NeuroImage **10,** 91–106 (1999)
Article ID nimg.1999.0457, available online at http://www.idealibrary.com on IDEAL®

Real-Time fMRI Paradigm Control, Physiology, and Behavior Combined with Near Real-Time Statistical Analysis

James T. Voyvodic

MR Research Center, Department of Radiology, University of Pittsburgh Medical Center, Pittsburgh, Pennsylvania

Received January 6, 1999

This study presents an integrated approach to on-line fMRI data processing that combines real-time paradigm control and real-time MR image statistical analysis with nearly real-time integration of fMRI behavioral and physiological data. The real-time paradigms involve accurate timing control of multiple independent processing streams for stimulus presentation, physiological monitoring, behavioral response recording, and scanner synchronization. The real-time image analysis provides high resolution MR image reconstruction, head motion detection, translational motion correction, and *t* test statistical activation maps for either block design or single-trial based paradigms. The near real-time analysis allows physiological and behavioral data collected during a paradigm to be combined with the MR time series and provides extended data filtering and statistical processing within a few minutes after the end of the scan. This integrated approach improves fMRI reliability for both clinical and research studies. © 1999 Academic Press

INTRODUCTION

Although functional magnetic resonance imaging (fMRI) provides unprecedented opportunities for observing human brain function noninvasively, the technique has thus far lacked the reliability needed to make it a practical approach for many clinical and research applications. The small signal changes observed across a time series of MR images are only partially due to task-specific effects; they also involve a variety of task-independent variables such as physiological fluctuations caused by the cardiac and respiratory cycles (Jezzard *et al.*, 1993; Weisskoff *et al.*, 1993; Noll and Schneider, 1994; Hu *et al.*, 1995; Le and Hu, 1996), as well as task-related confounds such as head motion, attention, anxiety, or poor task performance. Variations in any of these parameters can affect the fMRI time course and obscure the paradigm-specific patterns of brain activation being investigated. Problem conditions due to any component variable are usually not detected in time to be corrected because fMRI data sets tend to be quite large and are therefore usually not analyzed until after the subject has left the scanner. In practice, therefore, obtaining usable fMRI results is often limited to scanning cooperative subjects under optimal conditions. This limitation is a problem for many research studies and is clearly a major obstacle for applying fMRI in a clinical context.

FMRI reliability could be improved by monitoring many different physiological and behavioral parameters during scanning and by performing MR image processing on-line in real-time. Serious problem conditions, such as poor task performance or excessive motion, could thus be identified in time to correct them and repeat the scan. Less serious sources of variation in physiological and behavioral parameters can be integrated into the analysis either to filter out related fluctuations in the MR time course or to sort the task-dependent time series data into different subconditions.

Previous studies involving real-time fMRI analysis (Cox *et al.*, 1995; Goodyear *et al.*, 1997; Cohen *et al.*, 1998; Frank *et al.*, 1998; Posse *et al.*, 1998) have demonstrated the feasibility of on-line processing. In general, however, these approaches lack the flexibility to cope with fMRI paradigms of different designs (e.g., cyclic or randomized block designs or single trial) or the ability to reduce nonspecific noise sources associated with scanner instability, detector noise, and image reconstruction artifacts. In addition, most do not include the subject's behavioral (task performance and head stability) and physiological (cardiac and respiratory oscillations) variables into the on-line statistical processing of MR images.

The problem in routinely achieving comprehensive fMRI analysis in real-time for a wide range of applications lies in efficiently integrating many different processes all running in parallel. These largely independent processes include audio or visual stimulus presentation, behavioral and physiological response

1053-8119/99 $30.00

recording, the scanner's acquisition of MR images, and statistical image analysis. Such parallel processing can be achieved by time-sharing multiple tasks on a single computer, which facilitates coordination and passing of information between tasks, and/or by distributing different tasks across multiple computers, which eases the real-time processing burden on any single machine.

In designing software for real-time fMRI a number of important performance constraints must be considered. For example, the subject's environment must be carefully controlled and monitored throughout each scan with accurate timing and smooth presentation of sensory stimuli, while at the same time achieving real-time recording of subject performance. Recording cardiac and respiratory oscillations during each scan can significantly improve subsequent fMRI results (Hu *et al.,* 1995; Le and Hu, 1996). It is also essential to have accurate synchronization between the paradigm and the MR scanner, rapid MR data transfer between the scanner and the analysis software, and efficient transfer of task timing and other paradigm data between the paradigm and analysis components. The entire system should be sufficiently flexible to accommodate many different activation paradigms and analysis strategies and yet easy enough to use so that it can be routinely applied without the need for complicated user interactions. Finally, it is not enough that fMRI analysis be done quickly; the statistical processing itself should also be capable of providing results of comparable quality as those that could be obtained by off-line processing methods.

This report demonstrates an approach to improving fMRI reliability that combines flexible software tools for accurate real-time paradigm control and on-line fMRI analysis. Paradigm control uses a single standard personal computer to achieve continuous real-time paradigm performance that includes simultaneous stimulus presentation, automatic scanner synchronization, and monitoring of a variety of voluntary and physiological responses. The fMRI analysis can perform MR image reconstruction, head motion detection, translational motion correction, and *t* test statistical activation maps for either block design or single-trial-based paradigms, all in real-time. (For this discussion, real-time fMRI analysis is defined as processing and displaying the data during the scan itself and as fast as the scanner can acquire the data.) The analysis can also combine the paradigm's behavioral and physiological data with the MR time series for extended data filtering and statistical processing within a few minutes after the end of the scan. These software tools can all be coordinated via a simple menu-driven interface on the paradigm computer, thus demonstrating that reliable real-time fMRI can be efficiently accomplished by a single MR technologist on a routine basis.

METHODS

Hardware

Paradigm control. Functional MRI paradigms were implemented using a Macintosh computer (either Quadra 660 AV computer or a PowerMac 7500; Apple Computer Corp., Cupertino, CA) containing an analog/digital processing board (MacADIOS II, GW Instruments, Cambridge MA). Visual stimuli were presented to the subject in the MR scanner via a standard video projector (Epson America Inc., Torrance, CA) that back-projected images onto a translucent screen mounted above the subject's chest within the bore of the scanner. The video projector was contained within a custom-built RF shielded box (MRA Inc., Washington, PA); all connections between the computer in the scanner control room and the shielded box in the magnet room were via fiber optics (Thulborn *et al.,* 1996). Audio stimuli were presented via speakers mounted within the RF shielded box, with sound conveyed from the speakers to the subject via hard plastic tubing and audio headphones placed inside a pair of sound deadening ear muffs.

MR acquisition and analysis. MR images were acquired using a 3.0 Tesla Signa scanner (General Electric Medical Systems, Milwaukee, WI) with echo-planar imaging capabilities (Advanced NMR Systems, Inc, Wilmington, MA). Image analysis was performed on a SGI O2 workstation (R10000 CPU), a SGI Power Challenge L computer containing 8 R10000 CPUs (Silicon Graphics Corp., Mountain View, CA), or a Sun Ultra 2 (Sun Microsystems, Mountain View, CA). The analysis computers were connected to the scanner's data acquisition computers via a high speed CDDI network (approximately 900 Kbytes/s throughput) and to the Macintosh computers via a 10BaseT Ethernet link.

Software

The MR scanner operated under version 5.2 Signa software (GE Medical Systems) and echo planar imaging was controlled using the APD2 software package (Advanced NMR). The Macintosh computers used version 7.5 MacOS, the SGI analysis computers used Irix 6.2 (Power Challenge) or Irix 6.5 (O2) operating systems, and the Sun Ultra used Solaris 2.5 operating system. In addition to this commercial software these studies depended on three locally developed programs: CIGAL, MacDaemon, and fScan, which are briefly described in the following sections.

CIGAL. Paradigm control software was written using the Compilable Imaging, Graphics, and Analytical Language (CIGAL, formerly known as IMAGR; Voyvodic, 1986, 1996; Purves and Voyvodic, 1987) running on a Macintosh. CIGAL is a general purpose programming language designed for scientific image

processing, graphics, and the control of scientific experiments. FMRI paradigms were prepared using CIGAL's high-level data manipulation and menu features to generate or import image, text, or sound stimuli and create data tables specifying the sequence and timing of paradigm events. In order to achieve real-time paradigm performance for fMRI, the CIGAL language was extended by the addition of a real-time command processor. The central feature of the real-time processor is its ability to specify the exact time (to within 20 μs), at which each operation is to be executed.

The real-time processor allowed multiple processing streams to execute in parallel and all in real-time. This was accomplished by creating independent process pointers to control the execution of separate blocks of code, and then automatically interleaving the progress of each processing stream to ensure that every operation was performed as close as possible to its scheduled execution time. The obvious problem with multiple parallel streams was that the real-time accuracy in any single stream could be off by as much as the time it took for the slowest operation performed by the other streams. This problem was minimized by making the time to execute each individual operation short, and in most cases of known duration (see Results). A priority hierarchy was used to resolve timing conflicts.

MacDaemon. Unix command files generated by CIGAL on the Macintosh were executed on a Unix workstation by means of a simple C program called MacDaemon written in-house. Communication between the Macintosh and the host computer was entirely via Unix disk files that were cross mounted as Macintosh files using K-AShare/K-Talk software (Xinet, Inc., Berkeley, CA). MacDaemon was executed on the host computer and simply waited for a particular file to appear in its cross mounted directory. Once that file appeared its contents were read and each line was executed as a standard Unix shell command line. This simple scheme provided the Macintosh with an efficient way to initiate the execution of Unix-based analysis software.

fScan. All fMRI image analysis in this study was performed by fScan, a custom written C program run on a Unix-based computer, either initiated directly on the host or by the paradigm computer via MacDaemon. fScan was used for a variety of analysis tasks as shown in Results. This was possible in large part due to its flexibility in accepting input data. Thus, for real-time processing it read image or raw *k*-space data directly from the scanner's disk files, waiting as necessary for the data to be acquired, whereas for postprocessing it read MR raw or image data either from the scanner or from previously saved local files. It could read the entire data set or any rectangular subset. It was similarly flexible about accepting other time course input data, such as the timing of different task conditions, the subject's behavioral responses or physiological wave form recordings. When combining data with different sampling rates fScan converted all time courses internally to the time base of the MR acquisitions.

Having read the data fScan also provided flexibility in data processing, either one acquisition at a time (e.g., image reconstruction, image motion estimation, spatial filtering, intensity normalization, calculation of means and variances, generation of *t* test activation maps, or single trial time course averaging) or one voxel time course at a time (e.g., temporal filtering, detrending, spike filtering, spectral analysis, correlation, cross-correlation, or regression with a reference time course). In general, the former operations could all be performed in essentially real-time during scanning (see Results), whereas the latter needed to wait for the entire time course to be acquired before they began. Optimal software performance was achieved by combining efficient processing algorithms with flexible run-time options that allowed fScan to process multiple time courses or matrices in parallel using multiple CPUs on a single computer or to break up the data set for distributed processing across multiple workstations. FScan also provided flexible X-window based display tools for real-time presentation of activation maps and head motion plots, with interactive options for manipulating and querying all of its different types of data.

Although most of fScan's statistical options involved standard fMRI algorithms, some were less straightforward and are briefly summarized here. Pulse sequence-dependent image reconstruction was implemented for echo planar scanning involving single shot acquisitions or two-shot mosaics, using full or partial *k*-space reconstruction, with optional optimization filtering to reduce ghosting. Head motion was estimated by plotting changes in the spatially weighted center-of-intensity for each image in both the horizontal and vertical dimensions, and for the entire 3-D volume in the normal direction (see Fig. 3). Motion correction filtering was patterned after the method of Eddy *et al.* (1996) and involved using the image motion estimates to adjust the raw *k*-space data matrix to compensate for translational motion within the image plane before undergoing a second image reconstruction pass. Voxel time course "detrending" removed low frequency variations by subtracting out a smooth low frequency time course based on a quadratic spline curve fit through four evenly spaced nodes.

The most flexible time course filtering tool allowed voxel time courses to be regression filtered against any reference time course. This was done by sorting time points by reference value, calculating a smoothed mean time course curve across the parameter range, and then subtracting the appropriate mean variation from every time point in the MR signal (after the method of Hu *et*

al., 1995; see Fig. 6). Up to 10 different time course parameter lists could be regression filtered in a single pass. Regression maps were generated by calculating a mean reference-sorted time course for each voxel, as for regression filtering, and then calculating the variance of the reference-binned time course divided by the variance of the entire time course (see Figs. 4 and 6).

Performance Testing

The relative efficiency of fScan's processing algorithms, and its ability to achieve real-time performance, were tested by timing the execution of multiple data analysis runs. Functional MRI data sets of three different matrix sizes were collected in 8-min scans at the scanner's maximum duty-cycle acquisition rates. The three acquisition matrices used were: 128 × 64 full *k*-space, 256 × 128 partial *k*-space, and 512 × 128 partial *k*-space 2-shot mosaic (zero filled to 512 × 256 for image reconstruction).

Processing speed was determined in two ways. First, the time required to perform each individual component of the analysis was measured independent of the scanner's acquisition rate; this was done by first acquiring an entire data set and then starting the analysis. Second, the actual real-time processing rate was measured by analyzing and displaying fMRI results during the scan itself and measuring the latencies between the time of different stages during data acquisition and the time at which the associated data appeared in the displayed results. For these latency measurements only the SGI Power Challenge computer was used. The component speed measurement tests were run at night when the computers were relatively not busy, whereas the latency measurements were performed in the morning (between 9 a.m. and noon) under normal usage conditions for both the computers and the network. For all measurements the tests were repeated at least three times to determine means and standard deviations.

Paradigms

For testing purposes, three different fMRI paradigms were used. In the first, subjects watched a central fixation crosshair on a black background, around which appeared radially moving white dots in animated 20-s blocks, separated by 5-s fixation-only blocks; they were instructed to move a joystick and randomly press two button box keys during the animated sequences (see Fig. 1). In the second paradigm, subjects fixated on a central crosshair while a hemifield 8 Hz flickering checkerboard rotated about the fixation point (after Engel *et al.*, 1996), making a complete revolution every 32 s (see Figs. 3–5). For the third paradigm, subjects alternated between 30-s blocks in which they either read simple sentences (black text on a white screen) and answered associated true/false questions using a button box, or scanned a line of consonant-only text and then randomly pressed a button (see Fig. 6). Pulse sequence parameters for each paradigm are described in the figure legends in the Results section.

In addition to presenting stimuli, all CIGAL paradigms also recorded heartbeat and respiration signals continuously throughout each scan, sampling both channels at approximately 100 Hz. The subject's behavioral responses were recorded from analog joystick inputs sampled at 100 Hz and from digital push-button inputs sampled at over 1 KHz. Except where otherwise stated, paradigms were programmed to begin task execution automatically as soon as they detected that the scan had begun, as indicated by a change in the analog channel connected to the current monitor for one of the scanner's gradient amplifiers. In addition to these fMRI paradigms carried out on human subjects, other fMRI scans were performed using inert phantoms in order to assess the speed and reproducibility of data processing and data transfer operations. The specific scanning and analysis parameters used for each of these paradigms are described in the Results section.

Paradigm Analysis

For real-time fMRI analysis the CIGAL paradigm program generated a text file specifying fScan analysis commands and sent it to the analysis workstation. For the analysis, raw data were read from the scanner as they were acquired, transferred across the local network and reconstructed into MR images; these images were then Gaussian filtered and separated by task condition to calculate *t* test activation maps. Other statistical tests performed included measurements of mean image intensity and head motion during the scan. The results were displayed on an X-windows terminal in the scanner control room, with the display being updated once per task condition for activation maps and every eight images for motion plots.

fMRI Scans

Subjects were normal healthy volunteers between the ages of 18 and 45. Each subject was screened for contraindications to MRI and gave informed consent before entering the magnet. Subjects were fitted with foam ear plugs and sound dampening ear muffs to reduce acoustical noise. Each subject was connected to the scanner's respiratory bellows monitor and a finger pulse-oximeter (Nonin Medical Inc., Minneapolis, MN). Scans of the brain were performed using a standard bird cage quadrature head coil (GEMS, Milwaukee, WI). An echo planar gradient echo pulse sequence was used with a 25 ms TE, 90° flip angle, 40 × 20 cm FOV, and matrix sizes of 128 × 64, 256 × 128, or 512 × 128. Details of the matrix used and the number of slices and

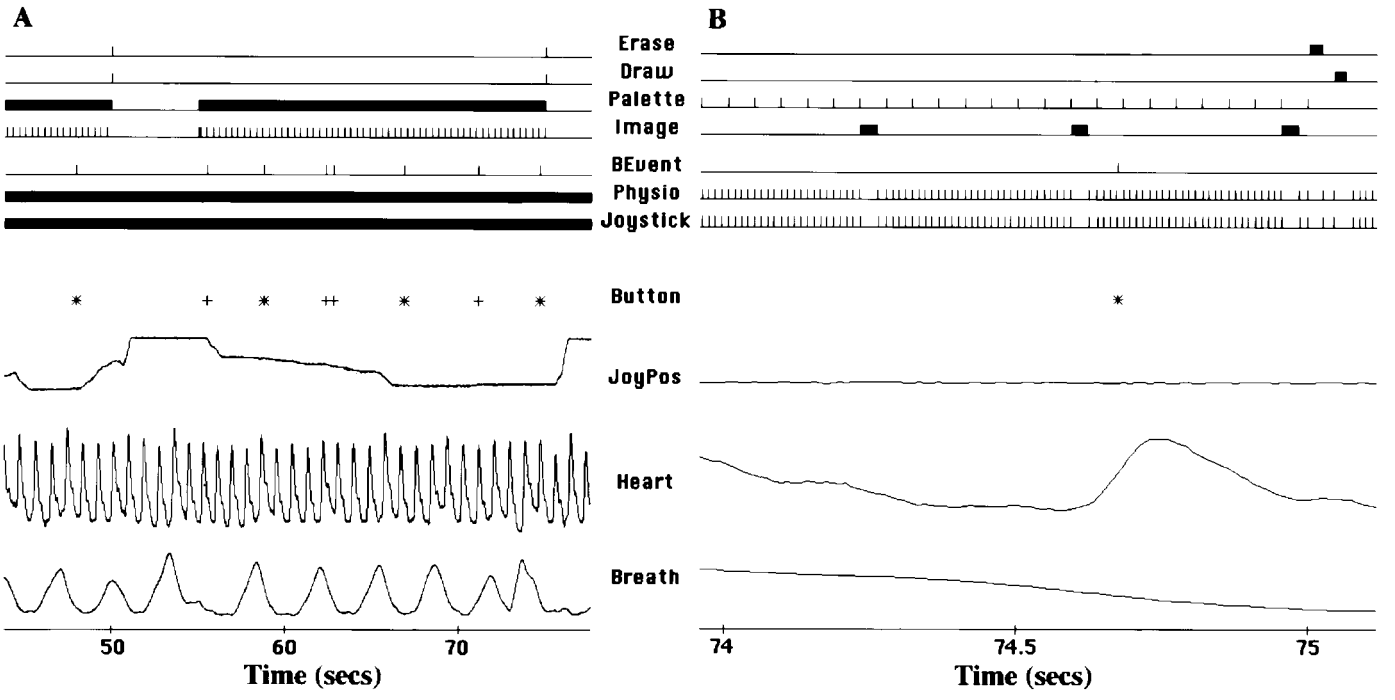


FIG. 1. Interleaved timing for sample CIGAL real-time paradigm program. A and B show the same data plotted on two different time scales (times in seconds). The top seven traces in each panel indicate the timing of different events during the paradigm, with the width of each tick mark indicating the approximate amount of time needed to process each event. The top four traces show the major events involved in presenting the visual stimulus (erase the screen, draw a crosshair, change the color palette, transfer a new image), which cycled between 5 s of a static fixation crosshair followed by 20 s of continuous 22 Hz animation of radially moving dots (410 × 210 pixels). The next three traces show the timing of response recording events (waiting for button presses, recording analog physiological data and analog joystick data), and the bottom four plots show the data that were actually recorded (* and + symbols indicate which of two buttons). The four stimulus events (erasing the screen, drawing the crosshair, changing the video palette, and copying an image to the screen) were all controlled by a single real-time processing stream, whereas the three response recording events were each handled by separate asynchronous processing streams. Note the regular timing of the stimulus events and smoothness of the response recordings in B.

TR times are specified where appropriate in the Results.

RESULTS

Paradigm Software Real-Time Performance

Before running fMRI paradigms CIGAL's real-time paradigm performance was first evaluated by measuring the speed of its real-time operators. Using a Macintosh Quadra 660AV computer, 52 of the 101 real-time operators had mean execution times of under 100 μs; this time included all standard command overhead as well as the time it took to read the computer's real-time clock (approximately 20 μs). An additional 28 real-time operators had mean execution times of 1 ms or less. Of the remaining 21 operators only 7 took more than 10 ms to execute; those are therefore the only operations likely to cause any timing conflicts among multiple parallel processing streams executing commands at 100 Hz or less. These slowest operators all transferred blocks of data for either images (>300 × 200 pixels) or sound and took approximately 20–60 ms to execute. Comparing execution times across different Macintosh models (PowerMac 7100/80 or 8100/100, Quadra 660 AV, or Performa 475) gave similar results, presumably because most of the operations are rate limited by the video display and peripheral hardware devices, rather than the CPU itself.

The accuracy of real-time control for the real-time processor was determined by measuring the difference between the time at which the software specified an operator should begin execution and the time at which it actually began. Running only a single processing stream, event timing was accurate on average to within 6.7 ± 19 μs (mean and SD) over a sample of 500,000 test events, with over 98% of all events starting within 1 clock tick (20 μs) of the specified time. Of 500,000 test events only 1 started more than 600 μs after the scheduled time; that event was late by almost 4 ms.

These accuracy measurements therefore reflect the degree of real-time control available to the user for initiating the execution of any operation, in the absence of any prior or parallel timing conflicts. The only operators that could not be controlled with this accuracy were those that involved video display changes that were forced to occur in synchrony with the video

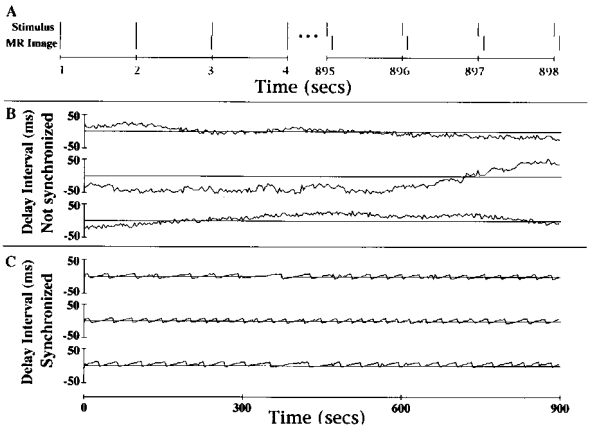


FIG. 2. Synchronization timing between paradigm and scanner. A shows the timing of stimulus events generated by the paradigm software every 1000 ms plotted as tick marks above the horizontal line, and scanner image acquisitions with a TR of 1000 ms plotted as tick marks below the horizontal line. The stimulus events indicate the onset of 100 ms flickering checkerboard. The first 4 s and the last 4 s of a 900-s scan are shown. MR images were initially acquired with 0 phase relative to the stimulus, but by the end of the 900 s the pulses had a phase delay of 75 ms relative to each stimulus. B plots the phase delay between paradigm stimulus and MR image acquisition for the entire time course of six scans similar to that shown in A. The top three curves show scans in which the stimulus and acquisition were started together and then not resynchronized during the rest of the scan. Variability in the delay interval arose from variability in the 60 Hz line signal used to control scanner timing. The bottom three curves show scans in which the stimulus was resynchronized on every MR image acquisition. Variability in the delay interval was due to each stimulus change being phase-locked to the computer's video refresh cycle. (Scan parameters: Gradient echo EPI, matrix = 128 × 64, single slice, total time = 900 s.)

refresh rate. CIGAL automatically rescheduled such operators to start 3 ms prior to the next video refresh.

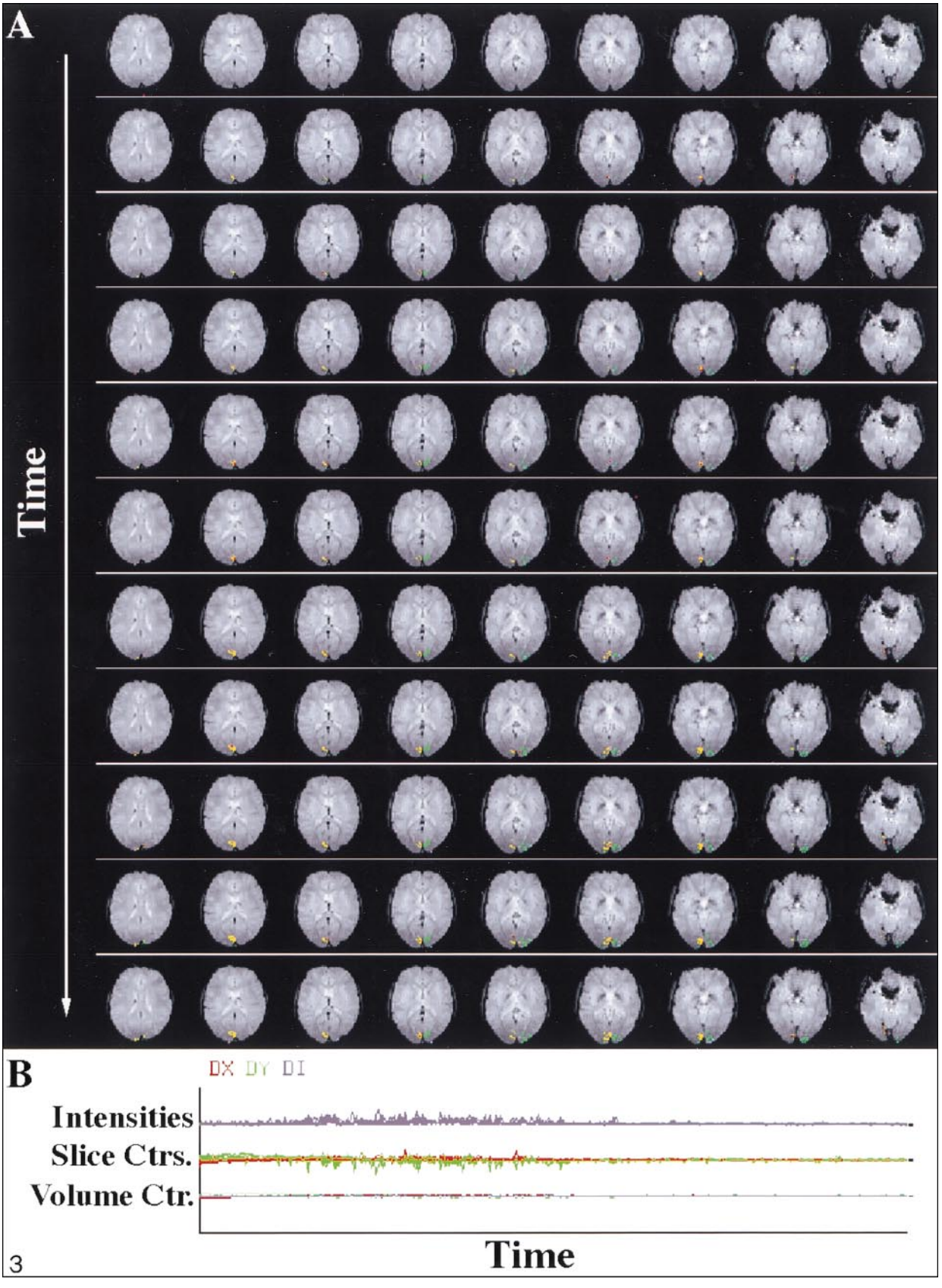
For fMRI paradigms, in which multiple processing streams executed in parallel, the effective real-time accuracy was necessarily less than the ideal accuracy demonstrated above. Figure 1 shows the actual execution timing for a typical paradigm. This paradigm involved four asynchronous processing streams executing in parallel: the first controlled a continuous animated video stimulus in which the video palette was changed every 45 ms and a complete new image (410 × 210 pixels) was copied from program RAM to video memory every 360 ms; another read cardiac and respiratory oscillations on two analog channels every 10 ms; a third sampled two additional analog channels every 10 ms to record the X and Y positions of a joystick being moved by the subject; the last continuously polled the response buttons on a digital input channel and displayed a colored spot on the video screen for the investigator (outside the subject's viewing region) each time a button was pressed. Figure 1B shows that the regular 100 Hz sampling of the physiological and joystick channels was briefly affected by each video image transfer, although this did not significantly compromise the shape of the recorded wave forms. Video animation was observed to be very smooth throughout the paradigm, which is reflected in Fig. 1 by the regular timing of palette controlled frame changes and video memory transfers.

Synchronization with MR Scanner

Real-time synchronization between the paradigm software and the MR scanner was tested in two ways: by synchronizing the paradigm to the scanner and by synchronizing the scanner to the paradigm. In the first method the paradigm program was started before the scanner and then waited for the scanner to start pulsing. CIGAL detected the start of scanning as a voltage change on an analog input channel connected to the current monitor on one of the gradient magnet power supplies, so that the paradigm timing began automatically as soon as the scan began. Figure 2A shows an example of the timing differences observed between the scanner and paradigm during the course of a 15-min scan. In the example the scanner's nominal 1000-ms TR was actually 1000.083 ms on average, resulting in a 75-ms phase difference by the end of the scan. Such timing differences arose from variability in the 60 Hz line signal, to which the GE scanner automatically phase locked every TR interval. As the accuracy of the local electrical utility company's 60 Hz signal is ±0.06 Hz (±0.1%), in the worst case the scanner acquisition timing could be off by up to 900 ms over a 15-min scan. In our timing tests the actual observed variation was typically under 100 ms over 15 min, although as seen in Fig. 2B the line frequency could be observed to vary even within a single scan. For most fMRI paradigms 99.99% timing accuracy is probably sufficient, so that careful synchronization at the start of MR acquisition usually ensures adequate synchronization throughout the scan.

For high temporal resolution imaging more precise

FIG. 3. Progressive fScan real-time analysis. Each row in A shows *t* maps (yellow: $t >= 4$, green: $t <= -4$) generated at a single time point for nine oblique axial slices during a visual field mapping paradigm in which the subject watched a flickering hemifield checkerboard rotate around a central fixation point once every 32 s. The display was updated every 16 s and the figure shows every 4th time point. The *t* maps gradually revealed the activated visual cortex. B shows the image motion plots generated for the same scan, with time represented on the horizontal axis. This display appeared gradually with eight new time points added every 16 s. The figure shows the final result. The blue curves at the top plot relative mean image intensity for each slice. The red and green curves in the middle plot the intensity weighted horizontal and vertical centers of the images for each slice and the bottom set of curves plot the displacement of the center of each nine slice volume in all three dimensions. The distance between the two sets of motion curves corresponds to a displacement of 1 voxel unit. The scan parameters were: GE EPI, TR = 2 s, TE = 25 ms, FOV = 40 × 20 cm, matrix = 128 × 64, voxel size = 3.2 × 3.2 × 3 mm, total scan time = 640 s.


A
Time
B
DX DY DI
Intensities
Slice Ctrs.
Volume Ctr.
Time
3

synchronization was achieved by having the paradigm software continuously monitor the scanner's gradients and resynchronize the paradigm with every echo-planar data acquisition pulse (Fig. 2C). This approach was tested using 15-min scans of a phantom and measuring the delay intervals between each 1 Hz video stimulus and the corresponding 1 Hz MR acquisition pulses (TR = 1000 ms). Three successive 15-min scans without continuous resynchronization resulted in delay intervals with 54-, 58-, and 107-ms variability between each stimulus onset and the subsequent MR acquisition (Fig. 2B). Once again this variability was because the nominal 1000-ms TR interval varied with fluctuations in the 60 Hz line signal. Performing the same test on three scans with continuous resynchronization resulted in intervals with 15-, 16-, and 19-ms variability (Fig. 2C). In this case the variability and saw-toothed patterns seen in the delay interval plots were because every video stimulus change was automatically phase-locked to the display system's video refresh rate. Except for this 15-ms video phase-locking limitation, therefore, resynchronization ensured that there was a nearly constant time interval between each visual stimulus presentation and subsequent scanner acquisition.

Another synchronization strategy tested was to have the paradigm software control scanner timing by explicitly triggering each TR interval. In this case the scanner was started first but did not actually acquire MR data until after the paradigm was started and began to send trigger pulses to the scanner's external trigger input. Although the timing of this trigger pulse could be as accurate as the real-time processor allowed (e.g., within 600 μs as shown above when flagged as a high priority event), such accuracy proved to be undesirable because triggering at a fixed TR interval resulted in trigger pulses at varying phases of the 60 Hz line cycle and high noise artifacts in the resulting MR time series (not shown). These noise artifacts disappeared when the specified TR interval was phase-locked to the line frequency using CIGAL's real-time analog trigger operator. The paradigm program could deliver such phase-locked trigger pulses and also compensate for drifts in the line frequency to ensure that the actual TR interval always remained within 17 ms (i.e., 1 line cycle) of the nominal interval. Given the extra overhead involved in generating accurate phase-locked trigger pulses, however, it has proven to be advantageous for the paradigm software to trigger the scanner only for scans that require very accurate time resolution or necessarily involve TR's of varying time intervals.

On-Line fMRI Analysis

An example of fScan's real-time fMRI analysis for a CIGAL paradigm is shown in Fig. 3. This scan involved a visual field mapping study in which the subject fixated on a central spot, about which slowly moved a flickering hemifield checkerboard pattern. For real-time processing the data were copied off the scanner as they were generated, a copy was stored on the host's local disks, and differences in cortical activation were analyzed by a *t* test. The figure shows the gradual emergence of *t* map differences in activation in the two cortical hemispheres, corresponding to the timing of primarily left versus right sided stimuli. The analysis also progressively plotted image stability curves as an index of image motion. The display was updated every 16 s during the scan.

Once the scanner had finished acquiring data a second pass of fScan processing was used to analyze the data set more comprehensively. For fMRI scans using the CIGAL paradigm software any behavioral or physiological data were copied from the paradigm computer to the analysis computer within 15 s of the end of the paradigm and so were immediately available to be integrated into the second pass of fScan processing. Figure 4 shows an example of a typical second pass of fScan's on-line analysis using the same visual mapping data set as that shown in Fig. 3. All the results shown in the figure were generated and displayed within 1 min after the end of the scan, using the reconstructed images already generated in real-time by the first pass of fScan. This second, "nearly real-time," analysis included a display of the raw data which could be inspected for signs of any data acquisition problems, and it performed a cross correlation analysis to calculate maximum correlation coefficients and optimal stimulus phase for each voxel. It also analyzed the cardiac and respiratory wave forms to identify peaks and determined the phase and oscillation rate of each wave form corresponding to the time of each MR data acquisition. The time courses of all these parameters were displayed as well as the time courses of interactively selected single voxels or groups of voxels.

This second pass also generated "regression maps," which graphically indicated the amount of signal variability in each voxel time course that could be attributed to the variability in a specific parameter. In the example, each voxel time course was sorted by six different parameters resulting in maps indicating the relative magnitude and spatial distribution of signal variability due to: task condition, cardiac phase, cardiac rate, respiratory phase, respiratory rate, and time. Time was used as a regression parameter to determine the baseline stability of the time course across the entire duration of the scan; this tends to be the largest source of variability, as was the case for the scan shown in Fig. 4. This data set also showed typical cardiac phase regression maps highlighting the major blood vessels, and respiratory phase regression maps with most variability around the edges of the brain. As expected, the task reference produced regression maps with localized sites of variability, corresponding to the

areas detected by the *t* test and correlation maps, whereas cardiac and respiratory rate regression maps were relatively dark because those variables remained relatively constant throughout the scan.

The anatomical regions associated with functional activity could be identified more accurately during the on-line analysis by mapping the activation maps onto higher resolution structural images, provided that a series of structural images had been acquired prior to the functional scan. Figure 5 shows the functional phase maps generated in the scan from Figs. 3 and 4 superimposed over a spoiled GRASS data set that covered the entire head with approximately 1-mm resolution in each dimension. The two data sets were approximately aligned by fScan automatically and then the alignment was improved interactively by a manual adjustment that took about 15 s. Such high resolution mapping could also be done in the first, real-time, pass if the structural images were available.

Rather than simply plotting maps of signal variability, fScan's regression processing algorithm could also be used to filter parameter-specific variability out of each voxel time course. An example of such regression filtering is shown in Fig. 6. This shows the regression maps and task *t* test activation maps from a language study in which the subject viewed alternate 30-s blocks of either consonant-only text or simple English sentences. The maps were generated both before and then again after filtering out variability associated with time, as well as both phase and rate for cardiac and respiratory oscillations. On an SGI O2 computer fScan took approximately 1 min to regression map or regression filter all of the variables for the entire data set (using an automatic region of interest mask option to ignore voxels outside of the head).

In the example of Fig. 6 there was considerable variability in the subject's cardiac and respiratory rates, which was apparently due to the subject being slightly more anxious when reading the sentences than when scanning the nonsentences. The prefiltered *t* test map exhibited considerably more regions of activation than are normally seen for this language task. Because the physiological parameters were not completely independent of the task conditions, filtering out those variables significantly reduced the amount of activation seen in the postfiltered *t* map at a given *t* threshold ($t >= 3.5$ in Fig. 6). Although it still showed some atypical areas, the resulting *t* map clearly showed Wernicke's language area as well as a weak signal in the region normally corresponding to Broca's language area, both in the left hemisphere.

Analysis Software Real-Time Performance

The speed of fScan's on-line fMRI image analysis was quantified for three different size MR acquisition matrices and compared on two different computer platforms to determine how much computer power was necessary to achieve real-time results for each matrix size. The standard functional analysis involved displaying progressive plots of image stability to indicate head motion, and *t* test maps to show regional changes in MR signal intensity that were correlated with the task conditions. This analysis started either with image data reconstructed on the scanner or raw *k*-space data, in which case fScan reconstructed the images itself.

Table 1 shows the processing times for each of the steps involved in fScan's real-time analysis. For the smallest size matrix (128×64) the ANMR echo-planar computer system was capable of reconstructing images almost as fast as the raw data were acquired, so that raw data and image data were both available almost immediately to be read by the fMRI analysis. Using the already reconstructed images allowed nearly real-time performance for scans using 128×64 matrices. However, to truly keep up with MR data acquisition, even for these small matrices, image reconstruction had to be done remotely. All three computers tested were able to perform image reconstructions faster than the data acquisition rate, so that better overall performance could be achieved by transferring only the raw data from the scanner. The image stability and motion estimation calculations as well as the *t* test calculations took relatively little time and could easily keep up with the data acquisition rate.

For the larger matrix sizes, image reconstruction on the echoplanar system's computer lagged far behind the rate at which the data were acquired. Table 1, however, shows that the SGI computers could reconstruct the 256×128 partial *k*-space acquisitions in under 400 ms per image using a single CPU, which was still faster than the acquisition rate. For 512×128 2-shot partial *k*-space acquisitions a single SGI CPU took over 1500 ms per image to reconstruct. Using 6 CPU's in parallel, the effective reconstruction times could be reduced to 76 and 325 ms/image for the 256×128 and 512×128 matrices, respectively. These times were more than fast enough to achieve real-time performance. Fast reconstruction rates made it possible to estimate head movement after the first reconstruction and then adjust the raw *k*-space data so that a second reconstruction pass could generate images that were all aligned with respect to their centers in real-time. Although such motion correction, based on the Fourier space method of Eddy *et al.* (1996), was limited to in-plane translational motion, this first order correction significantly reduced motion artifacts for many scans. Because the center-of-intensity calculation is not a true measure of head position, however, this registration step can generate artifacts in slices that include eye or mouth movement and so is not yet a routine part of our real-time analyses.

Although fScan's analysis was fast enough to keep up

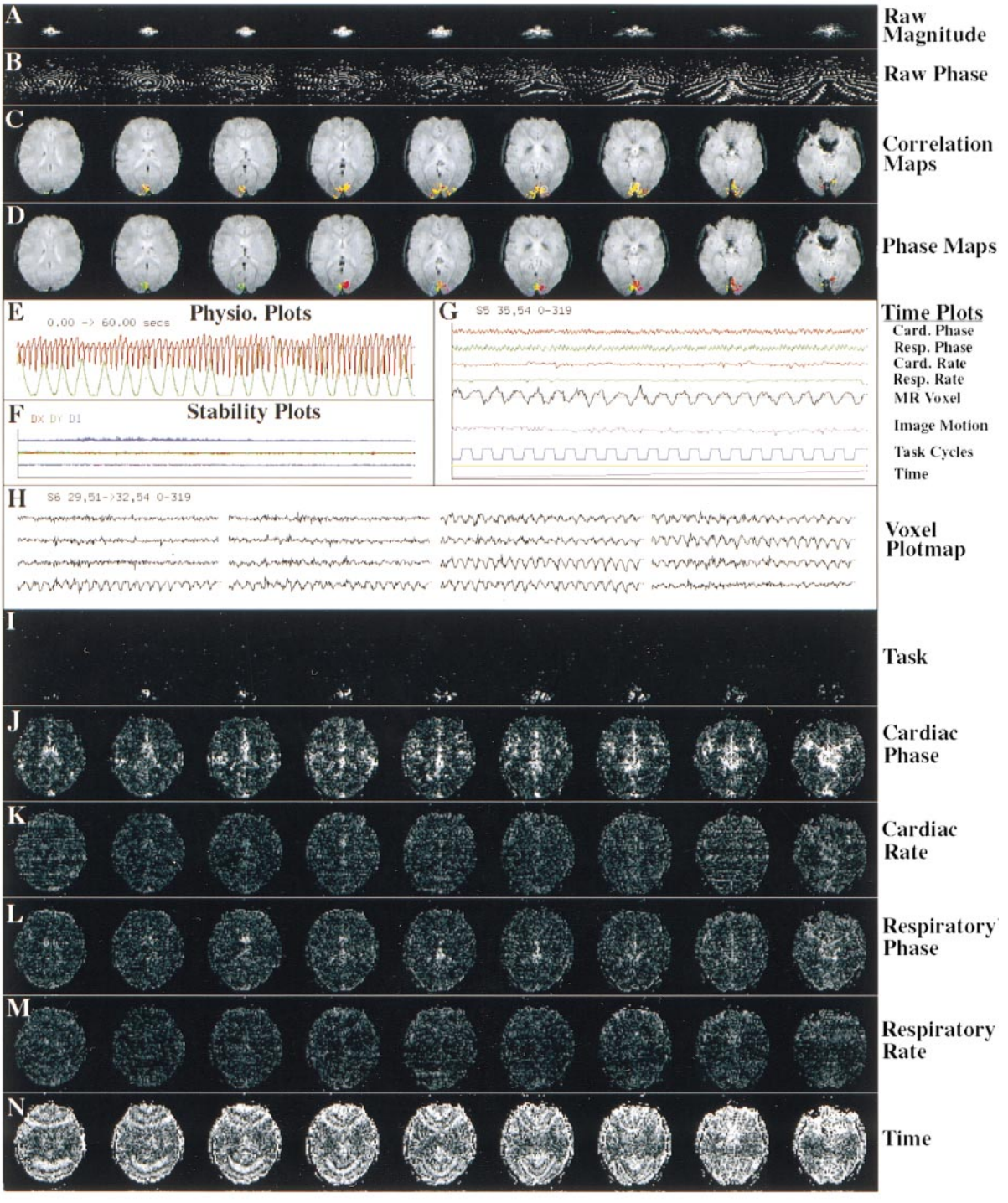


FIG. 4. Near real-time fScan analysis. These data are from the same visual mapping scan as described in the legend of Fig. 3 and could all be displayed within 1 min of the completion of the scan. A and B show the magnitude and phase displays of the raw *k*-space data. C and D show the correlation coefficients and phase maps resulting from a time course cross correlation with a boxcar reference wave with 32-s period. Panel E plots a portion of the cardiac and respiratory data collected during the scan, with tick marks indicating each peak. F shows the same stability plots as described in the legend of Figure 3. G shows multiple physiological parameter time courses for a selected slice as well as the task reference and the MR time series for a selected "activated" voxel. H plots the MR time courses for a selected (4×4) region of voxels. I–N show regression maps plotting the relative contributions of six different reference parameters to the variability in the MR time course for each voxel. The regression maps were only calculated over the head (brightest) portions of each slice. All six regression maps are plotted using the same intensity scale.

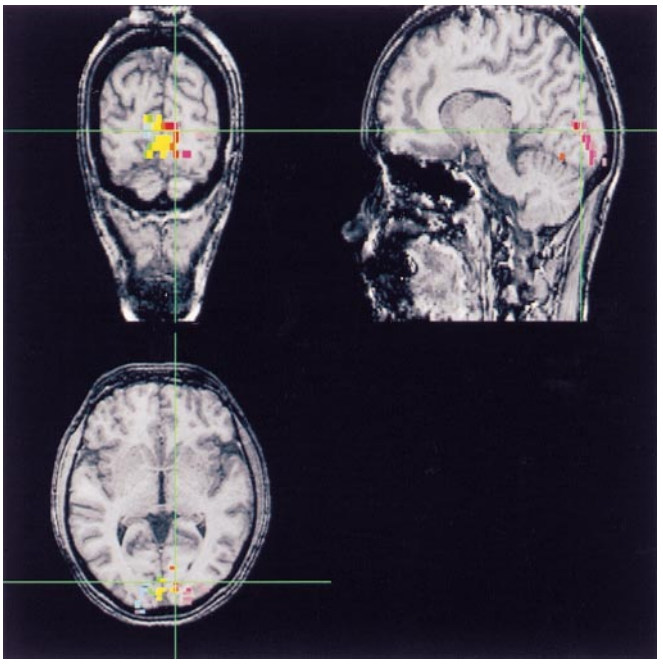


FIG. 5. Combining functional and structural data. Functional activated voxels are superimposed over a high resolution 3-D set of conventional MR images. The data set is from a visual mapping study similar to that shown in Figs. 3 and 4. The phase map voxels are plotted over coronal, sagital, and axial spoiled GRASS (SPGR) views to show visuotopy in the visual cortex. SPGR scan parameters: 3-D GRASS, TR = 25 ms, TE = 4 ms, FOV = 24 × 18 cm, matrix = 256 × 192, voxel size = 0.94 × 0.94 × 1.5 mm, total scan time = 460 s.

with data acquisition, and therefore to be considered essentially real-time, it necessarily involved some delay between the time of each MRI acquisition and the time that data appeared in the displayed results. This delay arose because of delays in the scanner's writing data to its disk files, delays in transferring data across the network, and the time required for fScan to process the data. The net result of these delays was measured for our hardware configuration to determine the overall latency involved in getting the data to fScan's display (Table 2). Once again time measurements were made for the three matrix sizes and acquisition modes tested in Table 1, but in this case only the SGI Power Challenge computer system was used and tested to determine the minimum number of parallel CPU's needed to achieve real-time performance.

As seen on the bottom line of Table 2, the fMRI analyses of the small, medium, and large matrix sizes tested were completed within 10 s of the end of scanning using one, two, and three CPU's in parallel, respectively. The analysis in each case involved image reconstruction with ghost tuning, image stability and motion estimation, T-test statistical mapping, visual data display, and storing of the reconstructed images to disk files. The program updated the plots of image stability after every eight images and the T-map display after every task block. There were longer and more variable delays at the beginning of each scan, primarily because the scanner tended to collect several acquisitions before writing them all together to disk files. As a result fScan typically had to wait for several acquisitions and then process the data in bursts. At the end of the scan the last acquisition was written to the scanner's disks as soon as the acquisition terminated and closed its output files, so that fScan runs that were fast enough to keep up almost always completed all data processing within 10 s of the last MR acquisition.

The processing performance for the voxel time course aspects of fScan's second-pass analysis were quantified using the same 3 MR matrix sizes and computer platforms as tested for Table 1. In this case the processing started after scanning had completed and the data (either raw k-space or MR images) were read from local disk files. Table 3 shows mean execution times for the most common second pass operations. The results demonstrate that these postprocessing steps can all be accomplished on-line within just a few minutes after completion of an fMRI scan.

DISCUSSION

This study demonstrates a comprehensive, flexible software environment for completely automated real-time fMRI paradigm control and immediate on-line analysis. The paradigm software is based on the CIGAL programming language (Voyvodic 1986, 1996; Purves and Voyvodic, 1997) and features a real-time processor with submillisecond accuracy and the ability to process many different streams of operations in parallel. This paradigm software allows a standard personal computer to simultaneously present audio and/or video stimuli, record behavioral and physiological responses, and ensure accurate synchronization between the paradigm and MR image acquisition. The analysis software (fScan) is based on a hierarchical approach that combines initial fMRI results in essentially real-time (with a few seconds latency) with more comprehensive statistical postprocessing. The real-time processing is capable of data acquisition, image reconstruction, head motion measurements, translational image registration, *t* test activation maps, and single trial time plots. The postprocessing analysis is completed within a minute or two after the end of each scan; it incorporates behavioral and physiological time course data and performs time domain analyses to filter out spike noise, signal drifts, cardiac and respiratory oscillations, and to generate correlation and phase maps and multiple regression maps.

CIGAL and fScan demonstrate that fMRI software can combine the flexibility to accommodate a wide range of imaging paradigms, with fast and efficient on-line data processing and essentially turn-key operation. In our facility these two programs have made real-time image analysis a routine part of fMRI scanning. Thus, a single MR operator can control the

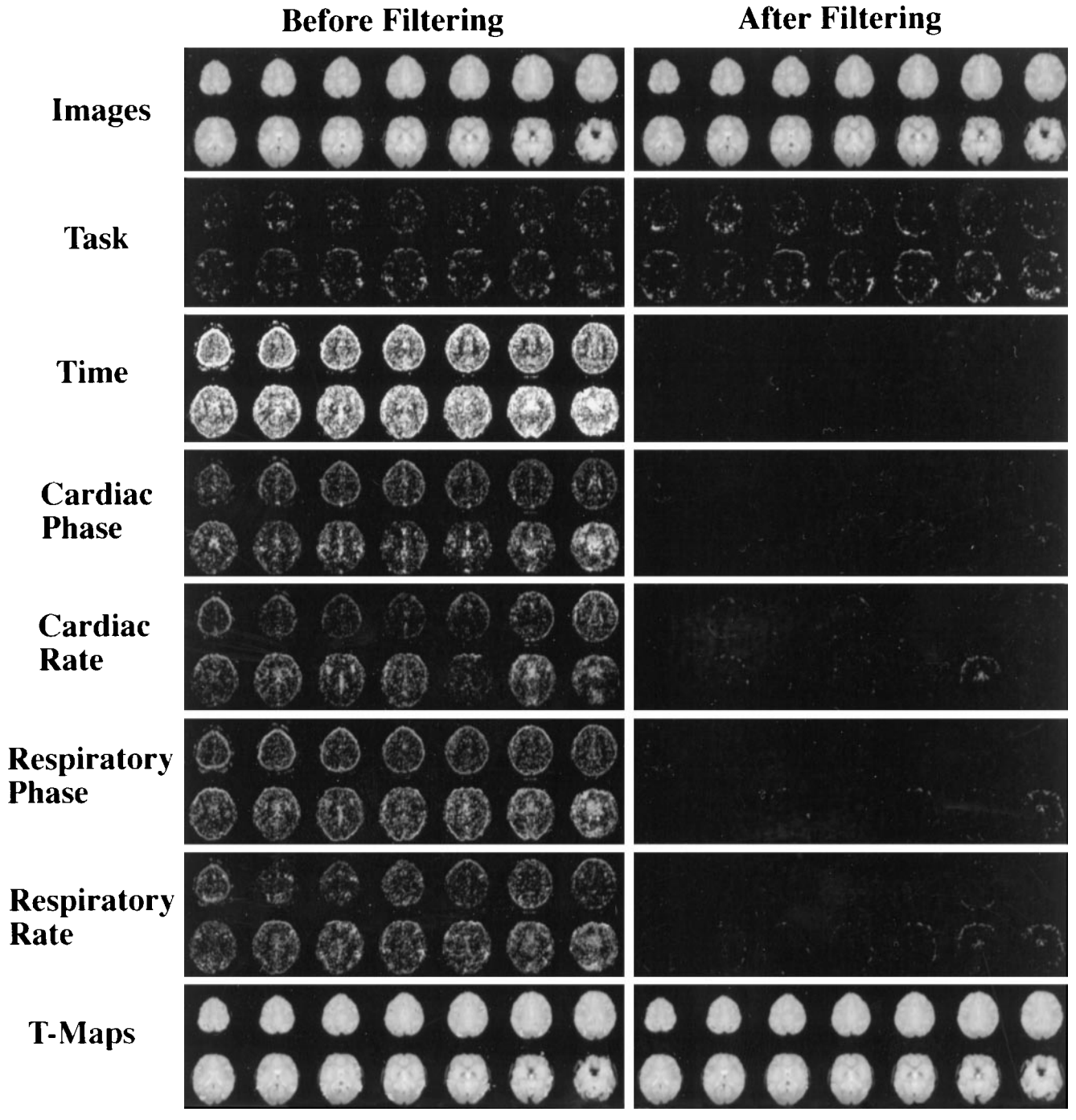


FIG. 6. fScan regression filtering. Regression maps for a language study are shown before and after fScan regression filtering. Filtering was performed to reduce MR signal variability associated with time (drifts), cardiac phase, respiratory phase, cardiac rate, and respiratory rate. The top row panels (images) show the first image for each slice. The middle six rows show the regression maps, all plotted at the same relative intensity scale. The bottom row (*t* maps) shows the *t* test activation maps ($t >= 3.5$) for the sentence reading task.

scanner and paradigm software for most functional studies to generate image stability plots and task-dependent *t* test maps of brain activation that are displayed and progressively updated during the scan itself. For most applications, new paradigms can be implemented in CIGAL and analyzed with fScan using only menu based interactions to specify data and timing parameters, without the need for any additional programming. Although this report emphasizes using the two programs together as a complete on-line fMRI package, they are each self-contained C programs and can also be used independent of each other. Both

TABLE 1

Real-Time Performance

Matrix size	128 × 64			256 × 128			512 × 128		
Clipped image size	64 × 64			128 × 128			256 × 256		
Voxel size (mm × mm × 3 mm slices)	3.1 × 3.1			1.6 × 1.6			0.8 × 0.8		
Acquisition mode	1-shot, Full K			1-shot, Partial K			2-shot mosaic, Partial K		
Raw data size (per image)	32 Kb			88 Kb			176 Kb		
Image data size (per image)	8 Kb			32 Kb			128 Kb		
Max. acquisition duty cycle (per image)	**214 ms**			**500 ms**			**1000 ms**		
ANMR image reconstruction (per image)	252 ms			2500 ms			7442 ms		
Network raw data transfer (per image)	34 ms			105 ms			210 ms		
Network image data transfer (per image)	14 ms			75 ms			298 ms		
Computer	SGI-O2	SGI-PC	Sun Ultra	SGI-O2	SGI-PC	SGI-PC	SGI-O2	SGI-PC	SGI-PC
Number of CPUs	1	1	1	1	1	6	1	1	6
fScan processing times (ms/image)									
Image reconstruction	49.2 ± .8	47.2 ± 0	97.1 ± .1	323.3 ± .5	382.0 ± 2.0	75.6 ± .2	1518 ± 2	1653 ± 1	324.6 ± 4.6
Image motion calculation	1.4 ± .0	0.8 ± 0	1.8 ± 0	5.5 ± .0	3.3 ± .1	5.4 ± .4	30.6 ± .2	11.7 ± 0	16.0 ± .4
Image registration*	41.4 ± .2	44.7 ± .1	104.9 ± .3	319.6 ± 1.8	375.6 ± 4.0	67.4 ± .3	1485 ± 29	1715 ± 4.4	325.4 ± 20
Image data disk write	0.7 ± 0	0.5 ± 0	1.8 ± 0	1.7 ± 0	1.0 ± 0	1.0 ± 0	3.3 ± 0	2.1 ± 0	2.1 ± 0
Spatial smoothing	7.1 ± .0	4.6 ± 0	13.5 ± 0	29.9 ± .0	20.0 ± 0	21.7 ± 0	27.7 ± .2	82.3 ± 0	51.0 ± 9.2
T test calculation	1.7 ± .0	0.8 ± 0	2.0 ± 0	7.1 ± .0	3.2 ± 0	5.4 ± 0	42.1 ± .2	14.6 ± 0	20.0 ± 0
Standard processing—image data	277	273	285	2620	2604	2610	7842	7851	7829
Standard processing—raw data (* not in "standard" analysis)	**95**	**88**	**150**	**473**	**515**	**214**	**1832**	**1973**	**624**

Note. Processing times, in milliseconds per image, for three different size echo-planar gradient echo acquisition matrices processed during data acquisition. Matrix sizes ranged from the simplest 128 × 64 matrix to 512 × 128 2-shot mosaic "catch and hold" acquisitions with partial *k*-space in the phase direction. Times were compared for three different computer systems: SGI O2, SGI Power Challenge (SGI-PC) using either 1 CPU or 6 CPU's in parallel, and Sun Ultra 2 using 1 CPU. The bold line near the top shows the scanner's maximum data acquisition rate per image for each matrix size. The bold line at the bottom shows fScan's mean processing time per image when starting with raw data from the scanner. "Standard" processing included network data transfer, motion calculation, spatial smoothing, and *t* test, plus image reconstruction on either the scanner or using fScan. Each time represents the mean of three successive runs.

TABLE 2

Real-Time Latency

Matrix size	128 × 64	256 × 128		512 × 128	
Image size	64 × 64	128 × 128		256 × 256	
Acquisition mode	1-shot, Full K	1-shot, Partial K		2-shot mosaic, Partial K	
Number of images acquired in 480-s scan	2240	960		480	
Acquisition rate per image (max duty cycle)	**214 ms**	**500 ms**		**1000 ms**	
Computer	SGI-PC	SGI-PC	SGI-PC	SGI-PC	SGI-PC
Num CPUs	1	1	2	1	3
Total scan time	480.0	480.0	480.0	480.0	480.0
Total analysis time	490.3 ± 5.9	570.4 ± 17.7	487.5 ± 4.8	968.0 ± 27.4	489.0 ± 3.6
Delay from 8th MR image to stability plot display	8.0 ± 3.5	12.3 ± 5.1	17.2 ± 0.5	37.7 ± 2.5	8.0 ± 1.7
Delay from end of 1st cycle to T-map display	36.3 ± 5.1	51.6 ± 9.2	7.0 ± 4.8	103.3 ± 7.0	30.7 ± 1.5
Delay from end of scan to end of analysis	**10.3 ± 5.9**	90.4 ± 17.7	**7.5 ± 4.8**	488.0 ± 27.4	**9.0 ± 3.6**

Note. Delay times, in seconds, between the moment at which different stages of EPI data were acquired on the scanner and when that data appeared within fScan's displayed results. The same data acquisition sizes as in Table 1 were tested, and in this case all processing was performed on the SGI Power Challenge computer. For each data set, processing involved transferring raw data from the scanner, reconstructing images, storing images on disk, spatial smoothing, image stability calculations, *t* test calculations, and displaying the data. Times for each matrix size are given using a single CPU and, where necessary, with the minimum number of CPU's running in parallel to keep up with data acquisition. The bold times at the bottom indicate those scans in which the complete analysis was completed within 10 s of the end of the scan. Each time represents the mean of three runs.

TABLE 3

Near Real-Time Performance

Matrix size	128 × 64			256 × 128			512 × 128		
Image size	64 × 64			128 × 128			256 × 256		
Image voxel size (mm × mm × mm)	3.1 × 3.1 × 3.0			1.6 × 1.6 × 3.0			0.8 × 0.8 × 3.0		
Acquisition mode	1-shot, full k			1-shot, partial k			2-shot, partial k		
Computer	SGI-O2	SGI-PC	Sun Ultra	SGI-O2	SGI-PC	SGI-PC	SGI-O2	SGI-PC	SGI-PC
Number of CPUs	1	1	1	1	1	6	1	1	6
Read image data from disk	1.7 ± .2	0.7 ± .3	1.3 ± 0	3.9 ± .2	1.6 ± .2	2.1 ± 1.3	14.3 ± .4	4.5 ± .4	5.2 ± .2
Filters:									
Detrend	11.7 ± .0	11.8 ± .1	13.6 ± 0	51.8 ± .5	60.9 ± .4	20.4 ± .2	230.4 ± .4	243.1 ± 0	68.3 ± 1.4
Regression filter—1 variable	9.4 ± .0	9.3 ± .2	9.1 ± 0	44.5 ± .0	52.0 ± .3	18.9 ± .1	203.1 ± 1.0	204.5 ± 0	60.4 ± .6
Regression filter—3 variables	12.3 ± .0	10.9 ± 0	13.9 ± 0	56.7 ± .4	57.2 ± .3	19.9 ± .2	248.5 ± .4	233.7 ± .2	66.8 ± .4
Maps:									
Correlation	5.7 ± .0	5.5 ± 0	8.1 ± 0	27.9 ± .4	27.8 ± .6	14.1 ± .1	120.6 ± .2	102.2 ± .8	39.5 ± .4
Cross-correlation	39.7 ± .0	30.8 ± .1	110.3 ± 0	159.0 ± 0	126.4 ± .4	24.0 ± 0	652 ± .2	500.2 ± 1.4	137.7 ± .8
Regression map—1 variable	9.8 ± 0	8.9 ± 0	10.3 ± 0	44.7 ± .1	49.0 ± 0	11.5 ± .1	189.4 ± 1.0	199.8 ± .7	45.2 ± .3
Regression map—3 variables	13.7 ± .5	11.7 ± .1	16.6 ± 0	58.8 ± .6	56.4 ± .3	12.7 ± .3	251.6 ± 1.9	232.7 ± 1.7	52.6 ± .7
Read raw data from disk	9.9 ± .4	10.0 ± .2	3.9 ± .1	14.6 ± .2	18.2 ± .6	17.0 ± 1.1	37.7 ± .2	34.5 ± 2.3	33.5 ± 1.6
Raw Filters:									
Detrend	85.8 ± .2	70.0 ± .9	114.4 ± 1.1	188.8 ± .3	186.5 ± 3.0	57.0 ± 1.3	396.4 ± .4	372.0 ± 1.2	139.9 ± 13
Regression filter—1 variable	75.1 ± .1	58.6 ± .6	86.7 ± .2	162.2 ± .4	156.6 ± .8	54.2 ± .3	335.4 ± 1.6	313.3 ± .6	111.4 ± .4
Regression filter—3 variables	80.8 ± .3	62.1 ± .2	96.6 ± .3	179.3 ± .1	162.2 ± 1.4	68.4 ± 4.7	376.6 ± .8	329.9 ± 1.0	132.5 ± 11

Note. Times for fScan processing of MR time course data one voxel at a time, starting after the completion of MR data acquisition. The same matrix sizes and computers were compared as in Table 1. All times are normalized to the number of images and expressed as mean time per image, in ms. Each time represents the mean of three successive runs.

CIGAL and/or fScan are available to interested users from the author.

A significant feature of this fMRI software is that it achieves real-time fMRI performance without the need for highly specialized hardware. CIGAL, for example, runs on either Macintosh or IBM compatible personal computers and has no scanner-specific features. Its analog and digital data input features are straightforward and could be accommodated by a variety of different peripheral data acquisition interfaces. fScan runs on Unix systems that support X-windows. It contains four scanner-dependent routines: one that reads the MR header information, two that read the MR data (one for raw and one for image data), and one that performs image reconstruction if using the raw data. Although these routines are clearly important for achieving real-time performance, they can be readily modified for other pulse sequences or different hardware or omitted altogether for analyses performed off-line.

Sufficient computing power for achieving the results shown in this study was obtained from standard commercially available computers. By manipulating the display using fast video lookup table changes, asynchronous file I/O, and efficient image transfer operations, CIGAL's parallel processor could routinely achieve smooth continuous video animation and behavioral or physiological response monitoring on a standard personal computer using only a fraction of the CPU's time. Although our SGI Power Challenge computer with eight parallel CPU's could achieve faster fScan processing times than most standard workstations, the results demonstrate that even for the largest matrices only two or three parallel CPUs were actually needed to obtain real-time performance on our GE scanner (see Table 2). Many moderately priced workstations offer such parallel processing. Alternatively, the analysis could be explicitly split among separate workstations. Rapid transfer of the MR data to the analysis computer was also an important performance requirement and was satisfied in this study using CDDI network interface hardware. Transfer of the paradigm data and command files from the paradigm computer to the analysis computer was less time-critical and was achieved using slower 10BaseT Ethernet network connections.

The ability to integrate physiological and behavioral

variables into the on-line analysis is another important aspect of this approach. As both cardiac and respiratory oscillations produce periodic changes in blood oxygenation, local tissue motion, and perhaps gross motion-induced susceptibility artifacts they can give rise to significant variations in the size of the MR signal (Jezzard *et al.*, 1993; Weisskoff *et al.*, 1993; Hu *et al.*, 1995; Le and Hu, 1996; Noll and Schneider, 1994). The present study demonstrates that cardiac and respiratory signals can be trivially and routinely recorded for any fMRI paradigm without needing any extra software or hardware setup. Moreover, these data can be incorporated into the statistical image analysis within minutes of the completion of each scan.

Although the current study does not represent the state of the art in terms of instantaneous real-time performance in MR image analysis, it does represent an important step toward making real-time fMRI truly practical. Cox *et al.* (1995), for example, have demonstrated better real-time performance by taking the MR data directly from the scanner's backplane or RAM memory, thus avoiding FScan's few second delays in reading MR data from the scanner via disk files. Similarly, others (Woods *et al.*, 1992; Friston *et al.*, 1996; Eddy *et al.*, 1996) have developed better image registration software than the simple alignment offered by fScan. For most applications, however, a few seconds latency between each MR acquisition and the updated analysis display is probably not a significant problem and scans that are seen to have significant head motion in the on-line analysis can still be aligned by a more complete three-dimensional registration in postprocessing. The emphasis in the current work has not necessarily been to provide the technically most advanced real time features, but to address ways that can make fMRI a more reliable and effective technique. The goal has thus been to design paradigm software that provides very good real-time control and data monitoring so that one has optimal information about what the subject does within the scanner, along with analysis software that is capable of displaying the results and identifying potential problems fast enough to improve reliability by thoroughly evaluating each scan in very close to real-time.

Improving reliability is clearly an important consideration for fMRI, especially in a clinical setting in which successful results must be obtained for virtually every patient if fMRI is to be a useful procedure for diagnosis or treatment. Anxious patients cannot be expected to perform unfamiliar tasks correctly for every scan. Patients can also not reasonably be kept in the scanner until an off-line analysis has been completed, nor would it be practical to bring them back for repeat scans if subsequent data processing indicated a problem. To ensure a usable result, therefore, patient performance must be carefully monitored during each scan and the MR images must be analyzed on-line so that unsuccessful scans can be repeated immediately if necessary. Rapid data processing is also important in order to ensure that the functional activation maps are available in time to contribute significantly to patient care decisions.

Comprehensive real-time analysis is also useful for achieving maximum sensitivity in research applications involving high spatial or temporal resolution studies. Preliminary low resolution scans analyzed immediately can be used to localize functional areas and allow accurate slice prescription for subsequent more detailed scans. In addition, improving sensitivity through physiological noise filtering and on-line detection of major noise problems is likely to be even more important for higher resolution studies that have intrinsically lower signal to noise ratios, and therefore greater noise susceptibility. Real-time data processing can also be used to improve scanning efficiency by allowing long scans to be prescribed and then terminated interactively as soon as the analysis indicates a statistically significant result.

Overall, these results demonstrate that fMRI analysis quality does not need to be compromised for processing speed. By using efficient software and taking advantage of the speed of current computer processors and network technology, functional imaging studies can accurately control and monitor a wide range of important physiological and behavioral variables and provide immediate comprehensive statistical analysis of brain activation. Ongoing improvements to the processing algorithms as well as the availability of ever faster computer hardware should ensure that both the speed and quality of such real-time methods will continue to improve. Such advances should help to significantly improve the reliability of fMRI for both research and clinical applications.

ACKNOWLEDGMENTS

The author would like to thank Dr. Keith Thulborn for support and encouragement, Drs. Doug Noll and Lalith Talagala for many useful discussions, Dr. William Eddy for providing the ghost tuning algorithm, and Denise Davis for help with scanning and practical insights on fMRI. This study was supported in part by NIH grants R29NS37746 and PO1NS35949.

REFERENCES

Cohen, M. S., Dubois, R. A., and Scheding, W. L. 1998. Rapid artifact detection and correction for real-time fMRI. *NeuroImage* **7:**S564.

Cox, R. W., Jesmanowicz, A., and Hyde, J. S. 1995. Real-time functional magnetic resonance imaging. *Magn. Reson. Med.* **33:**230–236.

Eddy, W. F., Fitzgerald, M., and Noll, D. C. 1996. Improved image registration by using Fourier interpolation. *Magn. Reson. Med.* **36:**923–931.

Engel, S. A., Glover, G. H., and Wandell, B. A. 1996. Retinotopic organization in human visual cortex and the spatial precision of functional MRI. *Cerebral Cortex* **7:**181–192.

Frank, J. A., Ostuni, J., Yang, Y., Shiferaw, Y., Patel, A., Qin, J., Mattay, V., Lewis, B. K., Levin, R. L., and Duyn, J. H. 1998. A technical solution for an interactive fMRI examination: Application to a physiological interview and cerebral psychology. *NeuroImage* **7:** S566.

Friston, K. J., Williams, S., Howard, R., Fracowiak, R. S. J., and Turner, R. 1996. Movement-related effects in fMRI time-series. *Magn. Reson. Med.* **35:**346–355.

Goodyear, B. G., Gati, S. J., and Menon, R. S. 1997. The functional scout image: Immediate mapping of cortical function at 4 Tesla using receiver phase cycling. *Proc. Soc. Mag. Reson. Med.* 15th Annual Meeting, p. 1628.

Hu, X., Le, T. H., Parrish, T., and Erhard, P. 1995. Retrospective estimation and correction of physiological fluctuation in functional MRI. *Magn. Reson. Med.* **34:** 201–212.

Jezzard, P., Le Bihan, D., Cuenod, C., Pannier, L., Prinster, A., and Turner, R. 1993. An investigation of the contribution of physiological noise in human functional MRI studies at 1.5 Tesla and 4 Tesla. *Proc. Soc. Magn. Reson. Med.*, 12th Annual Meeting, p. 1392.

Le, T. H., and Hu, X. 1996. Retrospective estimation and correction of physiological artifacts in fMRI by direct extraction of physiological activity from MR data. *Magn. Reson. Med.* **35:** 290–298.

Noll, D. C., and Schneider, W. 1994. Theory, simulation, and compensation of physiological motion artifacts in functional MRI. In *IEEE Int. Conf. on Image Proc.*, Austin, TX.

Posse, S., Schor, S., Gembris, D., Muller, E., Peyerl, M., Krocker, R., Grosse-Ruyken, M. L., Elghawaghi, B., and Taylor, J. G. 1998. Real-time fMRI on a clinical whole body scanner: Single-event detection of sensorimotor stimulation. *NeuroImage* **7:**S567.

Purves, D., and Voyvodic, J. T. 1987. Imaging mammalian nerve cells and their connections over time in living animals. *Trends Neurosci.* **10:** 398–404.

Thulborn, K. R., McCurtain, B., Voyvodic, J., Chang, S., Gillen, J., and Sweeney, J. A. 1996. Functional MRI: The environment and technology for clinical application. In *Proc. European Seminars in Diagnostic and Interventional Imaging: Functional MRI*, No. 24.

Voyvodic, J. T. 1986. A general purpose image processing language (IMAGR) facilitates visualizing neuronal structures in fixed tissue and in vivo. *Soc. Neurosci. Abstr.* **12:**390.

Voyvodic, J. T. 1996. Real-time fMRI paradigm control software for integrating stimulus presentation, Behavioral and physiological monitoring, and statistical analysis. *Proc. Soc. Mag. Reson. Med. 15th Annual Meeting*, p. 1835.

Weisskoff, R. M., Baker, J., Belliveau, J., Davis, T. L., Kwong, K. K., Cohen, M. S., and Rosen, B. R. 1993. Power Spectrum Analysis of Functionally-Weighted MR Data: What's in the Noise? *Proc. Soc. Magn. Reson. Med.*, 12th Annual Meeting, p. 7.

Woods, R., Cherry, S., and Mazziota, J. 1992. Rapid Automated Algorithm for Aligning and Reslicing PET Images. *J. Comput. Assist. Tomogr.* **16:**620–633.