Short Communications

**Real-time fMRI processing with physiological noise correction –
comparison with off-line analysis**

Masaya Misaki^{1#}, Nafise Barzigar^{1,2}, Vadim Zotev¹, Raquel Phillips¹,

Samuel Cheng², and Jerzy Bodurka^{1,3#}

¹Laureate Institute for Brain Research, Tulsa, OK, USA

²Department of Electrical and Computer Engineering, University of Oklahoma, Tulsa, OK, USA

³College of Engineering, University of Oklahoma, Tulsa, OK, USA

[#]Corresponding authors

Email addresses:

jbodurka@laureateinstitute.org or mmisaki@laureateinstitute.org

Word count: 2840 (main text) +160 (figure legends)

Abstract

Background: While applications of real-time functional magnetic resonance imaging (rtfMRI) are growing rapidly, there are still limitations in real-time data processing compared to off-line analysis.

New methods: We developed a proof-of-concept real-time fMRI processing (rtfMRIp) system utilizing a personal computer (PC) with a dedicated graphic processing unit (GPU) to demonstrate that it is now possible to perform intensive whole-brain fMRI data processing in real-time. The rtfMRIp performs slice-timing correction, motion correction, spatial smoothing, signal scaling, and general linear model (GLM) analysis with multiple noise regressors including physiological noise modeled with cardiac (RETROICOR) and respiration volume per time (RVT).

Results: The whole-brain data analysis with more than 100,000 voxels and more than 250 volumes is completed in less than 300 ms, much faster than the time required to acquire the fMRI volume. Real-time processing implementation cannot be identical to off-line analysis when time-course information is used, such as in slice-timing correction, signal scaling, and GLM. **We verified that reduced slice-timing correction for real-time analysis had comparable output with off-line analysis.** The real-time GLM analysis, however, showed over-fitting when the number of sampled volumes was small.

Comparison with Existing Methods: Our system implemented real-time RETROICOR and RVT physiological noise corrections for the first time and it is capable of processing these steps on all available data at a given time, without need for recursive algorithms.

Conclusions: Comprehensive data processing in rtfMRI is possible with a PC, while the number of samples should be considered in real-time GLM.

Keywords

real-time fMRI processing, image processing, physiological noise correction, functional connectivity, neurofeedback

1. Introduction

Real-time functional magnetic resonance imaging (rtfMRI), first introduced by Cox et al. (1995), is a process that uses functional MRI to evaluate brain activation and display the processed image while keeping pace with data acquisition. While applications of rtfMRI are growing rapidly (Sulzer et al., 2013), there are still limitations in real-time data processing compared to off-line analysis. One limitation is that real-time image processing must be at least as fast as the time required to acquire and reconstruct a single brain volume. Another limitation is data availability. In a real-time analysis, the data sample is incrementally acquired so that processes utilizing time-course information cannot be the same as those used in off-line analysis.

Here, we demonstrated that computational time does not limit real-time processing with a current state-of-the-art personal computer (PC). We implemented a proof-of-concept rtfMRI processing (rtfMRIp) system that allows comprehensive real-time processing (RTP) on all available data at a given time, including adapted slice-timing and signal scaling, motion corrections, spatial smoothing, and RETROICOR (Glover et al., 2000) and RVT (Birn et al., 2008), the first instance of real-time physiological noise corrections with general linear model (GLM) analysis. RETROICOR and RVT are computationally intensive physiological noise correction methods that can evaluate variable noise effect voxel-wise, which is more flexible than global signal or white matter and cerebrospinal fluid average signal regression. Here we showed that they could be done in real time. We also verified that time-course analyses in slice-timing correction adapted for real-time processing could output comparable results with off-line analysis. The limitation in rtfMRIp was GLM analysis with a small number of samples. The more noise regressors used for correction, the longer we needed to wait before obtaining a reliable brain activation estimate.

2. Method

2.1. Real-time fMRI processing system (rtfMRIp)

The rtfMRIp system was implemented on a General Electric MR750 3 Tesla MRI scanner equipped with a custom-developed real-time fMRI system (Bodurka and Bandettini, 2008). See Supplementary Figure 1 for detailed system configuration. The RTP software was written in C code as an extension of an existing real-time plugin for AFNI (Cox, 1996) and runs on a Linux (Ubuntu 12.04) real-time processing (RP) computer (dual Intel Xeon (E5-2670) CPUs (8 cores), 128 GB RAM, and an NVIDIA Tesla K20X GPU with 2688 cores and 6 GB memory). The GPU was used only in the GLM analysis. The original real-time plugin features real-time image communication, motion correction, and data saving functions. We extended plugin functionalities by adding multi-threading to run multiple image processing tasks including slice timing, motion corrections, spatial smoothing, signal scaling, and regression with motion, slow drift, and physiological noise corrections. Cardiac waveforms were measured with a pulse oximeter, and respiration signal was measured with a pneumatic belt. Sampling rate of these signals was 200 Hz, which was down-sampled to 40 Hz by averaging five consecutive samples to reduce noise.

2.2. Real-time processing (RTP)

The initial few volumes are ignored to allow fMRI signal to reach a steady state. First, the slice-timing correction was performed by interpolating and resampling the time-course data at shifted time points. In the off-line analysis, either spline, Fourier, or Sinc interpolation is used. In RTP, however, available data points are restricted to past and current time, so high-order interpolation cannot be used. For RTP adaptation, we used resampling with linear and cubic interpolation. The linear method uses only the current and one past data points to interpolate, so it can be applied in RTP without modification. Cubic interpolation uses two points from the past, the current, and one future point to interpolate. We introduced a pseudo future point with the same value as the current one to perform the cubic interpolation. After the slice-timing correction, volume registration and

spatial smoothing were performed. We used the same code that was implemented in the original AFNI source code.

Voxel-wise signal normalization by converting each signal time course to the percent change relative to average signal was applied to the smoothed image. To normalize the signal, we used a fixed number of initial volumes to estimate the average signal, and this initial average is used for scaling all the subsequent time points. The number of volumes to wait before estimating the average is equal to the number of regressors in the GLM analysis. Since the GLM analysis requires a larger number of samples than the number of regressors, regression can only start once a sufficient number of samples have been acquired.

Once enough data had been acquired, the GLM analysis was performed. Polynomial regressors, 12 motion parameters (shift in x, y, and z directions, yaw, roll, and pitch rotations and their temporal derivatives), 8 RETROICOR (4 Fourier series basis functions each for cardiac and respiration) (Glover et al., 2000), and 5 RVT (Birn et al., 2008) regressors were included in the design matrix. The RETROICOR and RVT calculation was implemented in C (based on the AFNI's RetroTS.m MATLAB script). The RetroTS function read cardiac and respiration signals recorded since the start of the scan and calculated 4 cardiac and 4 respiration RETROICOR regressors. The phase of these regressors corresponded to the sample time of the first slice (the aligned time in the slice timing correction). Additionally 5 RVT regressors were also computed.

Every time a new volume was received, the data was incrementally copied to a pre-assigned memory location. The regressors of the motion parameters were also incrementally added. Updates of the RETROICOR, RVT, and polynomial regressors were not incremental, but all the vectors were re-calculated at every volume. Since the RETROICOR and RVT processes include frequency analysis for noise reduction and extracting peak points, recalculating whole estimates with an increased number of samples can improve the estimate of the regressor. This update, however, disables the use of the recursive regression method (Bagarinao et al., 2003) because the design

matrix for the past data was changed. Recursive regression is useful to reduce computational time, but the current implementation demonstrated that computational time did not restrict running a full GLM analysis. The polynomial order was determined by the length of scan at the time of the regression. The order of the polynomial is calculated as $1 + \text{int}(d/150)$, where d is the scan duration in seconds (default in AFNI's 3dDeconvolve). All the column vectors in the design matrix were normalized to unit variance at every update.

The GLM analysis was performed using the quadratic least square (QLS) solving (SGELS) routine in LAPACK (Linear Algebra Package). This routine solves the $Y = BX$ equation to find B while minimizing the squared difference between Y and BX . We used the Intel Math Kernel Library (Intel(R) Math Kernel Library 11.0) for solving this on the CPU and CULA (Humphrey et al., 2010) on the GPU. These high performance libraries enabled fast computation for large-scale data composed of whole-brain voxels. The QLS solution (beta value) was used to regress out the noise components to obtain the residual volume, $r = y - Bx$. While the entire history of the collected data is used to run the regression analysis, the residual volume was calculated only for the current volume since only the current estimate of brain activation is needed in a real-time application.

2.3 System evaluation

We used resting-state fMRI data (14 healthy subjects) from a previous study (Zotev et al., 2011) for evaluating the developed rtfMRIp system. The gradient-echo EPI data were acquired in a resting-state session with eyes open and fixated on a cross on the screen (TR/TE=2000/30ms, flip angle=90°, SENSE acceleration=2, 34 axial slices, FOV=240mm, 96×96 matrix reconstructed to 128×128 resulting in $1.9 \times 1.9 \times 2.9 \text{ mm}^3$ voxel volume, and 263 volumes). The fMRI data of this scan, and corresponding cardiac and respiratory signals, were fed to the rtfMRIp using Dimon software to simulate an experimental session. We also evaluated the rtfMRIp system during actual fMRI scanning (same MRI scanner, same EPI parameters except the number of slices=41 and 225 volumes). Eight

healthy volunteers (age 21–32 years, 5 females) performed a resting-state scan. The study was approved by the Western IRB and the subjects gave informed consent to participate.

Processing time was calculated from when the system received a volume to when the processed volume was saved on a disk, which did not include the time of image reconstruction on the scanner or data transfer from the MRI host computer to the real-time processing computer (less than 1 second). In the simulation, the computation time evaluation was performed with and without using the GPU for GLM analysis. In the actual experiment, four subjects were processed with the GPU and the other four subjects were processed without the GPU.

Comparison between the real-time image processing and the off-line data analysis was performed for the slice-timing correction, signal scaling, and GLM analysis. Only the simulated experiment data was used in these evaluations to test multiple settings of real-time analysis. For the slice-timing correction, on-line linear and cubic corrections were compared with off-line Fourier correction. AFNI's 3dTshift command was used for the off-line analysis. For the signal scaling, we evaluated how quickly the temporal average converged during a scan using correlation between the spatial patterns of the average up to each time and the average of all time points.

For the GLM analysis, output of real-time GLM analysis was compared with that of off-line analysis using AFNI tools. In the off-line analysis, slice-timing correction was applied with 3dTshift with the Fourier method. Motion correction was done by registering volumes to the first volume. Smoothing with a 4mm-FWHM Gaussian kernel was applied to the registered images. The signal values were scaled to percent change relative to the mean across time in each voxel. Twelve motion parameters (shift in x, y, and z directions; yaw, roll, and pitch rotations; and their temporal derivatives), 8 RETROICOR (4 basis functions each for cardiac and respiration), 5 RVT, and 4th-order polynomial regressors were included in the design matrix of the regression analysis. We used 3dREMLfit for the GLM in the off-line analysis. This command performed autocorrelation noise correction, which was not implemented in the real-time processing. The real-time processing used

the same parameters as in the off-line analysis except for the slice-timing correction, signal scaling, and regression. On-line cubic interpolation was used for the real-time slice-timing correction in this evaluation.

3. Results

3.1. Processing time

Figure 1 shows image processing time on the RP computer for the simulated real-time experiment (left) and for the actual real-time experiment (right). Regression analysis began at 27th volume. The graph shows mean processing time (bold lines) and standard error of mean across subjects (thin lines). The processing time for the first (27th) and the last volumes were excluded from the plot since they took an extra-long time due to initialization and closing processes. The mean number of voxels within the mask was 105,394 (SD = 7308) for the simulation dataset, 129,561 (SD = 9693) for the real-time experiment without GPU, and 132,956 (SD = 21809) for the real-time experiment with GPU.

Comparisons with off-line processing

Mean correlation coefficients between the output of the real-time and off-line (FFT interpolation) slice-timing corrections across subjects in each slice were calculated for both linear and cubic real-time interpolations. Even at the longest interpolation size (1.0 sec), the mean correlation was higher than 0.95 (Supplementary Figure 2), and cubic interpolation showed slightly better correlation with the off-line FFT processing than linear interpolation, though the difference was not significant. Regarding the scaling factor, we noted that correlation between the spatial pattern scaled by average until the time of each volume in real-time and that scaled by average of all volumes slowly converged with time (Supplementary Figure 3).

Figure 2A shows the correlation between the outputs of the real-time and off-line processing after the regression. Thirteen RETROICOR and RVT regressors and 12 motion regressors were

included in the regression. The red line is drawn by fitting a function; $y = a \cdot (1 - \exp(-(t-b)/\tau))$, where t is the number of volumes, b is a shift of time, τ is a time constant, and a is a scaling factor. Figure 2B shows the signal standard deviation within a brain in the real-time processed output. The small standard deviation of the residual image at early volumes indicates that over-fitting happened due to there being an insufficient number of samples to fit multiple regressors. The time course of standard deviation was similar to that of the correlation between the real-time and off-line processing outputs, suggesting that over-fitting made the output of the real-time processing less reliable compared to the off-line analysis. The same analysis was performed with the fewer regressors by excluding physiological noise regressors (Figs. 2C and 2D) and further excluding the six motion derivative regressors (Figs. 2E and 2F). Using fewer regressors shortened the time to converge the correlation between the real-time and off-line outputs.

4. Discussion

We have developed a proof-of-concept real-time fMRI processing system (rtfMRIp) and showed that computationally intensive fMRI processing could be done in real time with a PC equipped with a dedicated GPU. Current rtfMRIp implementation includes slice-timing correction, motion correction, spatial smoothing, signal scaling, and GLM analysis with motion parameters, polynomial regressors, and the first real-time implementation of physiological noise correction with RETROICOR and RVT regressors. These processing steps could be completed in less than 300 ms with more than 250 volumes for a whole brain with more than 100,000 voxels. While this is the first demonstration of intensive real-time fMRI data processing comparable to off-line analysis, computation time will no longer be a prohibitive factor in the further development of rtfMRIp capabilities in the near future.

Aside from computation time, RTP requires several different approaches compared to off-line processing in time-course data analysis. For slice-timing correction, reduced alternative methods for real-time processing could output comparable results with off-line processing. We note that

performance of low-order interpolation in the slice-timing correction depends on the length of TR, and this correction might not be necessary for a short TR. Nevertheless, we included this process to demonstrate that it could be implemented in rtfMRIp if necessary. For signal scaling, spatial pattern of scaled signal in real time was not the same as in off-line analysis due to the difference of scaling factor. This could be critical when a multi-voxel pattern analysis (MVPA) is used for neurofeedback. Another normalization (e.g. Lee et al., 2015) should be applied when MVPA is used in real time. We also note that the scaling does not affect the temporal pattern of signal change. Since updating the scaling factor at every volume could induce artificial signal fluctuation in the scaled time course, a fixed scaling factor should be used when a voxel-wise analysis is used.

For the GLM analysis, certain number of samples must be collected for the RTP output to converge, and time to converge was longer when a larger number of noise regressors was used. We do not have a fundamental solution for the problem of GLM convergence time other than simply waiting. This could limit an rtfMRI application, for example in a neurofeedback experiment. If one used a simple block design and brain activation is assumed to be large enough compared to noise, there is no need to include many noise regressors and neurofeedback could be started earlier. However, if one wants to use functional connectivity for a neurofeedback signal, physiological noise suppression should be included in RTP, resulting in a longer wait time before a reliable estimation of functional connectivity (with better noise suppression) is available. One must carefully balance the trade-off between thorough noise correction and early availability of neurofeedback signal.

We note that the presented system is not necessarily an optimal implementation of rtfMRIp, nor is the aim of this report to propose a computationally optimal implementation. Rather, we aim to demonstrate that modern PC technology has already given us the freedom to implement computationally intensive processing. Hinds et al. (2011) have already presented a similar approach for rtfMRIp, in which the noise model was regressed out from fMRI data by incremental GLM

analysis. They, however, did not implement slice-timing correction, signal scaling, or spatial smoothing and the processed region was restricted to within an ROI. Our system implemented all of these processing steps for whole-brain data, in addition to real-time implementation of RETROICOR and RVT physiological noise correction. Such comprehensive real-time fMRI analysis has not been demonstrated before. Our results indicated that this computationally heavy processing can be realized in real-time fMRI even without a supercomputer facility but with a commonly available PC.

The current report focused on real-time fMRI image processing before showing a feedback signal. **An application of our system to neurofeedback experiment was demonstrated in Misaki et al. (2015).** While processing step selection could vary depending on the purpose of experiment, we showed that multi-step and computationally intense the whole-brain fMRI preprocessing, including GLM analysis, can now be conducted with all available data at a given time, in real time. The current results will increase the possibility of rtfMRI application, such as feedback of connectivity measures, which requires physiological noise correction, and MVPA, which requires whole brain signal to evaluate feedback signal.

Acknowledgement

This research was supported by the Laureate Institute for Brain Research and the William K. Warren Foundation and in part by W81XWH-12-1-0697 award from the U.S. Department of Defense. Source code of the extension of AFNI real-time system reported in the paper is available upon request.

References

- Bagarinao E, Matsuo K, Nakai T, Sato S. Estimation of general linear model coefficients for real-time application. *Neuroimage*, 2003; 19: 422-9.
- Birn RM, Smith MA, Jones TB, Bandettini PA. The respiration response function: the temporal dynamics of fMRI signal fluctuations related to changes in respiration. *Neuroimage*, 2008; 40: 644-54.
- Bodurka J, Bandettini PA. Real-time software for monitoring MRI scanner operation. 14th Annual Meeting of the Organization for Human Brain Mapping: Melbourne, Australia, 2008: S85.
- Cox RW. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Comput Biomed Res*, 1996; 29: 162-73.
- Cox RW, Jesmanowicz A, Hyde JS. Real-time functional magnetic resonance imaging. *Magnetic Resonance in Medicine*, 1995; 33: 230-6.
- Glover GH, Li TQ, Ress D. Image-based method for retrospective correction of physiological motion effects in fMRI: RETROICOR. *Magn Reson Med*, 2000; 44: 162-7.
- Hinds O, Ghosh S, Thompson TW, Yoo JJ, Whitfield-Gabrieli S, Triantafyllou C, Gabrieli JDE. Computing moment-to-moment BOLD activation for real-time neurofeedback. *Neuroimage*, 2011; 54: 361-8.
- Humphrey JR, Price DK, Spagnoli KE, Paolini AL, Kelmelis EJ. CULA: Hybrid GPU Accelerated Linear Algebra Routines. *SPIE Defense and Security Symposium (DSS)*, 2010.
- Lee D, Jang C, Park HJ. Multivariate detrending of fMRI signal drifts for real-time multiclass pattern classification. *Neuroimage*, 2015; 108: 203-13.
- Misaki M, Suzuki H, Luo Q, Wong CK, Phillips R, Zotev V, Bodurka J. Hyperscan Real-Time fMRI Neurofeedback for Studying Interacting Individuals. 21st Annual Meeting of the Organization for Human Brain Mapping: Honolulu HI USA, 2015: 4431.

Sulzer J, Haller S, Scharnowski F, Weiskopf N, Birbaumer N, Blefari ML, Bruehl AB, Cohen LG, DeCharms RC, Gassert R, Goebel R, Herwig U, LaConte S, Linden D, Luft A, Seifritz E, Sitaram R. Real-time fMRI neurofeedback: progress and challenges. *Neuroimage*, 2013; 76: 386-99.

Zotev V, Krueger F, Phillips R, Alvarez RP, Simmons WK, Bellgowan P, Drevets WC, Bodurka J. Self-Regulation of Amygdala Activation Using Real-Time fMRI Neurofeedback. *PLoS ONE*, 2011; 6: e24522.

Figure legend

Figure 1. Computation time of real-time image processing without GPU (CPU: blue lines) and with GPU (red lines) for the regressions. The bold lines and thin lines indicate mean processing time and its standard error across subjects, respectively.

Figure 2. A. Comparisons between the real-time processed image and off-line processed image after the GLM analysis. The thick black line shows mean correlation between the real-time and off-line processed images at each volume, and the thin blue lines show standard error. The red line shows a fitted function line. The dotted vertical line indicates the time point where the fitted line reached 80% of the correlation at the last volume. **B.** Time course of signal standard deviation within a brain in the real-time processed image. Conventions are the same as A. **C** and **D.** Same plot as A and B excluding RETROICOR and RVT regressors. **E** and **F.** Same plot as A and B excluding RETROICOR, RVT, and motion derivative regressors.

Figure 1

Figure 2