Report Number: WUCS-85-01

1985-01-01

# The Enhanced WUDMA Image Processing System

Andrew Laine and Seymour P. Pollack

This document describes recent enhancements to the WUDMA image processing laboratory that implement improvements suggested by experience gained during the WUDMA I program. More recent improvements to the software are mentioned as well as a discussion of some of the design issues faced during the early stages of conception. Finally, some ongoing projects and future plans are mentioned.

# The Enhanced WUDMA Image Processing System

Andrew Laine
Seymour Pollack

WUCS-85-01

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

# The Enhanced WUDMA†

# Image Processing System††

*Andrew Laine*
*(afl@wucs.UUCP)*
*(ihnp4!wucs!wudma!afl)*
*Seymour V. Pollack*
*(svp@wucs.UUCP)*
*(ihnp4!wucs!wudma!svp)*

Washington University
Department of Computer Science
St. Louis, MO. 63130

Technical Report No. WUCS 85-1

*ABSTRACT*

This document describes recent enhancements to the WUDMA image processing laboratory that implement improvements suggested by experience gained during the WUDMA I program. More recent improvements to the software are mentioned as well as a discussion of some of the design issues faced during the early stages of conception. Finally, some ongoing projects and future plans are mentioned.

## 1. Introduction

This report describes the preparation of an integrated image processing software system for use in the professional development program in computer science for image processing conducted by Washington University for the Defense Mapping Agency. The need to provide such an integrated system can best be understood by describing the software available for WUDMA I, the initial professional development offering, and mirroring the state of this software against the requirements and demands of the curriculum.*

No commercial software packages for image processing were suited for the needs and demands of the planned curriculum. Most of these packages were either specialized towards a particular set of applications or offered a limited collection of general image processing applications. The curriculum demanded a more powerful set of algorithms and a more "open ended" approach to application development. A survey of the various packages under development at

† Washington University-Defense Mapping Agency professional development program in computer science for image processing.

†† Supported by Defense Mapping Agency Aeronautical Center Grant DMA 700-84-C0037

* Curriculum objectives and requirements are described in detail in Washington University's proposal to the Defense Mapping Agency entitled "A Professional Development Program in Computer Science for Image Processing" dated August 15, 1983.

Universities throughout the nation indicated that no single constellation then available could adequately address all the areas of interest demanded by the WUDMA program. Although each software system offered useful algorithms that complemented those provided by other systems, the software components could not be merged because each system was based on its own image formats, data types, and data structures. Compatability among systems was virtually nonexistent. Consequently, exploitation of the various software capabilities, at least for the first WUDMA offering, required the participants to deal with these diverse characteristics.

Thus, since compatibility was not (and could not be) a major factor at this initial stage, software was selected based on its applicability and state of development. Six image processing software collections from various sources were identified as being desirable for support of the WUDMA curriculum. As it turned out, the originating organizations were (and continue to be) most cooperative, and we were fortunate in being able to acquire and install all of the packages. The names of the original sources and their respective organizations are listed below:

| System Name | Institution | Contact |
|---|---|---|
| PDS (Picture Display System) | Cornell University | Anthony Reeves |
| V Shell (Visual System) | Univ. N. Carolina | John Zimmerman |
| Vision System | Carnegie-Mellon | Dave McGowen |
| Pattern Recognition System | Cornell University | Anthony Reeves |
| Display System | PAR Corp. | Patricia Lentz and Robert Grey |
| DeAnza/UNIX driver | Purdue Univ./PAR Corp. | George Gobel |
| WUDMA System | Washington University | Andrew Laine and Steve Reichenbach |

The PDS system provided most of the fundamental algorithms needed in the early phase of the laboratory work (Project 1) for WUDMA I. Other systems used later in the WUDMA I semester included the V Shell system and the Display System. These systems were functionally good from the viewpoint of the algorithms they contained, but taken together were inconvenient to use due to file format conversion and data type reconciliations required for a mixed sequence of operations and display of image results. To get these additional algorithms into the WUDMA I laboratory as soon as possible, we developed conversion utilities for image format and display compatibility between systems. The students of WUDMA I used these utilities to prepare images before applying operations and then before those images were displayed.

While the WUDMA I students gained experience in having to deal with the compatibility problems between large software systems, we knew that we wanted to add even more software from other systems and that the then available solution of providing conversion utilities would eventually lead to a cumbersome user interface. Also, we wanted to provide an environment in which students could more easily experiment with applications. The students of WUDMA I not only wanted to run programs, but wanted to modify them to make them more effective for a particular application or image.

**2. Objectives**

Based on this assessment, the faculty concluded that the professional development course would benefit substantially if the problem of compatibility between the selected systems could be solved more generally via the development of a "user-friendly" application environment . Accordingly, we proposed (and DMAAC approved) a project to develop an integrated image processing environment that would:

1. Make image format compatibility differences betweens all subsystems "invisible" to the user. We felt the user should not have to deal with the different header types and data types needed to implement the various filters and image processing utilities within a particular system.

2. Provide a common user interface to all the subsystems. Such an interface would be "user-friendly" in that all commands would be executed from the user's top level shell and all commands would follow the same syntax for specifying command options and function modifiers. We hoped to achieve this "system independence" through the UNIX C-Shell command interpreter.

3 A User contributed software interface. We wanted to provide a framework that would encourage users (DMA students) to write programs of their own or modify an existing program to fit a particular image or application.

4. The WUDMA system should also offer display device independence as much as possible, using software to adapt to differences in hardware between the DeAnza display and the Vectrix display architectures.

5. We were also concerned with maintainability of the overall system. We wanted to provide a framework for the maintenance of individual programs within a particular subsystem as well as maintainability of the various subsystems couched in the overall WUDMA system. Some of the subsystems arrived poorly documented with respect to maintenance and did not use the *Makefile* utility available under UNIX* for maintaining large software systems. We sought to build a Makefile system that would embrace the entire system from the ground level up. This was also an ambitious task, but one that we felt strongly about since personnel changes and graduations in the University environment are frequent. Immediate benefits would include having the capability of rebuilding an entire subsystem via a single command, and encouragement to fix reported bugs immediately since file dependency information would be expressed in the **Makefile.**

6. Provide high-quality documentation. It was important that the students be able to "pick up" the software quickly. The uniformity of command syntax would also help in this area.

7. Develop advanced I/O routines for the DeAnza IP8500 via trackball and pushbutton controls. The addition of a trackball/pushbutton device would allow more interactive software for image registration, digitization and graphics.

8. Utilize the array processor (DVP) within the DeAnza IP8500 to study the performance of various image processing applications using parallel algorithms. Two additional memory boards for the DeAnza IP8500 would be purchased to allow implementation and execution of the parallel algorithms.

## 3. Major Design Issues

Initial studies identified the following alternative approaches, each of which was considered to be ultimately feasible:

1. Convert all systems to the one that offered the best collection of utilities and the one we thought offered the better framework for future program development. This would mean a long term commitment to one system and one image format. This would also mean that future applications originating from other systems would have to be converted to the chosen format. Having been more familiar with the PDS system, since it was the first system we used in the course, (the vsh system was not made available until late in the WUDMA I semester.) it was an attractive candidate. However, further study of the vsh system during early summer made it an attractive candidate as well. In PDS format, image information was stored directly in a header structure. This differed from the the vsh system were image information was made available to applications through subroutine calls only. In short, the PDS code was simpler to modify and more easily read.

2. Rewrite all the applications to interface with a new image format specifically designed for the WUDMA image processing laboratory hardware. This would probably result in faster execution times for most applications but would require lots more programming and a longer

* UNIX is a trademark of Bell Laboratories

lead time for program verification and debugging. Here, we felt that designing yet another format would buy us nothing new and would lessen the potential for an exchange of software and bug fixes between the various institutions.

3. Couch the various subsystem applications within C-Shell programs that would adjust input and output image format by calling the appropriate conversion utilities when needed. Image files could be tagged for format identification by a "magic number". An image file would be left in the format of the last application. This alternative seemed the most attractive. Here, we would need to write fast file format conversion programs and data format conversion utilities to do the "dirty work" for compatibility between the various subsystems. With this approach, we would remain compatible with the original code and format of each subsystem. Bug fixes and new programs could easily be exchanged between institutions and would encourage the development of an image processing community. The Shell command interpreter available under Unix seemed well suited for the task of unifying all the subsystems under a common command syntax. The C Shell programming language was selected (rather than the Bourne Shell) becaused it offered a more robust set of file path specification operators and had various other attractive programming features desirable for this project.

## 4. Implementation

The following section describes some of the issues faced during the implementation of system integration, the DeAnza IP8500 software enhancements, and the development of a user contributed software environment.

### 4.1. System Integration

An initial solution called for converting the vsh routines to operate on 8-bit images to make them compatible with PDS format. Since none of our display systems supported a 16-bit frame buffer but did support 8-bit frame buffers, this made sense. Also, images could be stored in half the space. Most of the vsh programs were converted to handle 8-bit data over a two week period. However, through comparison of 8-bit and 16-bit processing results we became aware of qualitative differences in the results of the more computation intensive vsh programs. In particular, adaptive histogram equalization, unsharp masking filter, Hough curve detection, Marr-Hildreth edge detector and most frequency domain-based transforms and filters showed noticeable differences in quantitative results when compared with their 16-bit counterparts. This result, along with the fact that it was more tedious to convert complex vsh based algorithms to 8-bit compatibility then previously believed, made us reconsider our approach.

The final solution took into consideration the long term goal of remaining compatible with the original formats of each subsystem. This was important because we wanted to be able to exchange new software and offer bug fixes that could be installed with minimum effort and error. Maintaining compatibility with the "mother" subsystem would "leave the door open" and encourage an exchange of programs/bug fixes among institutions.

We were able to accomplish these objectives through clever programming of the C-Shell. Development of the shell scripts were more complex but the payoff in long term compatibility and maintainability will be well worth the extra effort spent early on.

### 4.2. Gould/DeAnza IP-8500 Software Enhancements

The DeAnza as presently configured is a single standard resolution (512x512) RGB or black and white workstation. The WUDMA II class was the first to enjoy some additional utilities made possible through the purchase of additional hardware.

#### 4.2.1. Programming the DVP

The DeAnza IP8500 configuration attached to the WUDMA VAX 11/750 machine includes an array processor option called the "DVP", digital video processor. This option allows an entire frame buffer (512x512x8) image to be processed through one of its pipelines in less than 1/30th of a second. There are four such pipelines within the DVP, each 8 bits wide. Computational structures within the DVP include multipliers, two stages of ALU's, comparators, shifters and lookup tables. This device can provide real-time image processing. We were very excited about bringing this capability into the WUDMA program. Some immediate benefits are described below:

(1) DVP image processing would take some of the cycle load off the VAX cpu. Most of the image processing programs in the WUDMA system are compute bound VAX based programs.

(2) This enhancement would encourage experimentation with image processing techniques. Statistics gathered from the WUDMA I class showed that the VAX11/750 turnaround time for a typical 2-D convolution operation on a 512x512x8 image was about 12 to 20 minutes (depending on load factors ranging from 1.0 to 8.0). The same class of operations take only a few seconds using algorithms that exploit the parallel architecture of the DVP.

Early algorithm analysis revealed that the exploitation of such parallelism on this architecture required at least two additional frame buffers. These "scratch pad" memories are needed to hold the overflow/underflow information possibly generated by the ALU's during algebraic and convolution operations as well as hold window masks and intermediate pixel results generated during local pixel-neighbor based processing, typically used in many pattern recognition, statistical and classification algorithms.

The purchase of two additional frame buffers allowed us to utilize this piece of hardware for the first time during WUDMA II. Current image processing algorithms running on the DVP include Laplacian filters (four of them), local median, local max, local min, edge detection, edge enhancement, pixel averaging over various sized windows, a standard deviation weighted window operation, contour tracing and contrast enhancement utilities.

These programs motivate a new environment in which to conduct image processing research and production. We are very excited about having the opportunity to shape that environment. This opportunity is truly attractive in that there are no "image processing standards" or industry standards that might constrain a potentially great design. This is truly a luxury from an engineering viewpoint, and we are excited about being part of the academic community committed to image processing science that will lead the way to intelligent designs of systems in the near future.

We plan to expand our library of DVP based algorithms to include a broader range of image processing applications. A recent agreement between Washington University and Gould/DeAnza may provide an opportunity to utilize the DVP in pattern recognition problems and classification algorithms couched in an AI environment.

#### 4.2.2. Trackball/Pushbutton Peripherals

The addition of the trackball I/O device to the DeAnza configuration enables the use of some significant I/O utilities previously unavailable. The functional capabilities of the trackball device not only include a locator, but a set of pushbuttons (six of them) mounted on the trackball box. The combination of these two I/O devices made possible interactive zooming and roaming on images and motivated other I/O utilities. An area as large as 1024x1024 could be roamed and magnification factors from 1:1 to 1:8 were selectable via pushbutton control. Trackball sensitivity can be selected interactively to speed up roaming over large images. This utility was used extensively in WUDMA II, Project 2, for registration of large multi-spectral imagery photographed at different times of the year.

Added programs included a utility to select several different cursor types and pick off points with a pushbutton with the option of writing the set of points to a file. This utility was used heavily in digitizing map features for the cartographic database built in Project 4.

The *bnds* program can draw boundaries in user selected colors and bit planes, under trackball control. Since the last point drawn will remain fixed it is easy to draw straight lines between points. This can be a difficult task using other types of I/O devices such as a light pen. The fine control sensitivity of the trackball makes digitization of fine details a less tedious task. Another benefit of this program is that it allows the user to see the results of the boundaries while he is digitizing, rather than blindly picking off points that he thinks describe a feature.

### 4.3. User Contributed Software Environment

Another goal of this system was to provide a framework that would encourage students to write programs of their own design or modify an existing program to fit a particular image or application. To accomplish this, we developed prototype programs for each subsystem (PDS, vsh, Display) and made them available in a public directory. The prototype programs included many comments and were non-trivial but not overwhelming in complexity so that students could become familiar with each subsystem. This approach was needed in WUDMA I, and has proved very successful in WUDMA II, as seen by more then a dozen user contributed, high quality programs written by DMA students during the course of WUDMA II. Unix manual page prototype files were made available to encourage documentation soon after development. Based on student feedback, having these prototype programs really helped in getting the programs off the ground.

Prototype programs included several small "C" language programs, vsh prototype, PDS prototype (Fortran also available), Display prototype, DVP prototype, DeAnza Graphics, Vectrix graphics (Pascal and C) and EQUEL (embedded C), a QUEL query preprocessor for multi-database query development.

## 5. Results and Summary

We were able to accomplish our objectives completely as planned and on time for the benefit of the WUDMA II students.

A common user interface has been provided that models closely the standard Unix command specification syntax.

A User contributed software environment has been established to facilitate student program development under existing subsystems.

The entire system has been placed under *Makefile* control to provide software maintenance of both programs within a subsystem, and the system overall.

Documentation has been organized into a single manual, complete with index and cross references for subject material*.

The DeAnza/Gould IP-8500 software has been improved with the addition of advanced I/O programs that utilize trackball and pushbutton devices.

Programs for the Digital Video Processor within the DeAnza IP-8500 are now available for the study of parallel algorithms for image processing applications.

The WUDMA system, motivated by the demands of the curriculum, is a custom software package tailored specifically towards the needs of the WUDMA program. The tailoring is still going on, and will continue to reflect the changes in the WUDMA program as it evolves.

## 6. Present and Future Projects

The experience of the past two WUDMA classes have helped us tremendously in shaping the program. We are more aware of the problems facing DMAAC and continue to tune our laboratory towards application examples that motivate students toward solutions for some of these problems. The above mentioned manual is continually being improved based on student feedback. A pseudo user "imanual" receives all complaints/comments about the software and documentation through the mail system so that students can immediately report any frustrations experienced with the software as they encounter them. The second edition of the *WUDMA Image Processing Software Manual* is currently being prepared. It incorporates some excellent suggestions made by the user community that will benefit subsequent WUDMA classes.

* This manual entitled "Image Processing Software" is given to each WUDMA student as part of his or her text material. Additional copies have been supplied to pertinent DMA personnel.

The pattern recognition system available under PDS needed to be reviewed and brought into the program. Published references were gathered to help us understand the many utilities and determine if they would be applicable to DMA problems. A subset of these utilities will be used for the first time in WUDMA III and we are excited about application results possible with these programs.

The **Phoenix** system from Carnegie-Mellon University looks excellent. Most of the utilities under this system are hardwired to a Grinnel image display system. A port of the Phoenix system to the DeAnza and Vectrix devices will start up during the WUDMA III semester and be available for the WUDMA IV class.

A current project is concerned with long term maintainability. The entire WUDMA system is being placed under *RCS* (Revision Control System). Under this system we will be able to keep track of bug fixes and program modifications by revision number, so that previous versions can be referenced. If the WUDMA system is distributed to other sites, this would allow us to control updates by revision number.

The **TAE** (Transportable Applications Executive) under UNIX is being studied as a possible environment in which to couch WUDMA System applications. The **TAE** System has a strong user community under VMS. NASA at Goddard Space Flight Center is currently porting all of the VMS software to UNIX. The port is about 18 months from completion. We have received Version 1.2 of TAE/UNIX as a test site for evaluation of the executive environment. A long term benefit of using **TAE** would be compatibility with the Department of Earth and Planetary Sciences at Washington University, presently running TAE/VMS.

Future software efforts include building a LISP-based framework for artificial intelligence systems that attempt to solve some of the pattern recognition problems and classification problems facing DMAAC today. The use of a real-time disk attached to the DeAnza would motivate the design of a new type of image database system and allow real-time image processing of large multi-spectral imagery. The possibility of obtaining one of these hi-speed disks on loan is currently under negotiation between Gould/DeAnza and Washington University. A Unix port of the VMS based Gould/DeAnza Library of Image Processing Software (LIPS) and LIPS+ software will be designed to fit into the WUDMA system. The addition of these two packages will add even more depth to the WUDMA system.

The future is exciting, certainly challenging, and very bright. The program continues to improve, building on its past experiences. The environment provided by the program at Washington University seems to be extremely successful in bringing out the best of each student's ability and provides him or her with the tools needed to challenge the state of the art. We have taught and we have learned. The momentum built during the past two WUDMA programs has the potential to drive us onward and past some of the goals mentioned in this paper.