Int. J. Engng Ed. Vol. 11, Nos 4 and 5, pp. 322–328, 1995
Printed in Great Britain.

0949-149X/91 $3.00+0.00
© 1995 TEMPUS Publications.

Digital Signal Processing for Video Line Rate Analysis

RULPH CHASSAING
BILL BITLER*
Roger Williams University, Bristol, RI 02809, USA

Within the past eight years, students at Roger Williams University (RWU) have implemented a wide range of applications using digital signal processing (DSP) techniques in their Senior Project course. Students' projects in communications and controls, adaptive and multirate filtering, etc., have been discussed previously. As an example, we present a video line rate analysis project using a low-cost evaluation module (EVM) based on the TMS320C30 digital signal processor. The results of this analysis, utilizing image-processing algorithms in digital filtering, averaging and edge enhancement, are demonstrated and displayed on a PC host monitor. The image-processing options are available interactively using the function keys on the PC. A module was designed to sample the video signal, with circuitry built from readily available and inexpensive parts.

EDUCATIONAL SUMMARY

1. The paper describes new training tools or laboratory concepts/instrumentation/experiments in digital signal processing applications.
2. The paper describes new equipment useful in digital signal processing laboratory, senior (design) project and control courses.
3. Senior level students are involved in the use of the equipment.
4. The new aspects are real-time applications in digital signal processing.
5. How is the material presented to be incorporated in engineering teaching? As an application example of a real-time project.
6. The text accompanying the presented material is *Digital Signal Processing with C and the TMS320C30* (Rulph Chassaing, Wiley, 1992).
7. The concepts presented have been tested in the classroom. We have concluded that real-time applications can be very highly motivating to the students.
8. Digital signal processing techniques can be used for a wide range of applications including communications, control and instrumentation.

INTRODUCTION

SINCE 1986, one section of a Senior Project course offered at Roger Williams University has been devoted exclusively to applications in digital signal processing (DSP). Initially, projects were implemented based on the fixed-point TMS320C25 digital signal processor [1]. Recent projects are based on C and the floating-point TMS320C30 digital signal processor [2,3]. During the first half of the project course, the senior project students perform some experiments in order to become familiar with the architecture and instruction set of the TMS320C30 processor, and the available support software and hardware tools. The software tools include a C compiler, a simulator, and utilities for filter design and spectral analysis. The hardware tools include a TMS320C30–based evaluation module (EVM) and several input/output alternatives. Additional experiments are performed to implement finite and infinite impulse response (FIR/IIR) filters, and sinewave generation using both simulation and real-time environments. These experiments reinforce the concepts of discrete convolution and recursive difference equation. A real-time fast Fourier transform (FFT) and adaptive filtering are then demonstrated to the students. During the second half of the course, the students are engaged in the implementation of real-time final projects using C and/or TMS320C30 code.

The senior students have implemented a wide range of real-time design projects from communications and control to neural network, using DSP techniques.

SUPPORT TOOLS

The TMS320C30 is a third-generation digital signal processor which supports both integer and floating-point operations. It has a total of 16M words of addressable memory and 2K words of on-chip RAM. With an instruction execution time of 60 ns, it is capable of performing 16.66 million instructions per second (MIPS). Since many instructions can be executed in parallel, such as

* Current address: InfiMed, New York, USA

load and store, or multiply and add instructions, the TMS320C30 can effectively perform up to 33.3 MIPS. The TMS320C30 has two serial ports. One of the serial ports is connected to an analog interface chip on board of an evaluation module (EVM). The other serial port, available through a connector on the EVM, is used to interface the TMS320C30 to the video module.

The EVM is a relatively low-cost, yet powerful DSP development system, available from Texas Instruments, Inc. It is an 8–bit half card which plugs directly into a slot on an IBM compatible PC. It includes the TMS320C30, and 16K of user static RAM. A wide range of support tools are currently available [4–6].

VIDEO LINE RATE ANALYSIS

The video signal is the electrical representation of a complete television picture. The source of the video signal is a charged-coupled device (CCD) camera. A television picture as viewed on a television receiver (monitor) is comprised of approximately 30 complete images displayed per second, generally referred to as a frame. Each frame consists of two fields of video information, which are combined together to form a complete television picture. The two fields of video information are combined in an interlaced fashion in order to conserve system bandwidth and eliminate any flicker which would be present at a rate of 30 Hz [7]. The color subcarrier frequency is 3.57 MHz, chosen as the 455th harmonic of one-half the horizontal scanning frequency. The horizontal frequency is 2(3.57 MHz)/455 = 15.734 kHz. Since there are 525 horizontal intervals in a complete television picture (a frame), the frame frequency is 15.734 kHz/525 = 30 Hz. To reduce the flicker, interlace scanning is used such that the television picture is divided into two fields of information, with each field containing every other horizontal line of

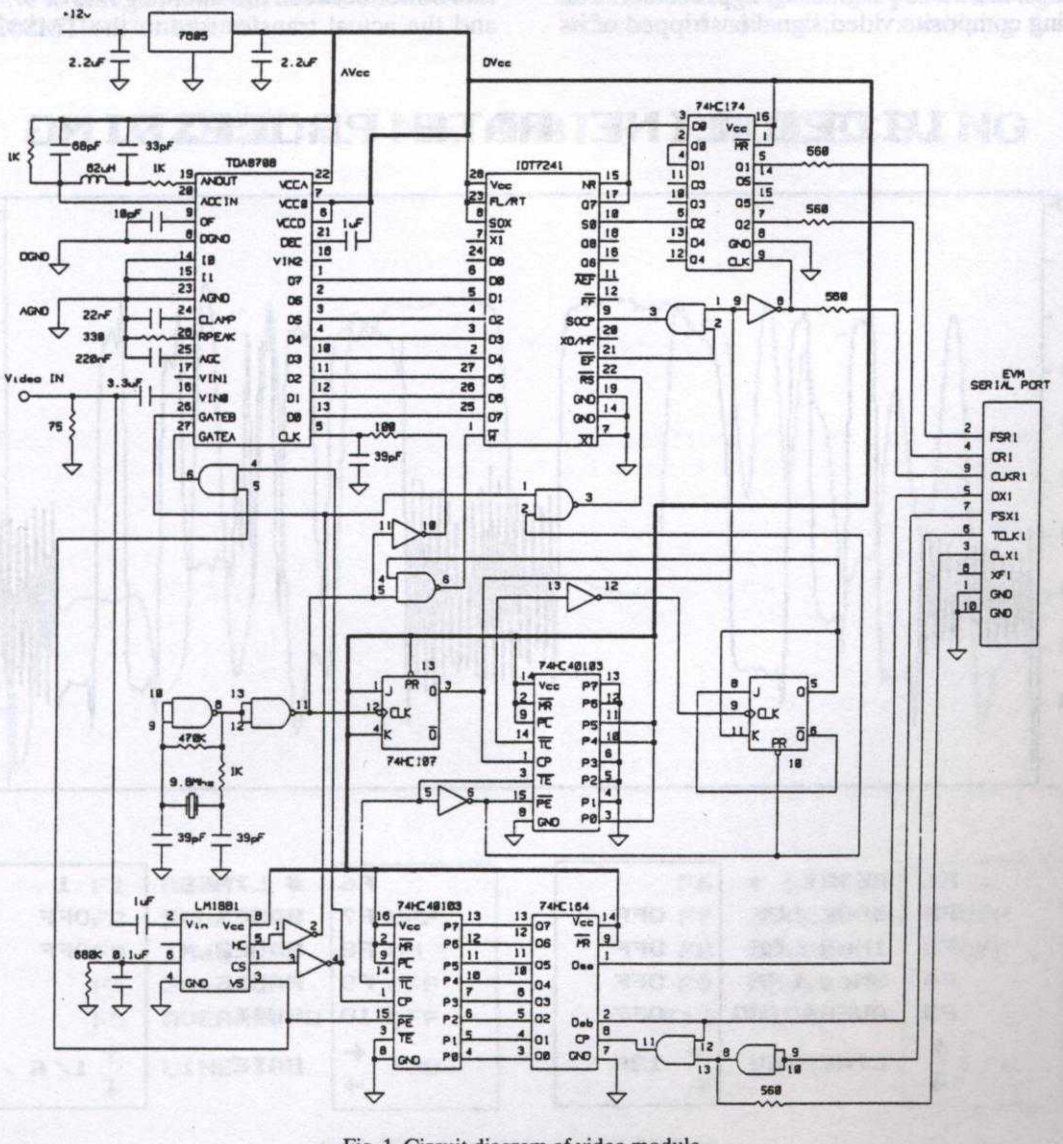


Fig. 1. Circuit diagram of video module.

information. Hence, the field frequency is 15.734 kHz/(525/2) = 60 Hz.

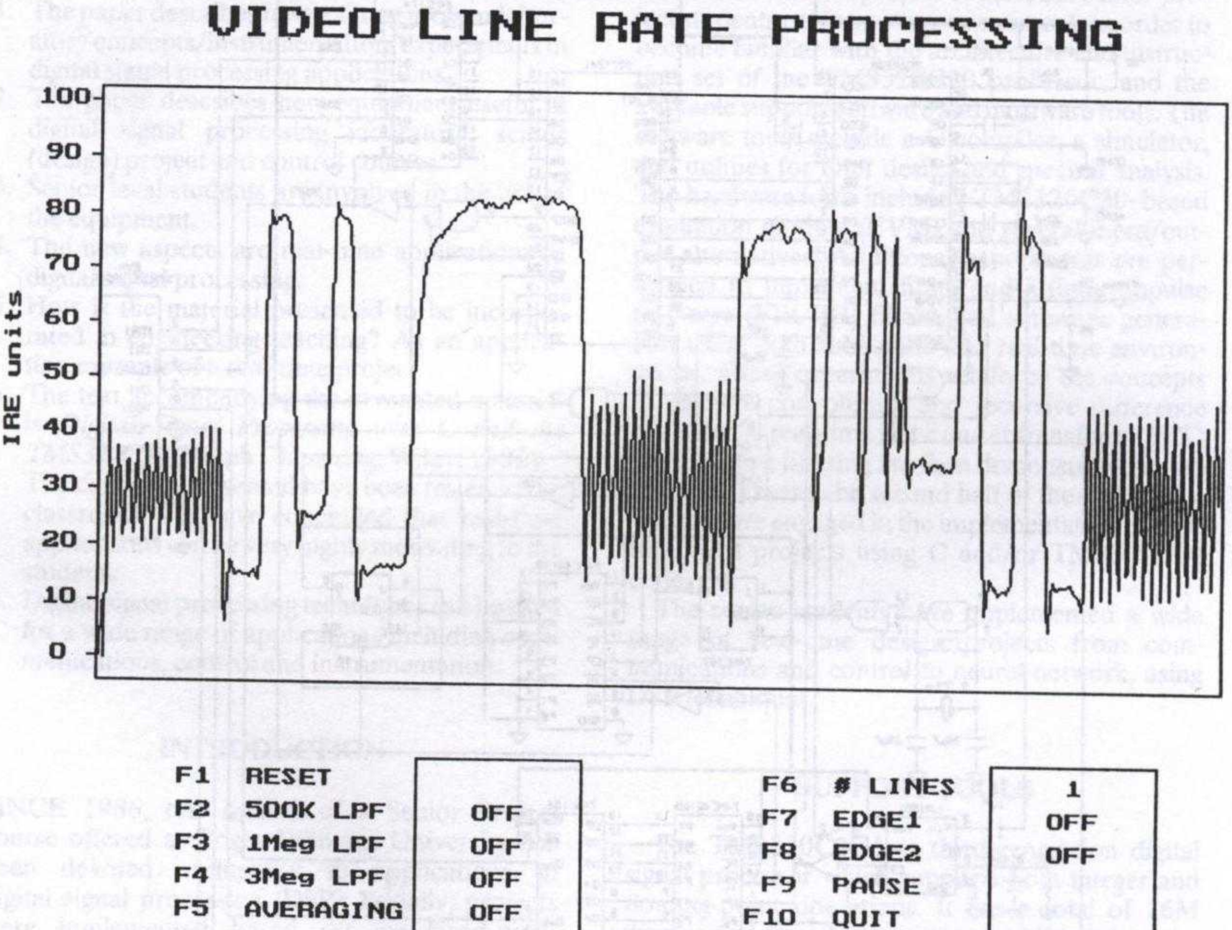
Each video field of information consists of 262.5 horizontal scanning lines, at a frequency of 15.734 kHz. The horizontal rate is 1/15.734 = 63.5 μs and consists of synchronization information occupying 11.1 μs (referred to as horizontal blanking). This leaves (63.5 − 11.1) = 52.4 μs of video information.

The desired sampling frequency of the video information contained in one horizontal line is (512 samples)/52.4 = 9.77 MHz.

The rate of the vertical interval is 262.5/(15.734 kHz) = 16.67 ms. The signal must be processed within this time interval, which includes the sampling of the desired signal, the transfer of an 8–bit digital representation of the signal to the TMS320C30, and performing any signal-processing algorithm. The results of any processing algorithm are then displayed on the PC monitor.

The circuit diagram of the video module is shown in Fig. 1. The majority of the circuitry is used to obtain proper timing between the video signal and the TMS320C30 digital signal processor. The incoming composite video signal is stripped of its synchronization pulses by a sync separator (LM1881). The composite and vertical sync are used by the timing circuitry to select the desired line of video to be sampled. The specific horizontal line of information to be sampled is transferred from the TMS320C30 on the EVM to the timing circuitry on the video module.

The video signal is connected to an 8–bit high sampling rate analog-to-digital converter (ADC). The line number to be sampled is transmitted by the TMS320C30 to a serial-to-parallel shift register where the 8–bit parallel output is loaded into a binary down-counter during the vertical interval. The horizontal sync pulse provides the clock to the binary down-counter. A second binary counter is enabled during the desired line of video and counts down to zero during the blanking interval; thus signalling the start of the video information. A J-K flip-flop, connected to a second binary down-counter outputs an 'enable pulse', asserted for approximately 52.4 μs (for 512 samples). During the video portion of the desired line, a parallel-to-serial FIFO buffer serves as a rate buffer between the sampling rate of 9.77 MHz and the actual transfer rate to the TMS320C30.



```
F1  RESET                            F6  # LINES     1
F2  500K LPF       OFF               F7  EDGE1       OFF
F3  1Meg LPF       OFF               F8  EDGE2       OFF
F4  3Meg LPF       OFF               F9  PAUSE
F5  AVERAGING      OFF               F10 QUIT
↑                                    ←
↓   LINE SEL       128               →   RATE        1/6
```

Fig. 2. Video line rate processing.

Using an 8-bit ADC, this transfer rate is (Fs/2)/8 = 610.6 kHz.

The time required to transfer the sampled line (512 samples) of video to the TMS320C30 is (512 x 8)/(Fs/2) = 0.84 ms. Since the vertical rate is 16.67 ms, the TMS320C30 has (16.67 − 0.84) = 15.8 ms to perform any image-processing algorithm and transfer the results to the PC host monitor.

IMPLEMENTATION

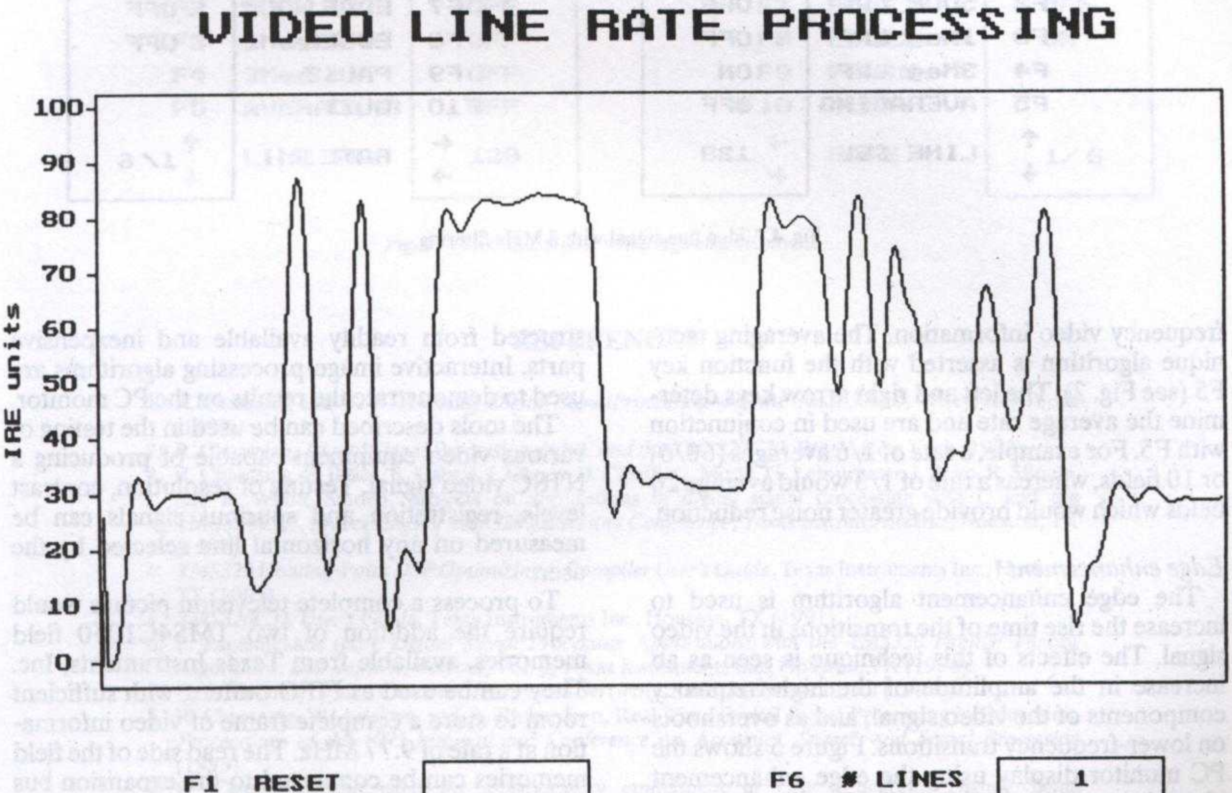
The following image-processing algorithms, implemented in C, are used for this analysis: digital filtering, image averaging and edge enhancement. Figure 2 shows a specific horizontal video line of information, obtained with the CCD camera pointing at a test chart. The video signal contains patterns that can be used to analyze the performance of a system. The desired horizontal video line is selected with the up and down arrow keys on the PC. Line number 128 was selected, as indicated in Fig. 2. The function key F6 is used to specify the number of lines to step through during the selection of the desired horizontal line.

Digital filtering

The digital filtering algorithm implements three sixth-order IIR low-pass filters, with the following cutoff frequencies: 500 kHz, 1 MHz and 3 MHz. Each filter consists of three cascaded second-order sections. Figure 3 shows the signal filtered by the 1 MHz filter (asserted with the function key F3 on the PC). While this filter reduces the noise considerably, higher-frequency information of the video signal is lost. The 3 MHz low-pass filter (asserted with the function key F4) can be used to conserve more of the higher-frequency information, although less noise would be reduced than with the lower-bandwidth filter, as shown in Figure 4.

Image averaging

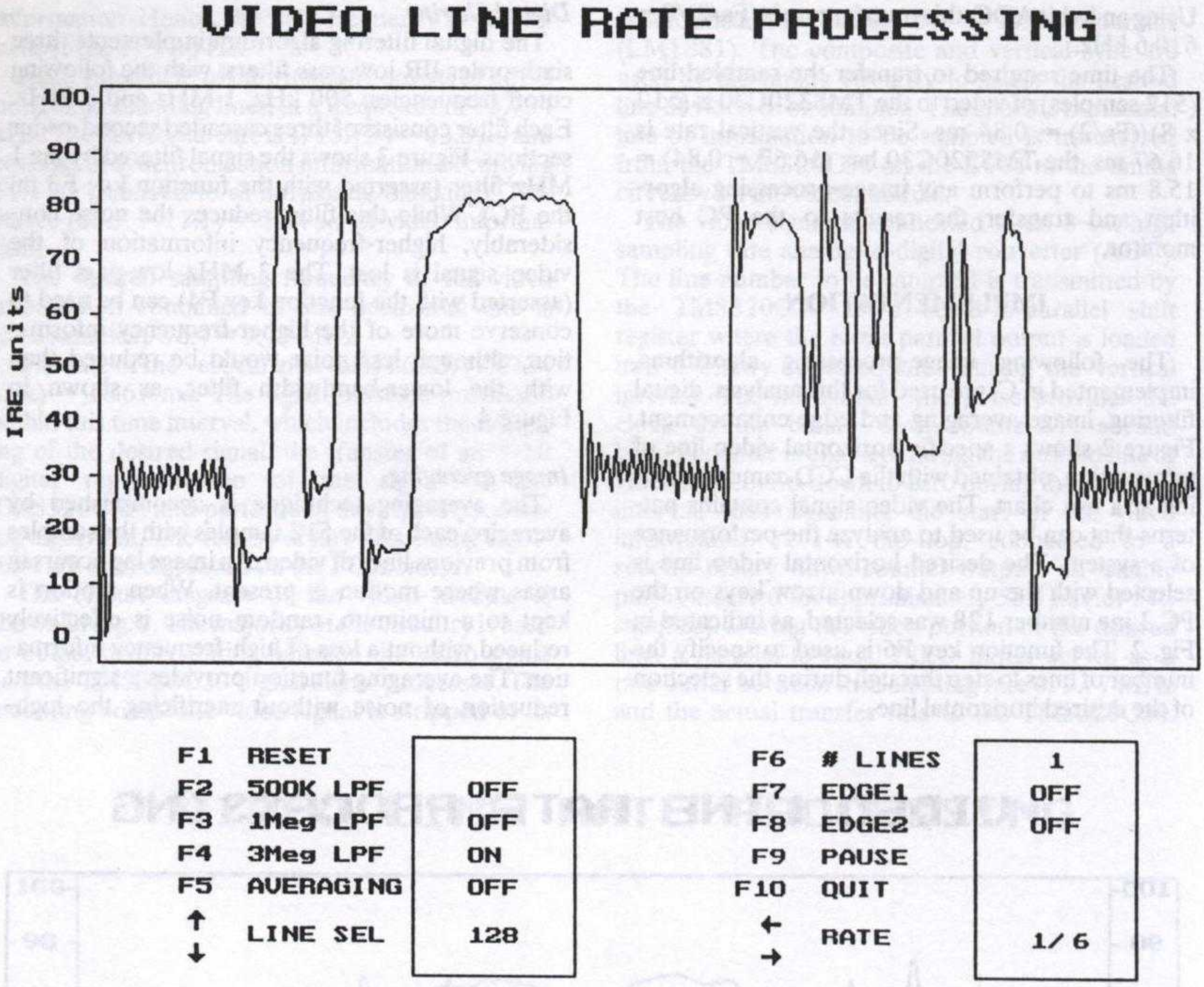
The averaging technique is accomplished by averaging each of the 512 samples with the samples from previous lines of video. An image lag occurs in areas where motion is present. When motion is kept to a minimum, random noise is effectively reduced without a loss of high-frequency information. The averaging function provides a significant reduction of noise without sacrificing the high-



VIDEO LINE RATE PROCESSING

```
F1  RESET                    F6   # LINES    1
F2  500K LPF     OFF         F7   EDGE1      OFF
F3  1Meg LPF     ON          F8   EDGE2      OFF
F4  3Meg LPF     OFF         F9   PAUSE
F5  AVERAGING    OFF         F10  QUIT
↑                            ←
    LINE SEL     128              RATE       1/6
↓                            →
```

Fig. 3. Video line signal with 1 MHz filtering.



```
F1   RESET
F2   500K LPF        OFF          F6   # LINES        1
F3   1Meg LPF        OFF          F7   EDGE1        OFF
F4   3Meg LPF        ON           F8   EDGE2        OFF
F5   AVERAGING       OFF          F9   PAUSE
↑                                 F10  QUIT
     LINE SEL        128          ←
↓                                      RATE         1/6
                                  →
```

Fig. 4. Video line signal with 3 MHz filtering.

frequency video information. The averaging technique algorithm is asserted with the function key F5 (see Fig. 2). The left and right arrow keys determine the average rate and are used in conjunction with F5. For example, a rate of 1/6 averages (60/6) or 10 fields, whereas a rate of 1/3 would average 20 fields which would provide greater noise reduction.

Edge enhancement

The edge enhancement algorithm is used to increase the rise time of the transitions in the video signal. The effects of this technique is seen as an increase in the amplitude of the high-frequency components of the video signal, and as overshoots on lower-frequency transitions. Figure 5 shows the PC monitor display using the edge enhancement algorithm, accentuating the signal at the edges (F7 provides a greater degree of edge enhancement).

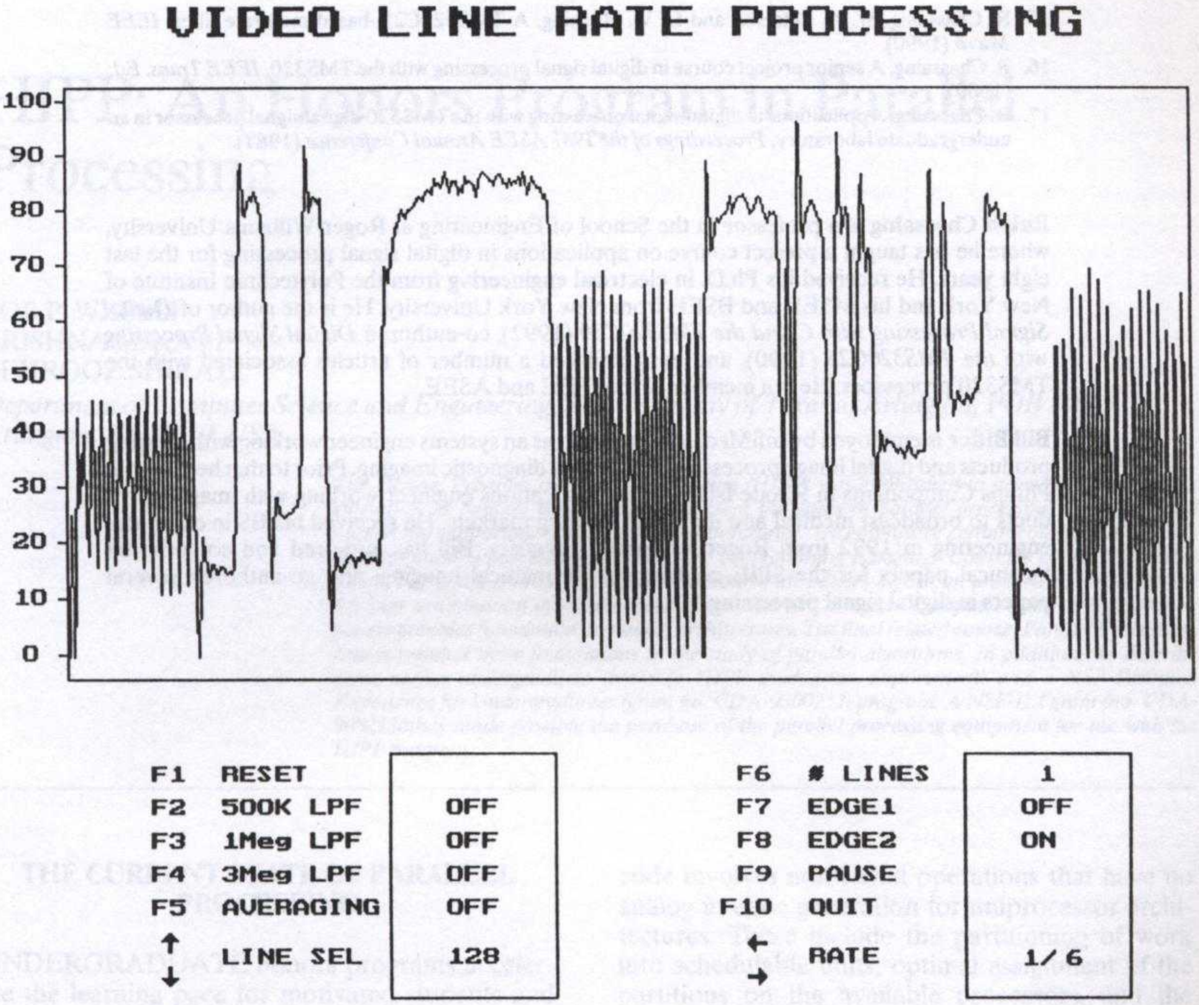
RESULTS AND CONCLUSIONS

The evaluation module based on the TMS320C30 provides a low-cost approach to video line rate analysis. The video module is constructed from readily available and inexpensive parts. Interactive image-processing algorithms are used to demonstrate the results on the PC monitor.

The tools described can be used in the testing of various video equipment capable of producing a NTSC video signal. Testing of resolution, contrast levels, registration and spurious signals can be measured on any horizontal line selected by the user.

To process a complete television picture would require the addition of two TMS4C1050 field memories, available from Texas Instruments, Inc. They can be used as FIFO buffers, with sufficient room to store a complete frame of video information at a rate of 9.77 MHz. The read side of the field memories can be connected to the expansion bus on the TMS320C30. A second buffer with two additional TMS4C1050s can be connected to the TMS320C30 to store the processed information for display on the television monitor.

Acknowledgements—The support of the National Science Foundation (NSF), through grants CSI-8650204 and USE-8851147, to develop the DSP laboratory at Roger Williams University is appreciated.



```
F1   RESET                          F6   # LINES        1
F2   500K LPF       OFF             F7   EDGE1          OFF
F3   1Meg LPF       OFF             F8   EDGE2          ON
F4   3Meg LPF       OFF             F9   PAUSE
F5   AVERAGING      OFF             F10  QUIT
↑                                   ←
     LINE SEL       128                  RATE           1/6
↓                                   →
```

Fig. 5. Video line signal with edge enhancement

REFERENCES

1. R. Chassaing and D. W. Horning, *Digital SignalProcessing with the TMS320C25,* Wiley, New York (1990).
2. R. Chassaing, *Digital Signal Processing with C and the TMS320C30,* Wiley, New York (1992).
3. R. Chassaing, B. Bitler, R. Cabral, J. Chopy, B. Cibelli, G. Monti, G. Letourneau, J. Rego, R. Siipola, and B. Wilde, Student projects on applications in digital signal processing with C and the TMS320C30, *Proceedings of TMS320 Educators Conference*, Texas Instruments Inc., Houston, TX (1992).
4. *TMS320 Floating-Point DSP Optimizing C Compiler User's Guide*, Texas Instruments Inc., Houston, TX (1991).
5. *TMS320C3X User's Guide*, Texas Instruments Inc., Houston, TX (1991).
6. P. Papamichalis (ed.), *Digital Signal Processing Applications with the TMS320 Family: Theory, Algorithms, and Implementations*, Vol. 3, Texas Instruments Inc., Houston, TX (1990).
7. G. P. McGinty, *Video Cameras: Theory and Servicing*, Sams, Indianapolis, IN (1984).
8. R. Chassaing, W. Anakwa, and A. Richardson, Real-Time Digital Signal Processing in Education, *Proceedings of the 1993 International Conference on Acoustics, Speech and Signal Processing* (1993).
9. R. Chassaing, The need for a laboratory component in DSP education, *Digital Signal Processing*,Academic Press, San Diego, CA (1993).
10. R. Chassaing and B. Bitler, Signal processing chips and applications, in *The Electrical Engineering Handbook,* CRC Press, San Diego, CA (1993).
11. R. Chassaing and B. Bitler, Adaptive filtering with C and the TMS320C30 digital signal processor, *Proceedings of the 1992 ASEE Annual Conference* (1992).
12. R. Chassaing and B. Bitler, Video line rate analysis using the TMS320C30 digital signal processor, *Proceedings of the 1992 ASEE Annual Conference* (1992).
13. R. Chassaing, B. Bitler and D. W. Horning, Real-time digital filters in C, *Proceedings of the 1991 ASEE Annual Conference* (1991).
14. D. W. Horning and R. Chassaing, IIR filter scaling for real-time digital signal processing, *IEEE Trans. Ed.* (1991).

15. R. Chassaing, W. A. Peterson and D. W. Horning, A TMS320C25-based multirate filter, *IEEE Micro* (1990).
16. R. Chassaing, A senior project course in digital signal processing with the TMS320, *IEEE Trans. Ed.* (1989).
17. R. Chassaing, Applications in digital signal processing with the TMS320 digital signal processor in an undergraduate laboratory, *Proceedings of the 1987 ASEE Annual Conference* (1987).

Rulph Chassaing is a professor in the School of Engineering at Roger Williams University, where he has taught a project course on applications in digital signal processing for the last eight years. He received his Ph.D. in electrical engineering from the Polytechnic Institute of New York and his MSEE and BSEE from New York University. He is the author of *Digital Signal Processing with C and the TMS320C30* (1992), co-authored *Digital Signal Processing with the TMS320C25* (1990), and has published a number of articles associated with the TMS320 processors. He is a member of the IEEE and ASEE.

Bill Bitler is employed by InfiMed, in New York, as an systems engineer working with imaging products and digital image processors for medical diagnostic imaging. Prior to that he was with Philips Components in Rhode Island as an applications engineer working with imaging products to broadcast medical and industrial imaging markets. He received his BS in computer engineering in 1992 from Roger Williams University. Bill has authored and co-authored technical papers for the SPIE conferences on medical imaging, and co-authored several papers in digital signal processing.