen in measurements of small quantities. The GPU-based analysis time is also linearly dependent on the analysis window size. The average overall processing time of the proposed parallel system is 12.80 ms for each 3D time frame, which is approximately 24 times faster than a CPU-based method without motion correction.

	Serial CPU Processing Time (ms)	STD	Parallel GPU Processing Time (ms)	STD
Gridding			6.46	0.78%
CUFFT	306.35	0.38%	0.86	31.96%
FFTSIFT			0.053	22.15%
Motion Correction	See Table 2.		5.39	5.80%
Analysis	6.40 ms/f		36.20 μ s/f	
•60 frames	383.85	2.68%	2.25	0.14%
•80 frames	513.07	3.62%	2.96	0.14%
•100 frames	637.01	3.05%	3.69	0.09%
•120 frames	768.8	2.18%	4.42	0.20%
Total Time / Frame	312.75 (w/o M.C.)		12.80	

Table 2
The proposed parallel inverse Gauss-Newton motion correction method shows significant improvement in speed compared to currently available image registration algorithms

The proposed parallel inverse Gauss-Newton motion correction algorithm is approximately 10 times faster than the real-time AFNI motion correction, approximately 71 times and 49 times faster than the offline methods SPM and FSL, respectively. The results are averaged from 5 tests conducted using each algorithm. Because our GPU motion correction is significantly faster and is subject to higher measurement error, it shows a slightly higher standard deviation of 5.80 % (± 0.31 ms). The MATLAB based SPM also shows a higher standard deviation of 3.88 % (± 14.88 ms), which is likely due to the MATLAB inter-commands switch time variation.

	time (ms)	STD
IGN	5.39	5.80%
AFNI	51.31	0.81%
SPM	383.39	3.88%
FSL	266.76	0.38%