POTENTIAL MEDICAL APPLICATIONS OF TAE

J. Ben Fahy, Robert Kaucic, and Yongmin Kim
Department of Electrical Engineering
University of Washington
Seattle, Washington 98195

POTENTIAL MEDICAL APPLICATIONS OF TAE

J. Ben Fahy, Robert Kaucic, and Yongmin Kim

Department of Electrical Engineering
University of Washington
Seattle, Washington 98195

ABSTRACT

In cooperation with scientists in the University of Washington Medical School, we have constructed a microcomputer-based image processing system for quantitative microscopy, called DMD1, for "Digital Microdensitometer #1." In order to make DMD1 transportable to different hosts and image processors, we have been investigating the possibility of rewriting the lower level portions of DMD1 software using TAE libraries and subsystems. If successful, we hope to produce a newer version of DMD1, called DMD2, running on an IBM PC/AT under the SCO XENIX System V operating system, using any of seven target image processors available in our laboratory. Following this implementation, we will transfer copies of the system to other laboratories with biomedical imaging applications. By integrating those applications into DMD2, we hope to eventually expand our system into a low-cost general purpose biomedical imaging workstation. This workstation will be useful not only as a self-contained instrument for clinical or research applications, but also as part of a large scale Digital Imaging Network and Picture Archiving and Communication System, (DIN/PACS). Widespread application of these TAE-based image processing and analysis systems should facilitate software exchange and scientific cooperation not only within the medical community, but between the medical and remote sensing communities as well.

INTRODUCTION

Quantitative microscopy is an important tool for researchers and clinicians in various medical disciplines. It is composed of two quite different methodologies: morphometry, in which spatial properties are measured, and densitometry/fluorometry, which measures mass or activity. The principles which underlie specific techniques of either kind are well understood, and analog instruments ranging in sophistication from conventional microscopes fitted with photomultiplier tubes (PMT) to scanning microdensitometers and flow microfluorometers have emerged. Unfortunately, these instruments tend to be specialized (inflexible), are expensive, and are slow and difficult to interface to computers. As a promising alternative, a digital technique has recently arisen based on interfacing a camera directly to the microscope, and using image processing operations to analyze the resultant digitized images.

In Dec. 1982, we began a joint effort with researchers in the Department of Pathology to develop our own image processing system, and apply it to the measurement of DNA content in hypertensive smooth muscle cells. By Dec. 1984, the prototype system, which we refer to as DMD1 (Digital MicroDensitometer #1) was completed, and it has been heavily utilized in running experiments and analyzing images since that time. New algorithms and features have been continuously incorporated into the software [Nicholls et al., 1985; Vinter et al., 1985; Vinter et al., 1986] with the result that new applications in automated grain counting, immunocytochemistry, and other disciplines can now be developed as natural extensions to our existing system. This expandability is a key advantage of digital methods in quantitative microscopy, since the same basic method can be applied to any number of different applications. Other advantages of DMD1 include improvements in speed (200 cellular DNA measurements per hour vs. approximately 40-50 per hour for conventional analog methods), accuracy, and the possibility of conducting simultaneous analysis of morphometry and densitometry. Comparison of analog and digital systems so far has indicated that digital methods for densitometry/fluorometry are at least as good as the best analog devices [Vinter et al., 1985].

We are now at the stage with our system that it is appropriate to consider making DMD1 (and its successor, DMD2) widely available to other medical researchers by moving the software to a more accessible combination of host and image processor. In its current implementation, DMD1 resides on a custom Motorola MC68010-based host microcomputer and uses a CAT 1600 image processing subsystem (Digital Graphics, Palo Alto, CA). We are in the process of transporting the entire image processing and analysis system to an IBM PC/AT with an ITI (Imaging Technologies Inc., Woburn, MA) FG-100-AT image processor board. Drivers for the image processor under the SCO XENIX System V operating system have been written, and a majority of the DMD1 software has been successfully ported. Because our original system runs UNIX System V and the software was written in a very layered fashion which isolated device-dependent portions of the code, other than writing the new drivers and emulating the CAT image processing functions, transporting the software package has been relatively straightforward.

Now that we have copies of DMD1 on at least two hardware configurations, we face a problem associated with maintaining software compatibility between the two implementations. As new applications are developed on each system, they should be ported over to the other as rapidly as possible. Every time an application takes advantage of the device-dependent features of one system, it will have to be emulated on the other. If a third system is added, then its hardware features will have to be emulated on the other two, and it will in turn have to emulate their special features. Rapidly, as the number of different image processors increases, this emulation strategy is likely to become overwhelmed by the sheer number of combinations which would have to be managed. Therefore, we are seeking to develop alternative transportation strategies now, which will lessen the difficulty in maintaining many different implementations of DMD1 in the future.

The issue of system transportability is being felt not only within our own research group, but is beginning to be recognized within the general medical

image processing community as well. There are many different kinds of image processing systems being used in various clinical and research laboratories. In applications including fluorescent microscopy, microdensitometry, neurological morphometry, autoradiography, and others, hosts range primarily among VAXes, LSIs, PDPs, Eclipses, and PC compatibles [Smith et al., 1985, Ramm et al., 1984, Puls et al., 1986]. The image processors commonly used are manufactured by vendors including Gould, IIS, Grinnel, MATROX, ITI, Datacube, and others . In some cases, such as in our laboratory, special-purpose image processors are designed and built from scratch to meet a particular application need. Other installations may have important peripherals available such as array or signal processors. Because nearly all of these systems have been developed separately, on widely varying combinations of host and image processor, very little constructive sharing has taken place between these research groups. Effectively, this means that each of these efforts has reduplicated the others, leaving little opportunity for the development and implementation of more sophisticated tools.

In what follows we consider the application of TAE to this transportability problem in biomedical image processing, and how we propose to use TAE to make our own system, DMD2, more widely available. We are especially interested in the development of a Biomedical Virtual Image Processor (BVIP), along the lines of the DMS subsystem of TAE. We will discuss briefly the path we intend to take toward incorporating the TAE structure within our system, its potential implementation on a range of seven image processors of widely-varying architecture and capability, and the ramifications of these modifications to the practical possibility of creating a general purpose biomedical imaging workstation. Finally, we will consider the use of such workstations in a large-scale Digital Imaging Network and Picture Archiving & Communication System (DIN/PACS). If we are successful in propagating a TAE-based DMD2 to this extent, it will open up new opportunities for direct and effective software exchange and cooperation.

METHODS

DMD2 is nearly complete, i.e., porting DMD1 to the IBM AT host and ITI image processor. The next step is to implement DMD2 on up to six other image processors in our lab, each of which will be discussed briefly below. For this task, we will replace portions of our system with the TAE Display Management Subsystem (DMS) and refine DMD2 and DMS as necessary to allow the same applications program to run on any of seven different image processors. Following the completion of this task, we will integrate the remainder of DMD2 into the TAE monitor structure, taking advantage of TAE's built-in help, tutor, and other facilities. As it becomes necessary, we will then be in a position to port DMD2 to other operating systems on which TAE is supported, and all new programs written for DMD2 can be created from the beginning within the TAE programming context.

In Figs. 1 and 2 are presented simplified representations of the DMD1 and TAE software architectures, respectively. We will compare the two systems in a top to bottom fashion, noting both the obvious differences and some important

similarities as well. To begin with, in TAE, the top level software module is the TAE monitor, which initiates processes or TCL command language procedures, provides access to the help and menu and tutor facilities, and in many ways can mimic the performance of a generic operating system. There is no equivalent to this module in DMD1. Instead, for most users, a pre-defined sequence of programs is executed, which together perform a full densitometry operation (from digitization through decalibration to cell identification and analysis). This is for the benefit of the medical researchers and technicians, many of whom are unfamiliar with computers to the extent that any deviation from a fixed and rigid pattern of interaction is considered undesirable. For system and application programmers, individual routines may be invoked directly using UNIX. Thus far, this interface has proved adequate, but as the general level of computer expertise within the biomedical community improves, and as DMD2 itself expands to the point where the programmers themselves will require some sort of a help facility, we will need to turn to some sort of a monitor such as is provided by TAE. In fact, as will be discussed below, we eventually plan to implement a version of DMD2 which runs as an application under TAE.

Below the TAE monitor, and as the top layer in our system, reside the applications programs. In DMD2 most of these routines are dedicated to the higher-level functions required for quantitative microscopy. This includes programs for:

- system initialization and calibration
- image digitization and image display (B/W, pseudo color)
- image management (image handling and cataloging)
- interactive device management (tree-structured menu generation)
- histogram generation (whole image or specified region)
- image contrast enhancement (lookup table manipulation)
- algebraic operations (+,-,x,/)
- geometric operations (translation, rotation, roam and zoom)
- 2-D convolution for spatial filtering (with region of interest)
- 2-D Fast Fourier Transform
- various edge enhancement and boundary detection algorithms
- user manipulable cursors for region of interest analysis
- image intensity profile along any specified line
- thresholding for object segmentation and selection
- decalibration to correct uneven illumination in the microscope, and the camera's nonuniform photometric sensitivity
- densitometry operations for quantitative measurements.
- automatic boundary detection algorithms
- automatic morphometry with densitometry
- complexity analysis

The total programming effort for our system in this regard thus far is approximately 4 man years. Each of these routines may be executed as stand-alone procedures or as part of a larger densitometry program chain. Moved into TAE, this chain could be implemented very easily, simply by using a TCL command procedure to invoke the individual applications programs one after the other.

Below the applications layer in TAE is DMS, which provides a device-independent library of standardized image processing functions. It is composed

of an X-layer which is written independent of the particular hardware device being used, and a D-layer, which is device-dependent. In the position corresponding to DMS in DMD2 is the Firmware Extension Layer, created to expand the capabilities of CAT firmware commands. It is important to appreciate that in both systems, these layers (DMS and the Firmware Extension Layer) act as the only interface between the applications programs and the lower-level routines below. This common feature of the two designs is what will make replacement of the Firmware Extension Layer with DMS straightforward.

Below DMS in TAE is the vendor interface layer, which corresponds to the Firmware Interface Layer in DMD2. Each of these layers is designed to contain software functions completely specific to the particular image processor supported. At the lowest level of either software structure are the device drivers, referred to as the Physical Interface Layer in DMD2. As has previously been mentioned, these have already been written for the IBM AT running SCO XENIX System V. Separate drivers are used to perform memory-mapped I/O, manage I/O channels, and service interrupts. In most image processing systems, frame buffers are directly memory-mapped, image processor registers are mapped either through memory or through I/O channels, and interactive devices such as mouses, trackballs, or bit-pads are best implemented using interrupt service routines.

In order to provide a comprehensive and thorough testing ground for the DMS implementation in our system, we intend to create up to seven versions of the device-dependent D-layer, one for each image processor which is available in our lab. These image processors range widely in capability: differences include the number of bits per pixel in the frame buffer (gray-scale resolution), width of the look-up tables, and prescence or absence of graphics overlays, hardware cursor support, display zoom, pan, and scroll, coprocessors, and dedicated image operations hardware, e.g., histogram generation, image arithmetic, and image filtering.

The Digital Graphics Systems CAT-1600 graphics board features real-time digitizing, zoom, pan and scroll, and dedicated graphics and image processing commands implemented in an Intel 8086-based subsystem. Our implementation uses one 512 x 512 x 8-bit bit frame buffer, which is directly memory-mapped in the host address space over an IEEE 696 (S-100) bus. Communication with the host is facilitated by 4 16-bit I/O ports in the IEEE 696 bus: a data port, command port, reset port, and status port. Three independent look-up tables of 8 bits each are assigned to red, green, and blue, enabling pseudocolor options. There is no dedicated hardware support for the cursor, however, a cursor is emulated in the firmware package by overwriting the frame buffer to display the cursor, and restoring the data when the cursor is moved.

The ITI FG-100-AT image processing board resides on the IBM AT bus and contains 16 I/O channel-mapped registers to initiate commands and receive status information. The 512 x 512 x 12-bit frame buffer is directly memory-mapped into the host address space. Three 4096 x 8-bit look-up tables are provided for pseudo color or even true color operation. Like the CAT, real-time digitization, zoom, pan and scroll are supported in hardware. There is no image processor. A feedback loop arithmetic unit is provided, however, whereby 6-bit images may be added, subtracted, multiplied or divided in one frame display time. In our standard operations, we use 8 bits of each pixel for gray level, 3 bits for

graphics, and 1 bit for the cursor. There is no hardware cursor support, which again must be emulated by writing into the frame buffer's cursor plane.

We have had some success already in moving portions of DMD1 to an IBM Professional Graphics Adaptor (PGA) in an IBM AT environment. This raster graphics board consists of a 640 x 480 x 8-bit frame buffer, three 256 x 4-bit look-up tables, and an Intel 8086 based coprocessor which supports three-dimensional graphics and several international standard graphics packages such as the Graphical Kernel System (GKS). Communication between the host and the PGA is accomplished via memory-mapped command, data and error buffers and additional registers. There is no support for hardware zoom, pan, or scroll, and the cursor must be emulated. This system is our lowest capability system.

On the other side of the spectrum is a Gould IP8400 image processing system, which supports many image analysis/processing features in hardware, resulting in high performance. The Gould IP8400 is equipped with three 512 x 512 x 8-bit frame buffers, a video output controller, a pipelined high-throughput digital video processor, and a library of image processing software from Gould. Its host computer is a MicroVAX II, which is networked through DECNET to other computers. In our initial implementation of DMD2, we will run the UNIX operating system, although later versions in which DMD2 runs as an application under TAE could be implemented under VMS.

The TISDB board is a Texas Instruments Software Development Board based on the TMS 34010 Graphics System Processor (GSP) chip. The TISDB has one 512 x 512 x 4-bit frame buffer. It has a look-up table scheme based on their palette chip which allows resetting of the three 16 x 4 bit look-up tables line by line. While this board does not have the gray-scale resolution to be useful for most image processing applications, it has proved a useful tool for gaining familiarity with the powerful GSP chip, and can similarly provide a useful test of DMS. The GSP is a fully programmable 32-bit graphics processor, with special hardware features such as a 256-byte instruction cache and block data move facility, which make it very effective for some image processing operations [Guttag et al., 1986]. Using a C compiler and loader, we have implemented a number of demonstration programs for the GSP, including zooming, convolution, look-up table manipulations, cursor, and menu support. We have also written a resident monitor for the GSP, which loops until given a command from the host to execute a local program. Upon completion, all programs return to the monitor. This interface is implemented both in DOS and XENIX.

We are in the process now of developing a sixth image processor, also based on the GSP, but with much greater capability and gray-scale resolution. This image processor, which we will refer to as UWGSP1, consists of two boards which reside completely within the IBM AT, containing five major sections; the graphics processor, frame buffer, video display, zoom hardware, and signal processor [Chauvin et al., 1986]. It contains four 512 x 512 x 8-bit frame buffers, plus four separate graphics overlay planes, one of which is allocated to the cursor. There are three 4096 x 8-bit look-up tables for red, green, and blue output. Independent vertical and horizontal zoom is supported in hardware, as are pan and scroll. A separate signal processing coprocessor is provided based on the TMS 32020 Digital Signal Processor (DSP) chip.

The last image processor to be used is also in the development stage. It is called UWIP1, and is designed around a special-purpose high speed (30 MBytes/sec) image bus called the IBUS. It features an expandable central frame buffer which currently contains 4 512 x 512 x 12-bit frames. The display processor is based on the Hitachi HD63484 Advanced CRT Controller (ACRTC), and provides independent x and y zoom, three 4096 x 8-bit look up tables for red, green, and blue, hardware support for pan, scroll, cursor movement, and various graphics and annotation functions. Other key modules on the IBUS include a pipelined reconfigurable convolver, arithmetic and logic unit, look-up table transformer and histogram generator, and host interface buffer. Region-of-interest (ROI) operations are implemented in hardware.

To implement DMD2 on such a wide range of image processors, we will begin by writing the DMS D-layer for each processor, using the D-layer provided for the IIS Model 75 as an example. The other portions of DMS, for example the XD, DM, DT, and XL subroutine packages, should port over in a fairly straightforward manner. Because our medical applications do not at this time involve multispectral image analysis, we will delay supporting the image configuration utility, until a specific need for multispectral classification in radiology using images from multiple imaging modalities arises. With DMS available for each several image processors, we will rewrite the Firmware Extension Layer of DMD2 so that the functions called by that layer are implemented using calls to DMS. This will provide the quickest port of DMD2 possible consistent with the TAE architecture, as the Application Layer of DMD2, which was built directly on its Firmware Extension Layer, should be immediately transportable from that point on.

Most DD routines will be fairly routine to implement. Of those routines expected to provide difficulty, most are related either to FORTRAN77 conversions or else to peculiarities of the IIS image processor which made their way into DD. The DDCRDF and DDZMRN routines exemplify special cases in which different image processors may have difficulty in implementing different functions. For example, DDCRDF is for the most part straightforward, except that it allows the possibility of a cursor BLINK attribute. While this could be supported fairly easily on either UWGSP1 or UWIP1, supporting this feature on other processors (without any dedicated cursor hardware) is probably more trouble than it is worth. In the case of the DDZMRN function, the problem is that the function is ambiguous for image processors which support independent x and y zoom, or arbitrary-size zoom, such as UWGSP1 or the TISDB.

It is important to note that DD in its current implementation leaves out many hardware features of our in-house image processors which can be quite important to the efficiency of the running system. For example, the Device Characteristics Mask of the DMS Display Device Table contains only 8 hardware characteristics which are checked for existence thus far: hardware zoom, histogram generator, split screen, image shift, scale on input, look-up table bypass, alpha generator, and keypad buttons. Clearly, these features have been singled out with a specific image processor in mind. But almost certainly in our implementation, in order to achieve maximum efficiency, we will also have to include characteristic flags for:

- independent X and Y zoom (ITI, UWIP1, UWGSP1, TISDB)
- hardware convolver (UWIP1, UWGSP1)
- array or signal processor (UWGSP1)
- arithmetic unit (ITI, UWIP1, UWGSP1, TISDB)

These capabilities will have to be incorporated into the system model of image processing characteristics in order to fully take advantage of the hardware available. In turn, new routines should be added to DD and XD to allow these functions to be callable from applications programs. Another important consideration which DMS currently seems to leave unresolved is the nature of communication with the image processor. Does it reside on a separate bus? Are frame buffers directly memory-mapped, which greatly simplifies image loading, or is a more complicated interface required? Should interrupt routines be used as an integral part of the communication strategy, or is a polling or master/slave communication model sufficient?

With the wide range of image processors available in our laboratory to experiment on, we expect that we will experience a great many difficulties in adapting a consistent DMS interface to each image processor. On the other hand, it is our hope that this experience will prove very profitable in that solutions to these problems inevitably will be worked out, and a more sophisticated model of image processor capability than is presently outlined in DMS may result. It should be noted that as image processing hardware continues to improve over time, the model will continually have to be updated to correspond to the new state of the art: the creation of a "virtual image processor," then, which is the fundamental goal of DMS, will indefinitely remain a very dynamic process. It is our feeling that the virtual processor model will be kept best up to date by continually subjecting the model to new and widely-varying capability image processors, such as we will be attempting to do in the case of our widespread implementation of DMD2.

DISCUSSION

The previous section has outlined the method by which the DMS subsystem of TAE may be used to aid in the portability of our digital microdensitometer to a wide variety of image processors. The immediate beneficial effects of such an exercise are (1) to achieve the port itself, and (2) to improve the virtual image processor model of DMS to include a more comprehensive list of image processor functions. In addition to these immediate benefits, however, there are other longer-term advantages which are also important to consider.

Either after or in parallel with the implementation of DMS in DMD2, we will turn to the inclusion of the DMD2 application programs within the general framework of TAE. This would most likely be attempted first for the IBM AT/ITI-based system. Besides creating TAE format help files for the programs, some restructuring might prove necessary because in DMD2, many function options are currently selected via the cursor on a display-based menu, as opposed to on a separate terminal. It should be noted, however, that it is likely that such restructuring will only have to be carried out once, since after menus are moved to a separate terminal, the rest of the TAE application program interaction

should be dependent only on the IBM AT host, and not on the particular image processor chosen.

With DMD2 available as an application package under TAE, it should become possible to port DMD2 to other operating systems on which TAE is supported, e.g., the VMS operating system available on our MicroVAX II. We will proceed to integrate DMD2 with other biomedical image processing systems and applications so as to fill out its overall functionality, approaching the creation of a general purpose biomedical imaging workstation. Since each installation will be based on the TAE structure and libraries, it should be much easier to share software between users of this system, as well as with the investigators of completely different fields of endeavor in the NASA remote sensing community.

As a final example of the utility of this approach, we consider the application of the resultant workstation to a large scale Digital Imaging Network and Picture Archiving and Communications System (DIN/PACS) [Alzner et al., 1986; Parrish et al., 1986]. The DIN/PACS is being be designed to aid radiologists in interpreting images and associated data for medical diagnosis, and will provide for:

- image capture by the system
- storage of images
- image retrieval for diagnostic and display purposes
- image manipulation, arrangement and enhancement during diagnostic viewing
- entry and retrieval of clinical data
- diagnostic report generation
- indexed retrieval and statistical analysis for research purposes
- medical education activities
- radiology department management

The fundamental advantage of DIN/PACS over the present image management approach used in a typical radiology department is that it will allow physicians and radiologists to combine and analyze simultaneously all available information on a patient, especially including internal images produced by such imaging modalities as Computer Aided Tomography (CAT), Magnetic Resonance Imaging (MRI), Digital Subtraction Angiography (DSA), Positron Emission Tomography (PET), Ultrasound, and digitized X-Ray film. Each modality presents a different view of the internal state of the body, and by combining the information obtained from all modalities, it may be possible to significantly increase the effectiveness of noninvasive radiological diagnosis.

Because of the wide ranges of image processing capability which is expected to be used in DIN/PACS, it would be advantageous if the software used to develop and manage DIN/PACS image processors contained an effective virtual image processor model such as DMS may eventually provide. Further, it would be highly desirable to have some means of facilitating communication, exchange of software and ideas, and general cooperation between the propagators of DIN/PACS and the general image processing community. Therefore, if we are successful in developing the biomedical imaging workstation in a timely fashion, we will attempt to incorporate it into DIN/PACS, providing a TAE link between medical

imaging and the remote sensing communities.

SUMMARY

Biomedical image processing applications have been developed relatively recently compared to applications in remote sensing. As a consequence, many biomedical laboratories and clinical installations are just now beginning to meet head-on the problem of software portability which TAE was designed to combat. In our own laboratory, we have developed a microcomputer-based image processing system for quantitative microscopy, which we would like to port to other hosts and image processors in order to expand its capabilities to that of a biomedical imaging workstation. Because of the wide range of image processors available to us, we have decided to replace the image processor interface portion of our system with the DMS subsystem, thereby forcing the development of a Biomedical Virtual Image Processor which should greatly aid portability within the biomedical community. As part of this process, DMS will be severely tested, refined and enhanced. As the workstation develops, it will be possible to incorporate it within a large scale DIN/PACS. By using TAE as an integral part of these medical image processing systems, general software exchange and scientific cooperation will be facilitated between the medical and remote sensing image processing communities.

REFERENCES

Alzner, E., and Murphy, L., "The ACR-NEMA digital imaging and communications standard: evolution, overview and implementation considerations, Proceedings of SPIE, Vol. 626, pp. 506-510, 1986

Chauvin, J. W., Steiner, A. R., and Kim, Y., "Image processing with the TMS34010 Graphics System Processor," Digital Design, submitted, 1986.

Guttag, K., Van Aken, J., and Asai, M., "Requirements for a VLSI Graphics Processor," IEEE Comput. Graphics Applica., Vol. 6, pp. 32-47, Jan. 1986.

Nicholls, W. H., Kim, Y., Schwartz, S. M. and Vinter D. W., "Development of a microcomputer-based microdensitometer," International Journal of Microcomputer Applications, accepted, 1986.

Parish, D. M., Creasy, J. L., Thompson, B. G., Rogers, D. C., Johnston, E. R., Perry, J. R., Staab, E. V., "Functional requirements for interfacing PACS to RIS," Proceedings of SPIE, Vol. 626, pp. 597-602, 1986

Puls, J. H., Bibbo, M., Dytch, H. E., Bartels, P. H. and Wied, G. L., "MicroTICAS: The design of an inexpensive video-based microphotometer/computer system for DNA ploidy studies," Anal. Quant. Cytology Histology, Vol. 8, pp. 1-7, 1986.

Ramm, P., Kulick, J. H., Stryker, M. P. and Frost, B. J., "Video and scanning microdensitometer-based imaging systems in autoradiographic densitometry," J. Neurosci. Methods, Vol. 11, pp. 89-100, 1984.

Smith, L. C., Jericevic, Z., Benson, D. M., and Bryan, J., "Digital imaging fluorescence microscopy," IEEE Frontiers Eng. & Comp. Health Care, Vol. 7, pp. 967-970, 1985.

Vinter, D. W., Nicholls, W. H., Kim, Y. and Schwartz, S. M., "Development of a biomedical image processing computer system," IEEE Frontiers Eng. & Comp. Health Care, Vol. 7, pp. 996-1000, 1985.

Vinter, D. W., Kim, Y. and Schwartz, S. M., "Assessment of digital microdensitometry for histochemical applications: comparison with fluorescent methods for DNA determination of vessel wall cells," Lab. Invest., submitted, 1986.

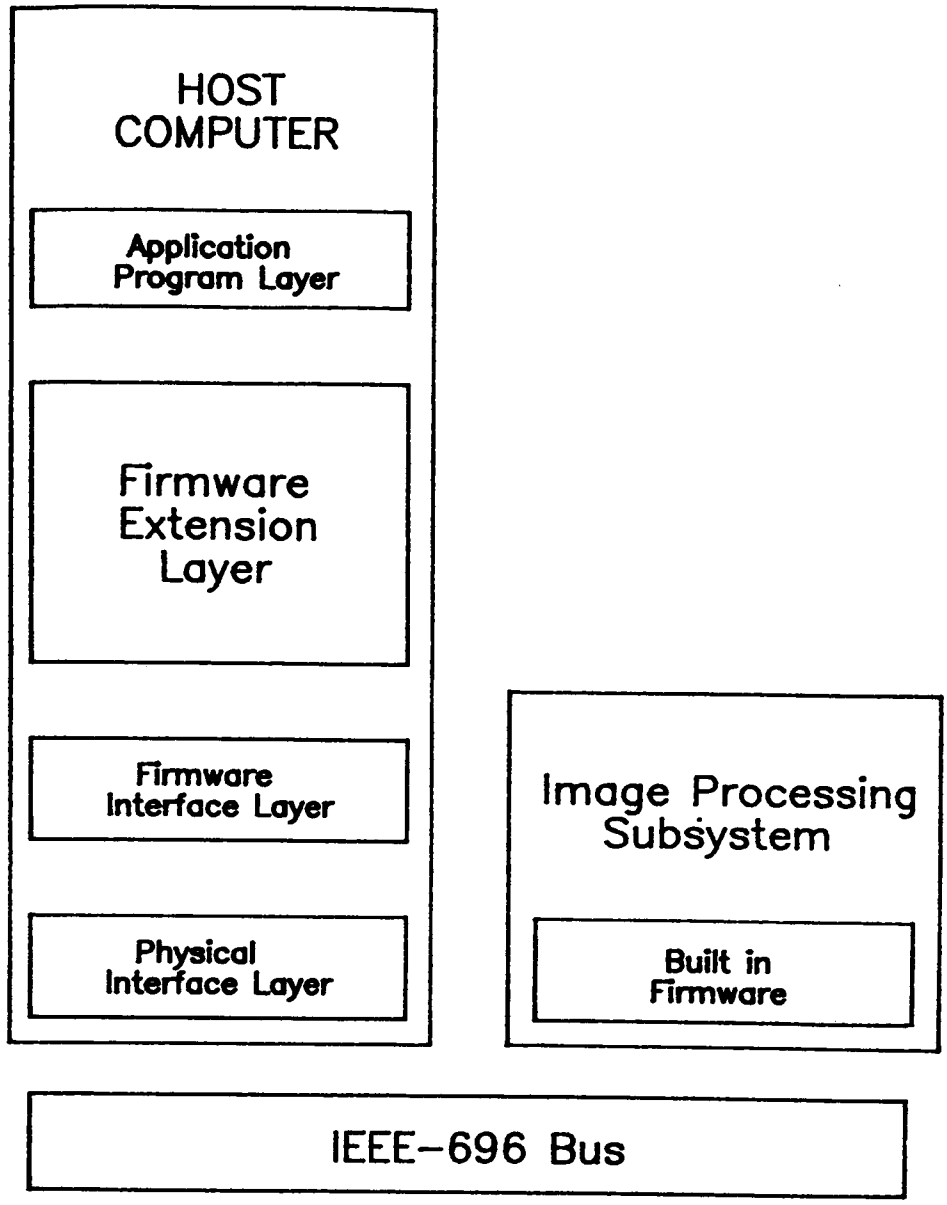

Figure 1. The layered interface software design in DMD1.

C-3

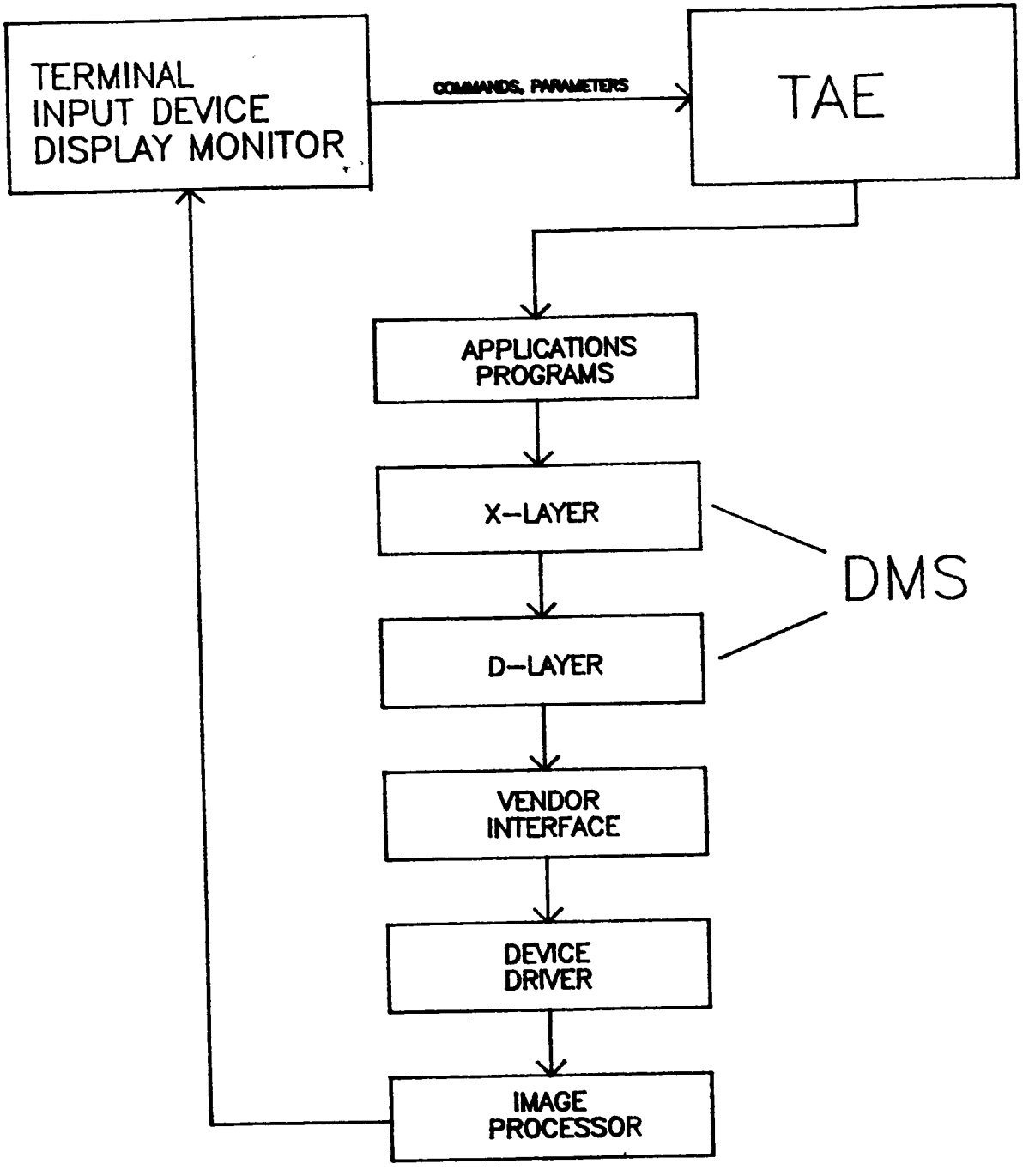

Figure 2. Simplified block diagram of TAE and DMS.