JOURNAL OF MAGNETIC RESONANCE **134,** 138–141 (1998)
ARTICLE NO. MN981498

Real-Time Image Reconstruction and Display System for MRI Using a High-Speed Personal Computer

Tomoyuki Haishi and Katsumi Kose

Institute of Applied Physics, University of Tsukuba, Tsukuba 305, Japan

Received November 12, 1997; revised May 6, 1998

A real-time NMR image reconstruction and display system was developed using a high-speed personal computer and optimized for the 32-bit multitasking Microsoft Windows 95 operating system. The system was operated at various CPU clock frequencies by changing the motherboard clock frequency and the processor/bus frequency ratio. When the Pentium CPU was used at the 200 MHz clock frequency, the reconstruction time for one 128 × 128 pixel image was 48 ms and that for the image display on the enlarged 256 × 256 pixel window was about 8 ms. NMR imaging experiments were performed with three fast imaging sequences (FLASH, multishot EPI, and one-shot EPI) to demonstrate the ability of the real-time system. It was concluded that in most cases, high-speed PC would be the best choice for the image reconstruction and display system for real-time MRI.

Key Words: **MRI; image reconstruction; real-time MRI; fast imaging; EPI.**

INTRODUCTION

Combination of a rapid imaging sequence, a fast image reconstruction, and a display hardware system has enabled real-time observation of NMR images. Up to now, various approaches to such fast image processing hardware have been reported (*1–10*). The hardware systems used in these studies are roughly divided into three categories: specially designed hardware (*1–3, 6, 7*), UNIX workstations (*4, 5, 9*), and personal computers (*8, 10*).

In the previous study (*10*), we reported that real-time MRI systems can be built using personal computers, though some problems remained. The first was that real-time data acquisition was sometimes interfered or interrupted by mouse or keyboard operations during time-critical imaging sequences (e.g., fast repetition FLASH sequences). The second was that the time required for the image display on Windows was much longer than when a hardware display board was used in a MS-DOS PC system. These problems mainly came from the fact that the computer program was written in 16-bit code and not optimized for the 32-bit multitasking operating system of Microsoft Windows 95, because Windows 95 was not preemptive when one or more 16-bit Windows applications were running.

In the present study, we have optimized the data-acquisition and image display programs for the operating system using 32-bit code and critically evaluated the system performance using imaging experiments. As a result, we have succeeded in constructing a stable and fast enough image reconstruction and display system for real-time MRI under an operating system by developing a computer program.

HARDWARE AND SOFTWARE OF THE SYSTEM

Figure 1 shows the block diagram of the system developed in this study. This system was built on a personal computer with a high performance microprocessor (Pentium, Intel Corp.) running under Windows 95. The microprocessor was operated at various clock frequencies between 75 and 200 MHz by changing the clock frequency of the motherboard (50, 60, 66 MHz) and the processor/bus frequency ratio (1.5, 2.0, 2.5, 3.0). For NMR signal acquisition, a commercial ADC board (PC-414G3, DATEL) was connected to the CPU via the ISA bus. The ADC board has a 32 Kword FIFO buffer memory and is able to digitize two-channel analog signals simultaneously at a 14-bit resolution up to the rate of 1 MHz.

However, several software problems had to be solved in developing the real-time image reconstruction and display program on Windows 95.

The first problem was how to write the data-acquisition program under Windows 95. This was because most major multitasking operating systems prohibit direct user access to the I/O devices. For such cases, it was required to develop or purchase driver software for the I/O devices. For Windows 95, however, direct user access to the I/O port is permitted by using standard I/O functions such as inpw() and outpw(), which can be used in most C compilers. We developed the data-acquisition program using these I/O functions.

The second problem was how to write the fast image display program under Windows 95. The standard method to display images in windows was to use the SetPixel() function, which we had used in the previous study (*10*). This function was, however, not fast enough for the image display (about 120 ms for one 128 × 128 pixel image on a 256 × 256 pixel window). Instead, we used the StretchDIBits() function, which is used to

1090-7807/98 $25.00

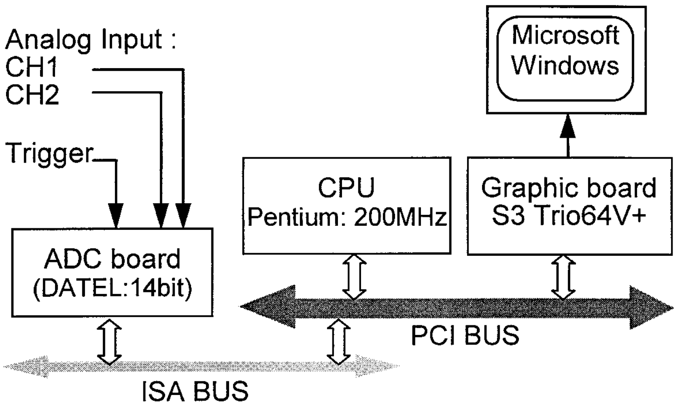

FIG. 1. Block diagram of the real-time NMR image reconstruction and display system using a personal computer running under Windows 95.

transfer image data from the main memory to the video memory with minimum system overhead. The use of this function drastically shortened the image display time as presented in a later section.

The third problem was how to write the time-critical data-acquisition program, such as that for the fast repetition (<5 ms) FLASH sequence under the multitasking operating system of Windows 95, in which several programs were running apparently at the same time. We considered two approaches to this problem.

The first approach was to develop a data-acquisition program as a 16-bit Windows application and prohibit any interruption from other programs during the data-acquisition process (*10*). This was possible because such a 16-bit Windows application could not be interrupted even under Windows 95 when it was running in its processing routine, which was performed in the WndProc() function of the Windows program. However, we had experienced a "hang-up" phenomenon caused by the loss of sampling trigger pulses and interference or interruptions by mouse or keyboard operations when the program went back to the WinMain() function of the Windows program.

The second approach was to develop a data-acquisition program as a 32-bit Windows application with a large capacity FIFO buffer memory for the acquired data. This was because the 32-bit application could be interrupted by other processes at any time, but the acquired data were not lost in most cases. By using this approach, a stable real-time data-acquisition, image reconstruction, and display program was developed as described in a later section.

Figure 2 shows the structure of the image reconstruction and display program which has overcome all of the problems described above. The program has the standard Windows program structure composed of WinMain() and WndProc(). External events such as a mouse click or keyboard operation are detected in the WinMain() function and the processing for the events is performed in the WndProc() function. Because the program is written in 32-bit code, a mouse click or keyboard operation can interrupt this program at any time, even during the WndProc() function, but the acquired data are not lost in most cases because of the large-capacity FIFO buffer memory.

EXPERIMENTS

The imaging experiments were performed using a homebuilt NMR imaging system with a vertical wide-bore superconducting magnet (field strength: 4.74 T, ^{1}H resonance frequency: 202 MHz) and an imaging probe (*10*). A water phantom was prepared as follows: An acrylic short pipe (14 mm O.D. and 10 mm in length) with an acrylic rod (3 mm in diameter) perpendicularly penetrating the pipe was placed in an acrylic pipe container (20 mm O.D. and 16 mm I.D., the bottom closed) with the pipe axes perpendicular to each other. The container was filled with $CuSO_4$ solution. The phantom, moving vertically at a constant velocity (5.7 mm/s) driven by a DC-controlled motor, was imaged in a vertical cross-section (4 mm slice thickness) and used to demonstrate the ability of the real-time MRI system.

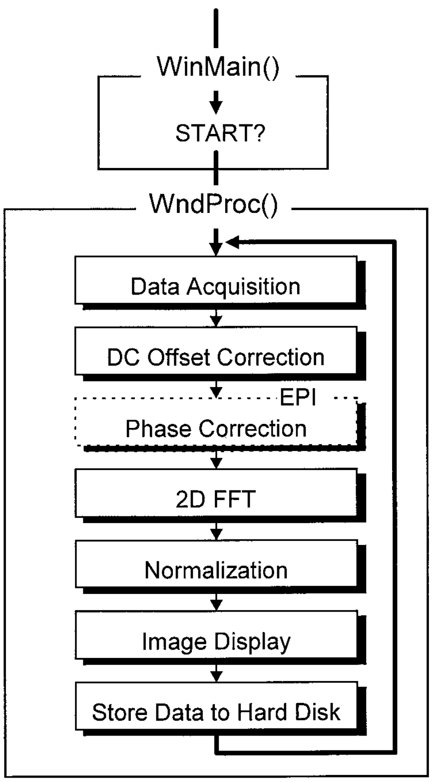

FIG. 2. The structure of the real-time NMR image data-acquisition, reconstruction, and display program on Windows 95. For EPI, phase shifts due to eddy current are corrected before the 2DFT operation (*3*). Because the program is written in 32-bit code, a mouse click or keyboard operation can interrupt this program at any time, even during the WndProc(), but the acquired data are not lost in most cases because of the large-capacity FIFO buffer memory. This program was developed using Microsoft Visual C++ compiler Version 4.0 under Windows 95.

The performance of the real-time image reconstruction and display system was measured using the following fast imaging sequences: FLASH (TR = 4.82 ms, TE = 2.70 ms), multishot EPI (TR = 100 ms, TE = 48 ms, four segments), and one-shot EPI (TR = 80 ms, TE = 48 ms). The image matrix size was 128 × 128 for FLASH and multishot EPI, and 64 × 64 for one-shot EPI. In all cases, the images were displayed in the windows with 256 × 256 pixels in an 8-bit gray scale.

To measure the image reconstruction and display times including the disk saving time, the image refresh intervals in the continuous FLASH image acquisition sequence were measured with a FIFO buffer operation disabled during the image processing. The FIFO buffer was disabled because if the buffer was enabled, the image reconstruction and display would be performed even during the FLASH data-acquisition period. The minimum time for 100 image refresh intervals was measured using a stopwatch, and the 100 FLASH image acquisition time was subtracted to compute the image reconstruction and display time. For comparison, we measured the processing time using personal computers with the PentiumPro CPU (Intel Corp.) and the Pentium II CPU (Intel Corp.).

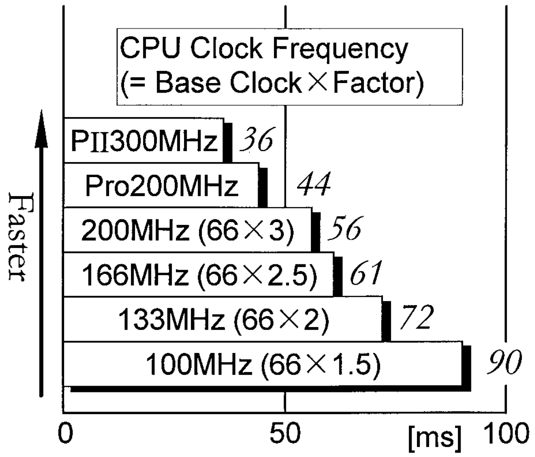

FIG. 3. Image reconstruction and display times including the disk save times for one 128 × 128 pixel image (FLASH) when the Pentium CPU was used at various clock frequencies. The numbers in the parenthesis show the motherboard clock frequency times the processor/bus frequency ratio. “Pro200MHz” and “PII300MHz” mean the processing times when the PentiumPro processor was used at the 200 MHz clock frequency and when the PentiumII processor was used at the 300 MHz clock frequency.

RESULTS AND DISCUSSION

The real-time image reconstruction and display program described in Fig. 2 worked very well even for the time-critical image acquisition of thousands of images. Because the program was written in 32-bit code, the multitasking of Windows 95 was preemptive, and other 32-bit applications such as the word processor and Web browser software could be used even during the data-acquisition period.

Figure 3 shows the image reconstruction and display times including the disk saving times for one FLASH image measured at various CPU clock frequencies when the motherboard clock frequency was 66 MHz. In the fastest case for the Pentium CPU (200 MHz clock frequency), the whole processing time was 56 ms; the image reconstruction time was about 48 ms, and the image display time was about 8 ms. The image reconstruction time of about 48 ms was estimated using a separate image reconstruction program. However, we could not estimate the time required for disk saving because the task for disk saving was performed as a system process.

The time of 8 ms for the image display corresponds to a data transfer rate of 8 Mbytes/s from the main memory to the video memory. This transfer rate is reasonable for the PCI bus ability. We also found that the image display time did not depend on the performance of the display card because the largest part of the StretchDIBits() function is to transfer the image data from the main memory to the video memory via the CPU and the PCI bus.

In the previous study (*10*), 120 ms was required for the image reconstruction and nearly the same time was also required for the image display when the Pentium CPU with the 133 MHz clock frequency was used. In the present study, we have achieved about 1.7 times faster image reconstruction time and about 10 times faster image display time even at the same CPU clock frequency by using the 32-bit program optimized for Windows 95. The use of the PentiumPro processor with the same clock frequency (200 MHz) further shortened the processing time by about 20%.

Figure 4 shows four successive instants of the images acquired using the three pulse sequences. Although motion artifacts are present in the images taken with FLASH and multishot EPI, the continuous translation of 5.7 mm/s is clearly visualized.

In conclusion, we have developed a real-time MR image reconstruction and display system using a high-speed personal computer running under Windows 95. The program was optimized for the 32-bit multitasking operating system and achieved enough processing speed for several fast imaging sequences. Because personal computers are much less expensive than specialized hardware systems and workstations, and the computer programs for PCs can be developed much more easily than those for workstations, we believe that the personal computer is the best choice for image reconstruction and display systems in constructing real-time MRI. Because this system requires only three analog lines (two-channel detector outputs and its sampling trigger) for connection, it can be used with many existing MRI systems. Connecting to MRI systems with direct digital RF detection circuits is planned as a future extension of this study.

ACKNOWLEDGMENTS

We thank Mr. Naotaka Adachi for a computer simulation of the moving phantom. This work was supported in part by a Grant-in-Aid for Scientific Research from the Ministry of Education, Science, and Culture in Japan.

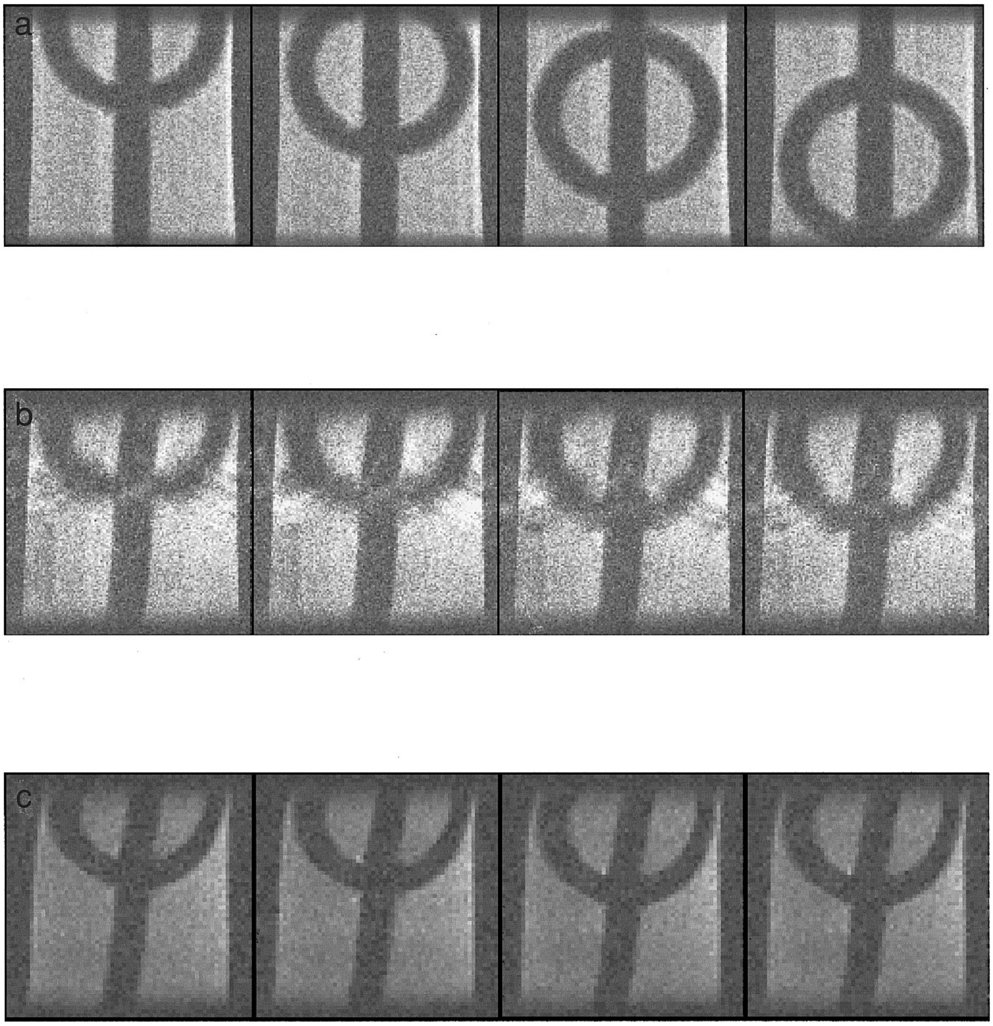

FIG. 4. Images of a water phantom moving at 5.7 mm/s taken with three pulse sequences. (a) FLASH, 128 × 128 pixel images every 638 ms; (b) multishot EPI, 128 × 128 pixel images every 100 ms; (c) one-shot EPI, 64 × 64 pixel images every 80 ms. FOV: (19.2 mm)^2, slice thickness: 4 mm. In (b) and (c), phase distortion due to eddy current effects was corrected in real time before the 2D FFT operations (*3*).

REFERENCES

1. R. C. Wright, S. J. Reiderer, F. Fazaneh, P. J. Rossman, and Y. Liu, Real-time MR fluoroscopic data acquisition and image reconstruction, *Magn. Reson. Med.* **12,** 407–415 (1989).
2. D. M. Kramer, C. Hawrysko, D. A. Ortendahl, and M. Minaise, Fluoroscopic MR imaging at 0.064 Tesla, *IEEE Trans. Med. Imaging* **10,** 358–361 (1991).
3. K. Kose and T. Inouye, A real-time NMR image reconstruction system using echo-planar imaging and a digital signal processor, *Meas. Sci. Technol.* **3,** 1161–1165 (1992).
4. C. S. Potter, C. D. Gregory, H. D. Morris, Z.-P. Liang, and P. C. Lauterbur, The NEUROSCOPE: an interactive system for real-time functional MRI of the brain, in Proc. SMR, 2nd Annual Meeting, San Francisco, 1994, p. 835.
5. R. W. Cox, A. Jesmanowicz, and J. S. Hyde, Real-time functional magnetic resonance imaging, *Magn. Reson. Med.* **33,** 230–236 (1995).
6. O. Ichikawa, K. Kose, and Y. Seo, Fast image reconstruction experiments using small-bore MRI for animals, *Jpn. J. Magn. Reson. Med.* **15,** 216–220 (1995).
7. A. F. Gmitro, A. R. Ehsani, T. A. Berchem, and R. J. Snell, A real-time reconstruction system for magnetic resonance imaging, *Magn. Reson. Med.* **35,** 734–740 (1996).
8. K. Kose and T. Haishi, Development of a real-time MR image reconstruction system using a high-speed microprocessor, *Jpn. J. Magn. Reson. Med.* **16,** 98–102 (1996).
9. J. Pauly, A. Kerr, C. Hardy, C. Meyer, D. Nishimura, and A. Makovski, Real-time dynamic imaging on a conventional whole body imaging system, Proceedings of the International Society of Magnetic Resonance in Medicine, 4th Annual Meeting, New York, p. 395, 1996.
10. K. Kose, T. Haishi, A. Caprihan, and E. Fukushima, Real-time NMR imaging systems using personal computers, *J. Magn. Reson.* **124,** 35–41 (1997).