CVIPtools: A Software Package for Computer Imaging Education

MARK ZUKE, SCOTT E. UMBAUGH

Computer Vision and Image Processing Laboratory, Southern Illinois University at Edwardsville, Edwardsville, IL 62026

Received 28 February 1997; accepted 18 March 1997

ABSTRACT: New computer imaging software is described. CVIPtools illustrates principles of computer imaging and enables rapid development of complex algorithms. The easy-to-use graphical user interface makes available over 170 functions to students and artists, astronomers, geographers, and others who want to explore imaging but lack programming knowledge. © 1997 John Wiley & Sons, Inc. Comput Appl Eng Educ **5:** 213–220, 1997

Keywords: computer imaging; computer vision; image processing; image-processing software

INTRODUCTION

Computer Vision and Image Processing tools (CVIPtools) [1] is a software package used in teaching computer imaging to undergraduate and graduate students at Southern Illinois University at Edwardsville (SIUE). Developed over the past 5 years at SIUE, CVIPtools is used both as an adjunct to traditional laboratory exercises and as a research tool that allows students to contribute their work to the ongoing project. Soon to be distributed free over the World Wide Web, CVIPtools presents more than 170 functions and algorithms organized into five main categories, in a windowed program that is easy for even a nonprogramming user to operate. Fast algorithms and the automatic display of processed images provide immediate feedback to the user, thus reinforcing classroom lectures and tutorials provided with lab exercises.

BACKGROUND

CVIPtools was conceived by one of the authors (S.E.U.) after his doctoral work at the University of Missouri–Rolla, where he developed an image-processing color segmentation method for use in the automatic evaluation of skin tumors in humans [2]. At that time, no single generic program existed that made available a large number of commonly used functions and algorithms to researchers in the fields of image processing and computer vision. Existing programs were typically hardware specific; commercial packages offered an extremely small number of tools useful for researchers. The author found that a majority of his research time was spent in

Correspondence to Scott E. Umbaugh (sumbaugh@ee.sine.edu).

Contract grant sponsor: NIH SBIR; contract grant number: 1-R43-CA60294-01.

Contract grant sponsor: FUR SIU; contract grant numbers: F-EN310, F-EN411.

Contract grant sponsors: System Dynamics International; Camber Corporation; Stealth Technologies.

 CCC 1061-3773/97/030213-08

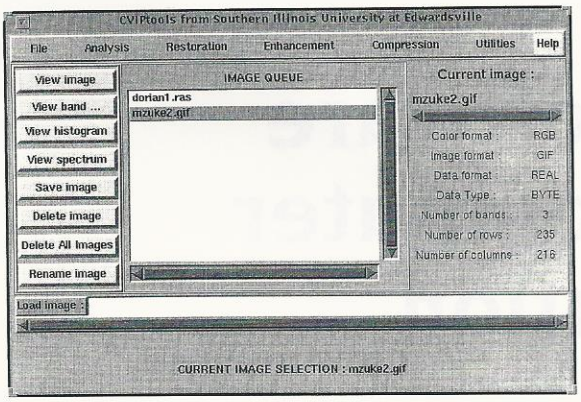

Figure 1 The main window for CVIPtools. Data for the selected (highlighted) image are displayed on the right side of the window.

writing computer code to perform algorithms and basic functions. CVIPtools was designed to provide a large number of basic and advanced operations to the nonprogramming user. Originally written as a textual menu-driven program, it was converted to its current windowed format in 1996 [3].

While CVIPtools has been designed and developed by the research team at the Computer Vision and Image Processing Laboratory at SIUE, funded by various grants, some elements of the program have been developed in fulfillment of students' master's projects [3–7]. CVIPtools contains one window (Feature Extraction) developed cooperatively by the students in a one-semester graduate course. The program is continually upgraded as new functions and algorithms are implemented in C-language routines. Work is under way to port the program from its native UNIX form to other platforms.

INTERFACE

The graphical user interface (GUI) for CVIPtools allows even those not familiar with computer imaging to experiment with sophisticated techniques. The GUI is easy to use because it operates in accordance with accepted principles of interface design [8,9]. Four important principles that underlie interface design are as follows: Consider the user's mental model of the task, maintain compatibility with other programs, maintain internal consistency, and use color sparingly. All users, regardless of their degree of knowledge of computer programming practices, have expectations as to how the program accomplishes its tasks. If the program operates in a fashion similar to that of the user's model, fewer errors will result. Because many users of CVIPtools are familiar with at least one or two other windowed programs, CVIPtools windows were designed with a menubar at top, messages near the bottom, and the functional area in the center. As the work flows from window to window within the program, standardized placement and use of buttons, command words, and methods of operating the functions maintain the required internal consistency. Finally, the use of color has been kept to a minimum in the program to avoid forcing the user to remember associations between a color and its meaning within the program, and to avoid the inconsistent display of colors that can occur on different computer systems.

The main, or root, screen for the program is shown in Figure 1. Operations that are common to all windows, such as loading, viewing, and renaming images, are located in the root window. The IMAGE QUEUE area lists images loaded by the user and those that result from operation of the program. Selection of an image on the list displays its basic image data and makes the selected image available to all functions for processing.

Choosing one of the major functional categories listed across the top of the window, Analysis, Restoration, Enhancement, Compression, or Utilities, causes that category's selection bar window to appear. All five major category selection bars can be

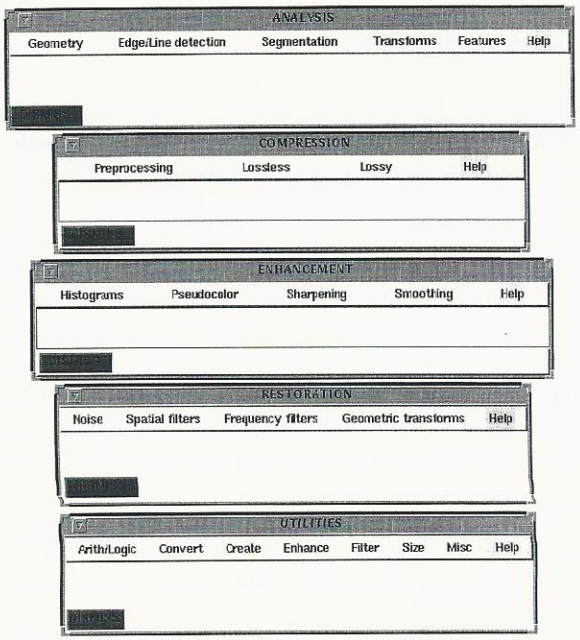

Figure 2 The selection bars for the five main function categories show the subcategories available to the user.

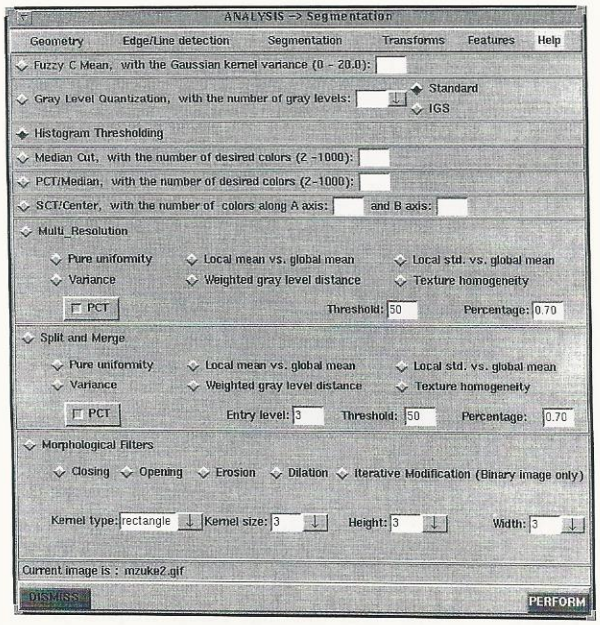

Figure 3 The segmentation window in CVIPtools. No parameters are required for histogram thresholding, the selected segmentation method.

onscreen at one time (Fig. 2), simplifying the locating of a needed function.

Choosing a subcategory from a selection bar causes that window to expand and display the available functions. Figure 3 shows the available segmentation functions within the Analysis window.

A help screen is available within the program, organized into the same categories available from the root window. The help pages list the required and optional parameters for functions in the selected subcategory and detail methods that can be used to enter the values. Additional help is available outside the program through the use of the UNIX *man* command, which can display a complete manual page of each CVIPtools function. During the program's operation, if the user's action will be irreversible, such as in deleting all the images at one time, confirmation to proceed is required with a pop-up alert message. An alert box is also used to notify the user of improper value selections or other program errors. Figure 4 shows sample alert boxes.

USE IN LABORATORY EXERCISES

The primary use of CVIPtools is as an adjunct to computer imaging courses within the electrical engineering department at SIUE. Laboratory exercises, tutorials, and a sample menu-driven C-language program are provided students to guide them toward an understanding of the methods and tools used in computer imaging. Classroom lectures introduce topics that are illustrated and reinforced in the lab exercises. The students add their own C-code routines to a sample program, then check the accuracy of their work by comparing the output of their program against that of the same function in CVIPtools. Tutorials make extensive use of CVIPtools, allowing the students to quickly view the results of making incremental changes to function parameters.

The nonprogramming laboratory exercises used in the imaging courses are of the form of the following example. The input and resultant images follow the text.

Lab Exercise

This exercise illustrates the effects of low-pass filtering on a real image and compares the effects of ideal filtering to those of Butterworth filters.

(a) Load a 256 × 256 complex gray-scale image (Fig. 5) and perform the FFT on it. Use a blocksize of 256. Filter the transformed image using an ideal low-pass filter with a cutoff of 32. Keep the DC value. The program will automatically perform the inverse trans-

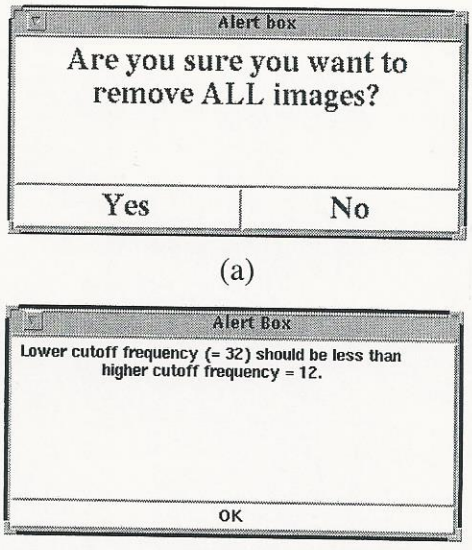

Figure 4 (a) Confirmation is required before deleting all images at one time. (b) Parameter error message specifies the exact problem.

Figure 5 Original image.

form and display the resultant image (Fig. 6). Notice the absence of high-frequency information and the ringing effect, which appears as waves emanating from edges in the image, caused by the sharp frequency cutoff.

(b) Now apply a Butterworth low-pass filter of order 1 to the Fourier-transformed image. Use the same cutoff as in (a). Keep the DC during filtering. Compare the result (Fig. 7) with that of (a). Is the ringing effect as noticeable now? Why/why not?

(c) Repeat (b) using an eighth-order Butterworth filter instead of a first-order filter. Compare the result (Fig. 8) to the results in (a) and (b). Because the frequency response of an eighth-order Butterworth filter is close

Figure 6 Ideal filtering; cutoff frequency, 32.

Figure 7 Butterworth filtering; cutoff frequency, 32; order, 1.

Figure 8 Butterworth filtering; cutoff frequency, 32; order, 8.

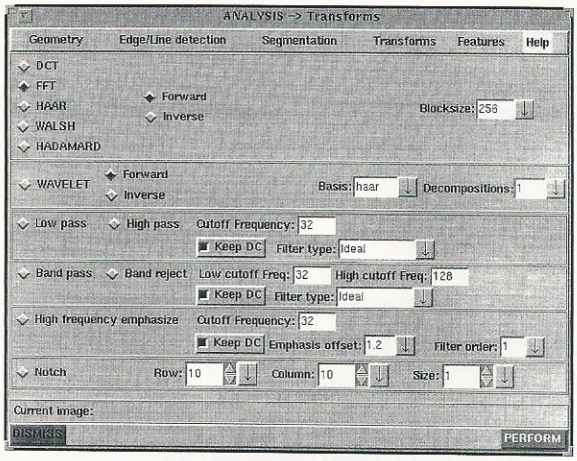

Figure 9 The transforms window in CVIPtools. Both transforms and transform filtering are available within the same window.

to that of an ideal filter, the image should be similar to that of (a).

Students use the Transforms subcategory of the Analysis window to perform the FFT (Fast-Fourier Transform) and the filtering called for in the exercise. The Transforms window is shown in Figure 9.

The ease of use of the GUI contributes to the success of the lab tutorials and exercises. For example, changing from the ideal low-pass to the Butterworth filter in the Transforms window is accomplished by clicking on the downward-pointing arrow to the right of "Filter type:" and selecting "Butterworth" from a pop-up list. Because changes are so easy to make and results are quickly displayed on the computer monitor, students often experiment more than called for in the exercises. Such experimentation was one of the desired results of the conversion of the program from its early text-based menu format to the current graphical interface form.

The CVIPtools project has attracted many students to the image processing and computer vision courses at SIUE. Beyond providing the usual satisfaction found in acquiring knowledge of a new field, the courses and CVIPtools offer students the opportunity to contribute their work to the project and see that work added to the software. The students in one graduate computer vision course cooperated in the creation of the feature extraction facility (Fig. 10) within CVIPtools. Research by some of the students concerning which features to extract was passed to others for implementation in C-language routines. Two students designed and constructed the window, and implemented the code needed to select a desired object within an image by using a mouse. A help page was written for the functions, and the entire window was integrated into the CVIPtools program.

CVIPtools is beginning to see use outside the Engineering Department at SIUE. Students in geography and computer courses, along with people having an interest in astronomy, biology, and art, are discovering applications of CVIPtools to their fields of interest. Self-study is facilitated by the ease of operating the GUI, and results are immediately available to the user. At a few selected sites outside the university, CVIPtools is being used by working professionals to preprocess substandard images for use with specialized computer applications.

USE IN APPLICATION PROJECT DEVELOPMENT

Another area of use for CVIPtools is in the development of application-specific algorithms. For example, in digital image-processing courses, projects often center on image improvement methods based on restoration or enhancement techniques. Figure 11 shows the RESTORATION → Frequency filter

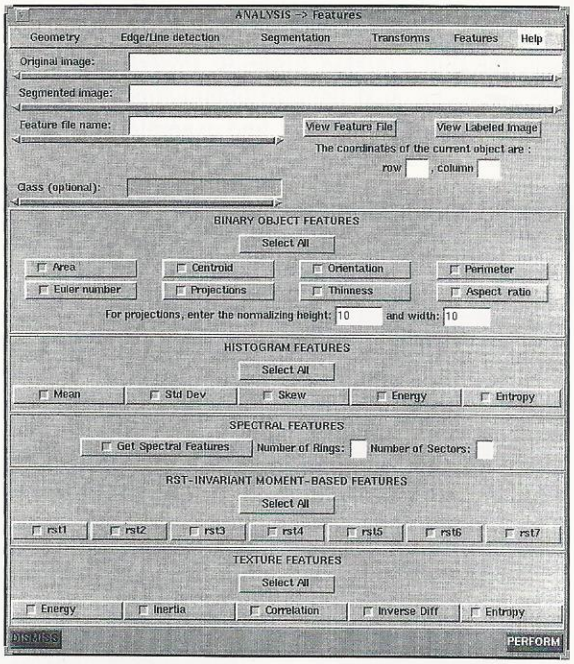

Figure 10 The feature extraction window in CVIPtools, designed and implemented by graduate students taking a one-semester course.

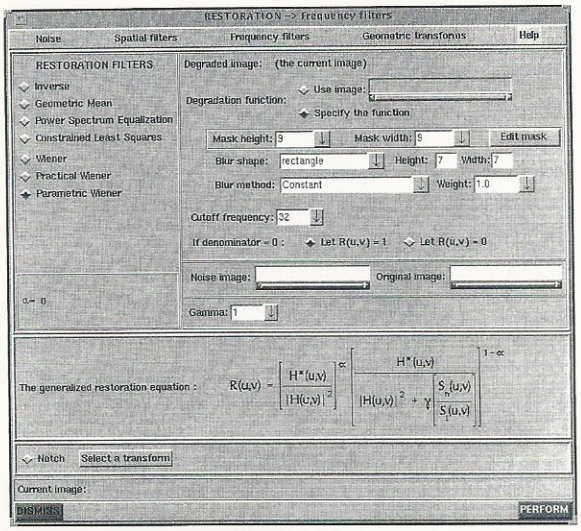

Figure 11 The frequency filter window in CVIPtools is one of four restoration screens.

window for CVIPtools; an Enhancement window, Histograms, is shown in Figure 12.

With the easy-to-use GUI-based program, students can develop algorithms experimentally. Various sequences of operations can be compared and contrasted quickly, allowing for extensive fine-tuning before any code is written. Once an algorithm has been developed, the student can rapidly create the application program because all functions within CVIPtools are accessible to C-language programmers through function libraries. Programmers can reference a manual page to obtain function prototypes, required parameters, and examples of a function's use. The example from the manual page can be copied directly and compiled to give the programmer an idea of how the function operates. For example, the manual page for the function *local_histeq* looks like this:

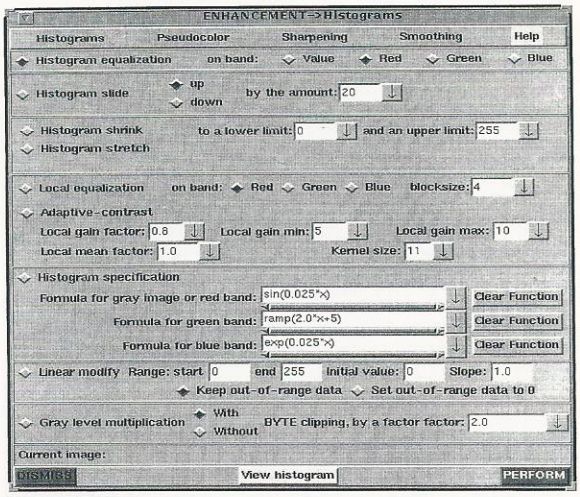

Figure 12 The histograms window in CVIPtools offers nine methods for image enhancement.

```
local_histeq(3)  C Library Functions
local_histeq(3)
NAME
  local_histeq—performs     histogram
equalization on block-by-block basis

SYNOPSIS
    #include ⟨stdio.h⟩
    #include ''CVIPimage.h''
    #include ''CVIPdef.h''
    #include ''CVIPcolor.h''
    #include ''CVIPhisto.h''

  Image *local_histeq(Image *in, int
  size, int mb)

    ⟨in⟩-pointer to an Image structure
    ⟨size⟩-desired blocksize
    ⟨mb⟩-RGB band on which to calculate
    histogram (0, 1, 2)

PATH
    $CVIPHOME/HISTOGRAM/
    local_histeq.c

DESCRIPTION
  This function performs histogram
  equalization on local areas of an im-
  age rather than on the image as a
  whole. The user may specify the size
  of blocks to use. By performing local
  rather than global histeq, detail can
  often be enhanced in large uniform
  areas of an image.

  The function will remap any INTEGER,
  FLOAT or DOUBLE image into SHORT. It
  operates directly on images of type
  SHORT and BYTE.

TYPES AND CONSTANTS
  None

RETURN VALUES
  A histogram-equalized image.

HISTORY
```

```
  History information recorded: None

EXAMPLE
    #include <CVIPtoolkit.h>
    #include <CVIPdef.h>
    #include <CVIPimage.h>
    #include <CVIPconvert.h>
    #include ''CVIPcolor.h''
    #include ''CVIPhisto.h''
    void main ()
    {
    Image *cvipImage;
    IMAGE_FORMAT format;
    char *inputfile, *outputfile;
    (void) setDisplay_Image
    (''picture'');
    print_CVIP(''\n\t\tEnter the In-
    put File Name: '');
    inputfile = (char *) getString_
    CVIP();
    format = getFormat_CVIP (inputfile);
    cvipImage = read_Image(inputfile,
    format, 1);
    view_Image(cvipImage, inputfile);

    cvipImage = (Image *) local_
    histeq(cvipImage, 16, 0);
    cvipImage = remap_Image(cvipIm-
    age, BYTE, 0, 255);

    print_CVIP(''\n\t\tEnter the Out-
    put File Name: '');
    outputfile = getString_CVIP();
    view_Image(cvipImage, outputfile);
    write_Image(cvipImage, outputfile,
    FALSE, FALSE, format, 1);
    free(inputfile);
    free(outputfile);
    }

BUGS
  None at this time

SEE ALSO
libhisto

AUTHOR
  Copyright (C) 1996 SIUE-by Scott Um-
  baugh and Arve Kjoelen.
```

Students refer to manual pages during creation of their final project in undergraduate and graduate imaging courses. Students are required to choose an area of interest, design experiments to pursue, define C functions to implement their experiments, and then code, debug, and test their algorithms. The sample menu-driven program provided to them, CVIPlab, is extended to include their own algorithms. Because CVIPtools' libraries can be linked to stand-alone programs, the amount of code written is minimal.

CONCLUSIONS

CVIPtools is a powerful collection of computer imaging tools, accessible through an easy-to-use graphical interface. The program allows students to see the results of performing both basic and advanced operations on digital images, and allows for easy experimentation that quickly answers the ''what-if'' questions that arise during the learning process. Because no specific knowledge of computer imaging is required for use of the program, people with interests as varied as cartography and graphic design are finding useful features within CVIPtools. The continual upgrading of the software will add to its usefulness for an ever-increasing range of users.

ACKNOWLEDGMENTS

The development of the CVIPtools software has been partially funded by Stoecker & Associates via NIH SBIR Grant 1-R43-CA60294-01; FUR grants from SIUE, Nos. F-EN310 and #F-EN411; and the U.S. Army funding via subcontracts through System Dynamics International, Camber Corporation, and Stealth Technologies. The authors also thank the numerous students who have contributed to the development, and the others on the development team: Greg Hance, Arve Kjoelen, Kun Luo, Yansheng Wei, Wenxing Li, and Zhen Li.

REFERENCES

[1] CVIPtools (a comprehensive computer vision and image processing software package), Southern Illinois University at Edwardsville.

[2] S. E. Umbaugh, ''Computer vision in medicine; Color metrics and image segmentation methods for skin cancer diagnosis,'' PhD dissertation, University of Missouri–Rolla, Rolla, MO, UMI Dissertation Service, 1990.

[3] M. A. Zuke, ''Design and implementation of a graphical user interface for CVIPtools,'' master's project report, Southern Illinois University at Edwardsville, 1997.
[4] G. A. Hance, ''Computer vision and image-processing software tool with application to color segmentation for skin cancer diagnosis,'' master's thesis, Southern Illinois University at Edwardsville, 1996.
[5] A. Kjoelen, ''Wavelet based compression of color skin tumor images,'' master's thesis, Southern Illinois University at Edwardsville, 1995.
[6] K. Luo, ''An image viewer for X window environment,'' master's project report, Southern Illinois University at Edwardsville, 1996.
[7] Z. Li, ''Development of CVIPtcl/CVIPwish for graphical user interface,'' master's project report, Southern Illinois University at Edwardsville, 1996.
[8] D. J. Mayhew, *Principles and Guidelines in Software User Interface Design,* Prentice Hall, Englewood Cliffs, NJ, 1992.
[9] A. Dix, et al., *Human-Computer Interaction,* Prentice-Hall International, Hertfordshire, UK, 1993.

BIOGRAPHIES

Mark Zuke received BS and MS degrees in electrical engineering from Southern Illinois University at Edwardsville in 1994 and 1997, respectively. As a research assistant to Dr. Umbaugh, he designed and implemented the first version of the graphical user interface for CVIPtools. In addition, he created and maintained the primary documentation for the software in the form of UNIX man pages. Since one of his interests is photography, he created many of the original photographs, as well as the final figures for Dr. Umbaugh's book, *Computer Vision and Image Processing: A Practical Approach Using CVIPtools.* He can be reached at the Electrical Engineering Department at SIUE, or at mzuke@ee.siue.edu.

Scott E. Umbaugh is currently an associate professor in electrical engineering at Southern Illinois University at Edwardsville, where he is in charge of the Computer Vision and Image Processing (CVIP) Laboratory. He received a BSE degree with honors from Southern Illinois University at Edwardsville in 1982 and MSEE and Ph.D. degrees in 1987 and 1990, respectively, from the University of Missouri–Rolla, where he was a Chancellor's Fellow. He worked in industry as a computer design engineer from 1981 through 1986, and he has worked as a computer imaging consultant since 1986. He currently serves on the editorial board for *IEEE Engineering in Medicine in Biology Magazine* and the *Pattern Recognition* journal. His professional interests include artificial intelligence, computer vision and image processing, medical applications of image processing, and engineering design education. He has authored numerous papers, coauthored two book chapters, and recently written a book to be published by Prentice Hall PTR, entitled *Computer Vision and Image Processing: A Practical Approach Using CVIPtools.* Dr. Umbaugh is also the primary developer of the CVIPtools software package. He can be reached at the Department of Electrical Engineering, Southern Illinois University at Edwardsville, Edwardsville, IL 62026-1801, or at sumbaug@ee.siue.edu.