

AC 2008-142: INTEGRATION IMAGE ANALYSIS PROJECTS IN AN INTRODUCTORY COMPUTATIONAL METHODS COURSE USING MATLAB SOFTWARE ENVIRONMENT

Abhijit Nagchaudhuri, University of Maryland Eastern Shore

Abhijit Nagchaudhuri is currently a Professor in the Department of Engineering and Aviation Sciences at University of Maryland Eastern Shore. Prior to joining UMES he worked in Turabo University in San Juan , PR as well as Duke University in Durham North Carolina as Assistant Professor and Research Assistant Professor, respectively. Dr. Nagchaudhuri is a member of ASME, SME and ASEE professional societies and is actively involved in teaching and research in the fields of engineering mechanics, robotics, systems and control and design of mechanical and mechatronic systems. Dr. Nagchaudhuri received his bachelors degree from Jadavpur University in Calcutta, India with a honors in Mechanical Engineering in 1983, thereafter, he worked in a multinational industry for 4 years before joining Tulane University as a graduate student in the fall of 1987. He received his M.S. degree from Tulane University in 1989 and Ph.D. degree from Duke University in 1992.

Integration of Image Analysis Projects in an Introductory Computational Methods Course Using MATLAB Software Environment

Abstract

“Computational Methods for Engineers”, is a 3 credit course offered to sophomore/junior students at University of Maryland Eastern Shore(UMES) to engineering majors. The course has been adapted from the “Introduction to Matlab” course that was being offered to engineering majors in the past. Some students in the Department of Mathematics and Computer Sciences have also taken the course. The course introduces students to basic programming concepts, vector methods, linear algebra, solutions of linear and nonlinear equations, two-dimensional and three-dimensional graphics, and numerical analysis, including numerical integration and numerical solutions of ordinary differential equations. The “Image Processing Toolbox” of Matlab is integrated in the project work for the course. It provides an avenue for introducing students to ongoing experiential learning and research endeavors in remote sensing that the author is involved in, with support from NASA and USDA. The image analysis projects not only provide a new and appealing dimension to the course, but also enhance comprehension of course material by active learning.

Students have worked in teams to perform a variety of image analysis tasks, including color segmentation in the RGB and HSI/HSV domains; computing normalized difference vegetation index (NDVI); mosaicking of overlapping digital frames; and other projects that have relevance to the ongoing research efforts in which the principal author is engaged in.

In this paper, an outline for the course will be provided and a few of the projects executed by the student teams in the last couple of years will be highlighted. Student feedback and assessment of the course, with particular reference to the project, will also be discussed.

1.0 Introduction

A significant number of textbooks have been recently published for use in introductory computational methods courses using Matlab software environment ^[1-4]. Also, textbooks are available for courses such as, Statics, Dynamics, Mechanics of Materials, Control Systems, Statistics, Image Processing, Basic Circuits, and Communication systems, ^[5-11] that are integral parts of general engineering curricula, as well as popular engineering

majors, including mechanical, civil, aerospace, electrical, etc. The books referred to above are just a few examples from a large and growing list of engineering text and reference books that are relevant to undergraduate and graduate engineering curricula, and bear testimony to the important role that the Matlab software environment plays in engineering education and research. Matlab is not a programming language like 'C/C++' or 'Fortran' which have been the languages of choice in the past for engineering educators to introduce students to scientific and engineering computation^[12,13]. Although the numerical recipes and other pre-written subroutines/functions in these languages can provide students a good introduction to computational methods, the growing use of Matlab software in engineering education, engineering research, and the workplace may be attributed to the ease of use, better student comprehension, less rigidity of syntax, and the availability of a variety of toolboxes that seamlessly integrate with upper level courses in undergraduate engineering curricula, as well as engineering research in graduate school and in the workplace. While Matlab is an excellent environment for providing instruction in computational methods, linear algebra, numerical analysis, and introductory programming concepts,^[14] it is not a programming language and has limitations when it comes to real-time applications. As such it is not a substitute for 'C/C++' for introduction to programming concepts, data structures, and real-time applications. References^[12,13] discuss the integration of 'C/C++' and 'Matlab' in an introductory computer programming course and the ongoing evolution, modification, and adaptation of the course framework.

In the next section of the paper, an outline for the "Computational Methods for Engineers" course at UMES based on the Matlab programming environment will be provided followed by a discussion of course highlights and project assignments. Learning objectives, outcomes, and assessment strategies relevant to Criteria 2000 of the Engineering Accreditation Commission will also be elaborated prior to the concluding section.

2.0 Course Outline- Computational Methods for Engineers

"Computational Methods for Engineers" is a 3-credit course in the baccalaureate engineering program at University of Maryland Eastern Shore (UMES). The course is offered to sophomores/juniors with a pre-requisite of "Calculus –II" and a co-requisite of "Differential Equation" courses. A working knowledge of basic programming concepts is also a desired pre-requisite. The course emphasizes fundamentals of vector methods, linear algebra, numerical analysis, programming, and 2-dimensional as well as 3-dimensional graphics. The first seven chapters of the textbook "An Engineer's Guide to MATLAB"^[1] are covered in the course lectures. The author also introduces students to fundamentals of image processing and the use of some of the basic functions in the "Image Processing Toolbox". The students are assigned a team project that is designed to reinforce analytical, computational, graphics, and programming concepts introduced in the course lectures. Utilization of the "Image Processing Toolbox" provides an appealing and practical dimension to the project. The project also emphasizes teamwork, report writing, and oral presentation.

The students work on practical problems relevant to engineering while learning the analytical and computational methods. For example, while solving systems of linear equations, examples are drawn from statics, dynamics, and electrical networks as the students learn basic linear algebra, matrices, Eigen-values, and inversion of matrices. Examples relevant to robot kinematics and homogeneous transformations are introduced, while introducing students to basic ideas of matrix multiplication. Solving nonlinear equations with the Newton-Raphson method and basic programming ideas, are introduced while developing a program to solve link positions and angles of a four-bar mechanism as the crank undergoes a complete revolution. Pre-written functions for performing numerical integration are introduced together with graphing tools to enhance visualization and perception of students. First order and second order differential equations that represent fluid/thermal systems, resistor-inductor-capacitor circuits, and mass-spring-damper systems, with appropriate practically relevant forcing functions and initial conditions are used as examples. Matlab graphic tools are utilized to introduce students to the Runge-Kutta method for obtaining numerical solutions of ordinary differential equations (ODE), using pre-written Matlab functions embedded in student developed m-files. While discussing numerical solutions of ODEs, basic ideas of vibrations and controls are also provided. Some examples involving nonlinearities are also discussed (bungee jumping, inverted pendulum) to elaborate the ease with which numerical solutions can be obtained when closed form solutions are arduous. After basic concepts from the first seven chapters of the textbook are covered, the students are also introduced to Image Processing Toolbox and basic statistics for laying out the foundation for the team project. At this stage, the students are also encouraged to leaf through the subsequent chapters of the text that span over vibrations, design, control systems, fluid mechanics, heat transfer, optimization, and statistics. The students are encouraged to utilize the text for subsequent courses in controls, design, statistics and robotics, fluid mechanics, heat transfer, digital signal processing, electro-magnetics, communication systems, and remote sensing, which some or all of them will take in their junior and senior years. The students are also encouraged to utilize Matlab for most of their computational needs subsequent to taking the course.

3.0 Project Work and Highlights

The principal author has been involved in experiential learning and research efforts in the fields of robotics, control systems, and more recently precision agriculture and remote sensing at University of Maryland Eastern Shore (UMES), with support from local industry, NASA, and USDA^[15-18]. Matlab and its toolboxes have been utilized by undergraduate and graduate students who have worked in these projects under the supervision of the author. The author has demonstrated the use of the dynamic data exchange capability of “Working Model” and “Matlab” to simulate control of mechanical systems^[17]. Matlab was utilized to develop a transformation from the overhead camera frame to robot base frame for the robot-vision system set-up in the University of Maryland Eastern Shore Mechatronics and Automation Laboratory (UMESMAL). Also the dual water tank, rotary pendulum and flexible rotary arm utilize Simulink and Realtime Workshop from Mathworks Inc. for real-time control applications in the UMESMAL. The visual representation of control algorithms in Simulink is translated to

Matlab m-files, which are subsequently translated to optimized 'C/C++' code and compiled using a Visual C++ compiler for real time control solutions ^[18].

Matlab "Image Processing Toolbox" has been utilized along with other software environments such as PCI-Geomatics, ArcGIS 9.2, ERDAS Imagine, etc., for image analysis related to remote sensing and precision agriculture efforts ongoing at UMES, with support from Maryland Space Grant Consortium, NASA, and USDA ^[15, 16]. The project assignment in the "Computational Methods for Engineers" course has been developed to introduce students to basic concepts in image processing, and to reinforce the basic analytical and computational concepts introduced in the course. Moreover, the relevance of the course project to ongoing experiential learning and research efforts provides students with an understanding of how the knowledge developed in the course may be utilized and expanded in a practical setting. The project work also helps attract undergraduate engineering student research assistants to work with the multidisciplinary team working in the "AIRSPACES : Aerial Imaging and Remote Sensing for Precision Agriculture and Environmental Stewardship" and "Environmentally Conscious Precision Agriculture(ECPA) -A Platform for Active Learning and Community Engagement", projects that are ongoing at UMES.

A few example projects that have been completed by student teams in the course are outlined below:

RGB Image Analysis

In one of the first runs for the course, the student teams were required to acquire a digital image of a flower bed and count pixels of different colors in the frame. Subsequently, they were provided remote images from a camera system mounted on a tethered blimp ^[19]. The project allowed students to understand that RGB images were basically three layers of 640 by 480 (typically, higher or lower resolution images are also possible) matrices. The first layer is formed by pixel intensities in the red band (R), the second layer is formed by pixel intensities in the green band (G) and the third layer is formed by pixel intensities in the blue band (B) of the electromagnetic spectrum. The students utilized the "imread" and "imshow" functions in the "Image Processing Toolbox" to read and display the RGB images in the Matlab environment. They also learned to display only the red, green, and blue layers as grey level images in Matlab. By using the 'impxel' and 'improfile' functions, the students readily realized that each pixel in the red, green, and blue layers contained numbers from 0-255, indicating the intensity level of the corresponding bands for an 8 bit image. Students investigated the images to understand the colors 'pink', 'yellow', and 'green' as observed in Figure 1(a), which represented a range of values in the red, green, and blue bands. Thus, to count the pixels of a certain color, a program had to be developed that will provide the ranges in each band, and increase the count as the images are traversed pixel by pixel using a three-level nested for loop so that every pixel value in the three corresponding layers is accounted for. Figure 1(a) show the original image of a flower bed captured on a digital camera displayed in the Matlab environment. Figure 1(b) shows a similar image in which the numbers of pink, yellow, and green pixels has been identified, counted, and recolored with uniform shades of pink, yellow, and green respectively, corresponding to the pink

flowers, yellow flowers, and, leaves and grass. All pixels not falling in the ranges for pink, yellow, and, green in Figure 1(a) were also counted in the same program and designated as 'other'. The results are subsequently displayed using a pie-chart in the Matlab environment in Figure 1(c). The same program was utilized to analyze remotely sensed images from a tethered blimp with a camera system, and the analysis results are shown in Figures 2(a), 2(b), and 2(c). Students also downloaded a freeware tool titled "MultiSpec" [20] developed at Purdue University and investigated image classification tools available in the software to compare the results obtained by them.



Figure 1(a) : Original Image



Figure 1(b) : Classified image

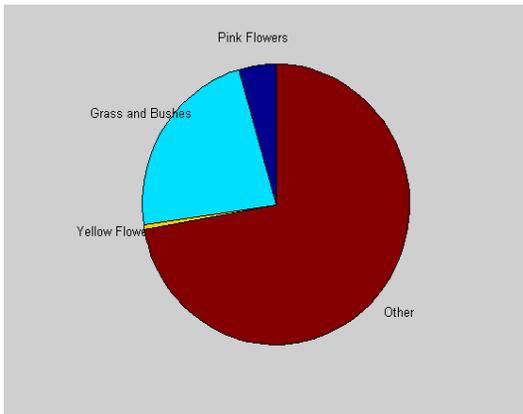


Figure 1 (c) : Pie Chart for Pixel Counts



Figure 2 (a): Original Image (Aerial)

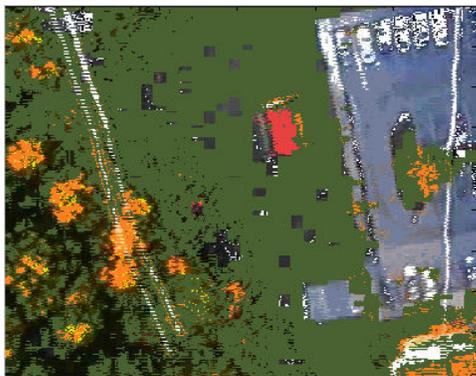


Figure 2(b): Classified Image (Aerial)

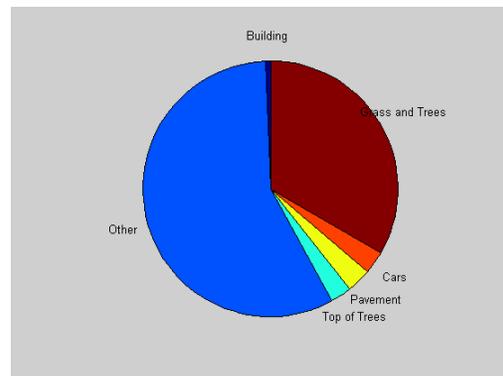


Figure 2(c): Pie Chart for Pixel Counts

The project reinforced linear algebra, programming, data visualization, and graphing concepts introduced in the class and helped integrate course material with applications in the real world. While some students indicated the project requirements made their workload very heavy in the course, most students felt the project helped them significantly and inspired them to learn on their own. The student feedback from the initial pilot offering of the course established that the project requirement added value to the course. In subsequent offerings, a variety of image analysis projects were integrated with the course. While most of the projects had relevance to ongoing federally-funded experiential learning and research activities pertaining to remote sensing at UMES led by the author, the scope was limited to the level that undergraduate students at the sophomore and junior levels, with limited exposure to programming and signal processing concepts, would be able to handle. A few more examples of project activities undertaken in subsequent offerings of the course are outlined below.

Mosaicking Multiple Images

When capturing remote images of a field from aerial platforms, it is seldom possible to capture an entire area of interest in a single frame; hence, several overlapping frames have to be digitally stitched or mosaicked to get a view of the entire area of interest. In remote-sensing applications, mosaicking is often performed together with rectification and georeferencing. Some of the early remote-sensing, mosaicking, rectification, and georeferencing efforts at UMES, conducted while executing objectives outlined in the AIRSPACES and ECPA projects, are reported in reference ^[21]. Graduate students involved with the effort use Matlab, along with ARCGIS, ERDAS Imagine, and PCI-Geomatics software environments, to perform most of the geo-informatics and image-processing tasks. The author learned about a mosaicking tutorial using Matlab, developed at Drexel University, while exploring basics of digital mosaicking techniques on the internet ^[22]. This provided a good project framework for integration with the course. The student teams were assigned to study in detail the script files(m-files) for image mosaicking provided in the website for stitching two frames and subsequently modify the code so that it can be used to stitch multiple frames together. Student teams experimented with several digital frames, including four overlapping sections of a flower bed, six overlapping frames of a mural that spans over an entire 40 ft by 20 ft wall in an UMES building, and remote images collected from aerial platforms with camera attachments. An example of a mosaick of digital frames, captured using an aerial platform for a portion of UMES campus developed using Matlab code adapted from the “Mosaicking Tutorial” provided in reference, ^[22] is shown in Figure 3.

This project not only enhanced the students ability to understand and use the Matlab software and the “Image Processing Toolbox”, it also helped students to relate the concepts to the remote sensing efforts integrated with the ongoing ECPA and AIRSPACES projects at UMES.

Normalized Difference Vegetation Index (NDVI)

NDVI (Normalized Difference Vegetation Index) ^[23] is utilized to determine the health of vegetation from multi-spectral aerial images. The Terra-hawk aerial imaging device ^[24]

acquired for the ECPA and AIRSPACES projects consists of a gimbaled ColorInfraRed(CIR) camera that can be retrofitted on the belly of an airplane. It automatically adjusts for roll, pitch, and yaw distortions. The camera system uses a user-defined shape-file and a GPS unit to automatically acquire images at an appropriate interval of time without the intervention of the pilot. The GPS unit determines whether the airplane has entered the shape-file region to trigger image capturing. The images can be captured in the RGB or CIR mode. In the CIR mode, the three bands captured are red, infrared, and blue/green. NDVI information can be easily determined from the first two bands using Equation 1:

$$\text{NDVI} = (\text{NIR} - \text{RED}) / (\text{NIR} + \text{RED}) \quad (1)$$



Figure 3: Multiple Image Mosaic Using Matlab Code.

To observe the effects of variable levels of nitrogen on 30-foot wide strips of wheat plantation on an agricultural field, an experiment was set up in the UMES campus as an integral part of the ECPA project. Images of the field were captured using both RGB and CIR modes from a height of 500ft using the Terrahawk system. As may be observed from Figures 4, 5(a), and 5(b), while the entire field could be imaged from a single RGB image, the CIR image capture rate allowed the whole wheat experiment field to be captured with two frames. Using the “Image Processing Toolbox”, the NDVIs were calculated for the individual frames and mosaicked together, following methods described in the previous section. Figures 6(a) and 6(b) show the mosaicked view of CIR frames and the corresponding NDVI frames. The information captured in the NDVI frames is critical in assessing the health of the wheat plant and can be correlated with measurements using hand-held GPS and field sensing devices.



Figure 4: Wheat Experiment Field (RGB)

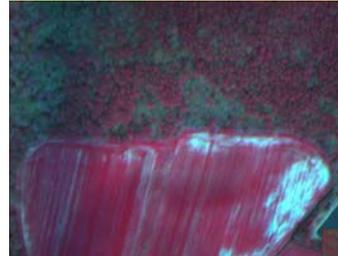


Figure 5a: CIR Image Of A Portion Of The Field

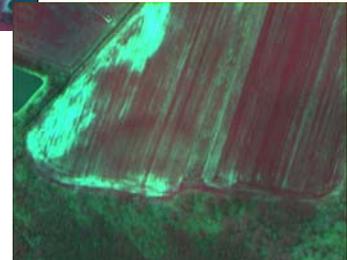


Figure 5b: CIR Image Of The Remaining Portion

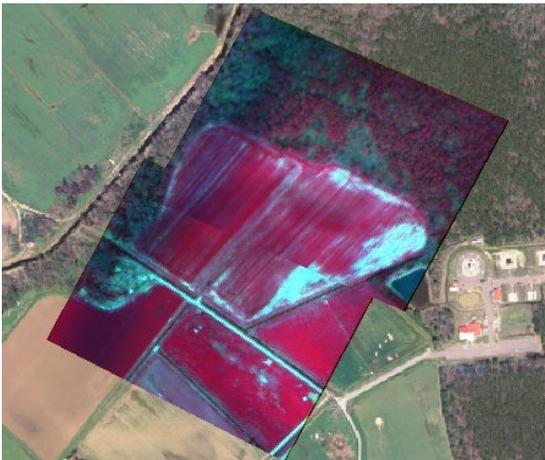


Figure 6(a) : Mosaicked View of CIR Frames



Figure 6(b) : Mosaick view of NDVI Frames

Once a CIR image is read into Matlab(with Image Processing Toolbox) using commands similar to:

```
c = imread('c:\pictures\cirimage.jpg');
```

The 'nir' and 'red' bands may be extracted using commands similar to:

```
nir = im2single(cirimage(:,:,1));
red = im2single(cirimage(:,:,2));
```

The dot division (element by element division) operation in Matlab can be efficiently utilized to obtain the NDVI by a single command:

```
ndvi = (nir-red)./(nir + red);
```

In any other programming environment, it would take a significant amount of coding to perform such computations, and exemplifies the power of Matlab for engineering and scientific computations.

Analysis Using Hue, Saturation, and Intensity (Value) – HSI/HSV Images

During the execution of the first example project “RGB Color Analysis,” the HSV (Hue, Saturation, and Value) ^[25] and RGB (Red, Green, Blue) color spaces ^[26] were discussed with the students; however, the pixel color explorations were performed in the RGB space. While manipulation of colors in the RGB space contributes to learning, as well as reinforcing computational knowledge using Matlab, it is not as efficient as using the HSV color space. Matlab functions `rgb2hsv ()` and `hsv2rgb ()` can transform images back and forth between RGB and HSV domains. In the RGB domain, a particular displayed color is a combination of ranges of pixel intensities in the first (red), second (green), and third (blue) layer matrices of the corresponding digital image. Hence, to search the pixels that are in a particular color range, a three-level nested for loop is necessary. However, once an image is transformed to the HSV space, all the color information is in the “hue” layer matrix. The pixel intensities in the “s” or “saturation” layer matrix indicate the richness level or vibrancy of the color; whereas, the pixel intensities in the “intensity” or “value” layer matrix indicate the brightness level. Once an image is transformed to the HSV domain, counting the pixels of a certain color can be performed by searching only the first (hue) matrix layer of pixel intensities. Moreover, by performing and displaying the hue ‘histogram’, it is possible to figure out the number of pixels corresponding to each color range. Also, if the hue level of a certain color range corresponding to a particular object is changed in the image without changing anything in the “saturation” and “value” levels, and the image is transformed back into RGB mode and displayed, only the color of the particular object will be changed without affecting any of the shading and brightness. Figure 7 corresponds to hue level manipulation of colors in a digital image of a group of “flamingo” birds and the corresponding “hue” histograms. The shift in hue histogram, as the flamingo birds are changed to a color corresponding to the hue value of 180 (purple) is clearly observable. Also, it may be observed that the change in color has taken place without changing the shades and brightness levels due to the “saturation” and “value” levels. Figure 8 is a similar example where the “hue” layer has been manipulated to replace the “blue” with a color corresponding to the hue value of 100 (green) color. Once again, the shift in histogram is clearly observable in Figure 8, and also it may be noticed the color change has not affected the brightness and shade since the “saturation” and “value” levels have remained intact.

The student feedback and anecdotal comments indicate this project has inspired them the most to learn on their own and continue to experiment with image analysis using Matlab.

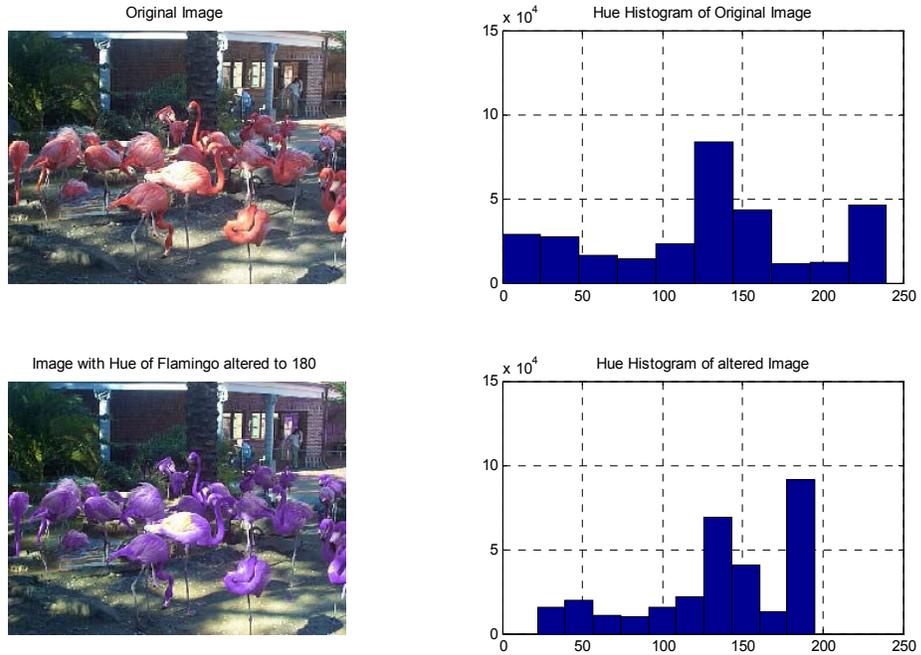


Figure 7: Hue Alteration of Flamingo Birds and Hue Histograms

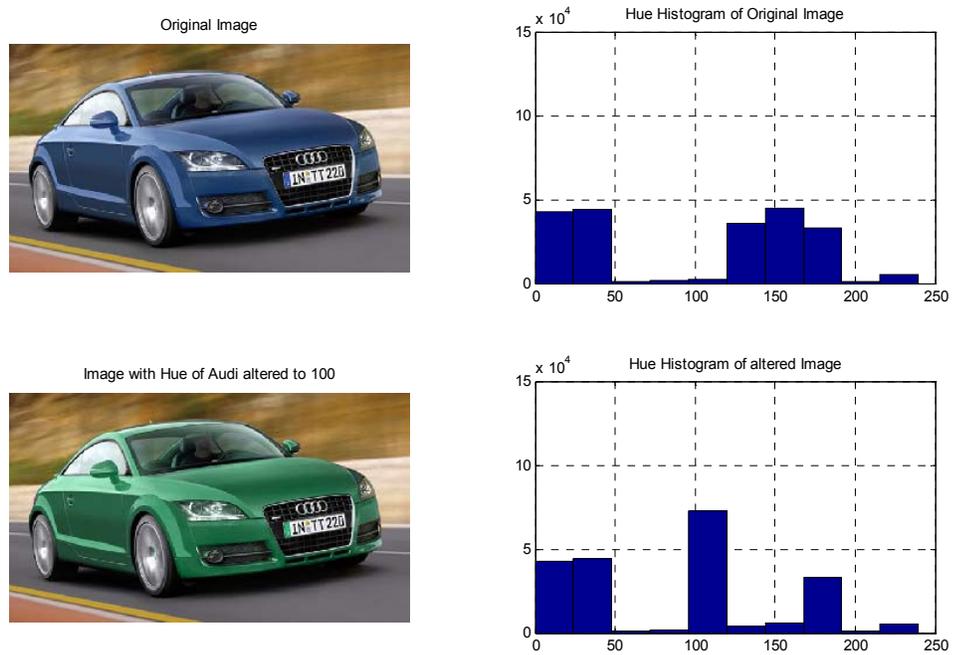


Figure 8: Hue Alteration of "Audi" Car and Hue Histograms

4.0 Learning Objectives, Outcomes, and Assessment

Following ABET Engineering Accreditation Commission (EAC) guidelines, the following outcomes were identified and assessed for the course; the alphabetical order is in accordance with Engineering Criteria (EC 2000):

- (a) an ability to apply knowledge of mathematics, science and engineering
- (b) an ability to design and conduct experiments, as well as to analyze and interpret data
- (d) an ability to function on multidisciplinary teams
- (g) an ability to communicate effectively
- (i) a recognition of the need for, and an ability to engage in lifelong learning
- (k) an ability to use techniques, skills, and modern engineering practice.

Course Objectives	Course Assessment Methods	Applicable ABET Outcomes <i>Competencies a through k</i>	Extent of Coverage of ABET Outcomes <i>1- Some, 2-Modest, 3-Significant</i>
1.To provide students with fundamental analysis and computational skills to effectively use the basic MATLAB software.	Tests, Homework, Quizzes, and Final Examination Class Discussions	a, b g	a -3, b -2 g - 1
2.To provide students with basic knowledge of linear algebra, numerical analysis, solution systems of linear equations, iterative solution of nonlinear equations, numerical integration and numerical solution of differential equations.	Tests, Homework, Quizzes, and Final Examination Class Discussions	a, b, i,k g	a -3, b -2,i - 1 g - 1
3. To provide students with basic programming skills so that they are capable of writing MATLAB script files and function files as well as effectively use pre-written function files in MATLAB.	Tests, Homework, Quizzes, and Final Examination Project Class Discussions	a,b,i,k b, d, i, k g	a-3, b-2, i-2 b-3,d-3, i-3,k-3 g-2 g-1
4. To provide students with data visualization skills using 2-D and 3-D graphics in MATLAB.	Tests, Homework, Quizzes, and Final Examination Project Class Discussions	b b,d,i,k g	b-3 b-3,d-3,k-3 g-2 g-1
5. Provide project assignments in robotics, remote sensing, and/or image analysis that will utilize MATLAB programming skills acquired in class.	Project Report and Oral Presentation	a,b,g,d,i,k	a-2, b-2, g -2, d-2 i-3, k-3

TABLE 1: Learning Objectives, Outcomes and Assessment Methods

The learning objectives, outcomes, and assessment for the course are summarized in Table 1. The grades were calculated using the following weights/percentages:

Attendance and Class Response - 5 %;
Hw and Lab Assignments : 15 %
Exam – I and Mid-Term : 20 % , Exam II : 15 %
Project: 20 %; Final : 25 %.

5.0 Conclusion

The author has learned from informal discussion with the students that while they struggled with the initial part of the semester in the course, they became comfortable with the content and were excited to actively work on the project. The teamwork was not always perfect, but most felt having teammates to bounce concepts around had a positive impact on their learning. A significant number of the students commented that numerical solutions of differential equations undertaken during the course and the easy to use graphing capabilities have enabled them to understand the physical manifestation of differential equations and reinforced mathematical fundamentals. As mentioned before, the course covered the first seven chapters of the text and also introduced students to the “Image Processing Toolbox” of Matlab. The subsequent chapters (8 -14) have relevance to courses the students will take in the future. The students readily recognized that they will be able to use the book as a reference in future courses and professional life. During the course lectures the author also mentioned about the other toolboxes that Matlab offers, and the extensive support material available in the Mathworks website ^[27]. By the end of the course, most students realized how the course was related to the ABET outcomes (i,k) :

- (i) a recognition of the need for, and an ability to engage in life-long learning; and
- (k) an ability to use the techniques, skills, and modern engineering practice, in a more meaningful way.

The “Image Processing” aspects integrated with the course project not only provided a challenging dimension to the course that motivated learning but attracted some students to participate in the ongoing “AIRSPACES” and “ECPA” projects.

6.0 Acknowledgment

The author would like to acknowledge the efforts of all the students who have taken the course and in particular Mr. Uditha Poddalgoda, Ms. Whitney Smith, Mr. Gabriel Ladd, Mr. Omar Omar, Ms. Amy Jarrett, Mr. Brian DeLeon, Mr. Olatunde Alade, Mr. Mathew Watson, Mr. Brian Vetter, Mr. Henok Gugsu, Mr. Bushara Dosa, Mr. Robert Fries, Mr. Shawn Ward and others. Support from USDA and Maryland Space Grant Consortium in the development and execution of ECPA and AIRSPACES projects also contributed to enhancing the course.

Bibliography

1. Magrab, E. et. al., "An Engineer's Guide to MATLAB", Prentice Hall, 2005
2. Moore, H., "MATLAB for Engineers", Prentice Hall, 2007
3. Palm, W.J., "A Concise Introduction to MATLAB" McGraw Hill 2007
4. Sticklen, J., "An Introduction to Technical Problem Solving with MATLAB v.7", Great Lakes Press, Inc., 2005
5. Turcotte, L.H., and Wilson, H.B., "Computer Application in Mechanics of Material Using MATLAB", Prentice Hall, 1998
6. Harper, B.D., "Solving Statics Problems in MATLAB", John Wiley, 2002.
7. Soutas-Little, R.W., and Inman, D.J., "MATLAB Supplement: Engineering Mechanics – Dynamics", Prentice Hall, 2000.
8. Franklin, G., and Powell, J., "Feedback Control of Dynamic Systems", 4th. Edition, Prentice Hall, 2002
9. Martinez, W.L., and Martinez, A.R., "Computational Statistics Handbook with MATLAB", Chapman and Hall/CRC, 2002
10. Gonzalez, R.C., Woods, E., and Eddins, S., Digital Image Processing Using MATLAB, Prentice Hall 2004
11. Proakis, J.G., and Salehi, M., "Contemporary Communication Systems Using MATLAB", Brooks/Cole, Publishing Co., 2000
12. Azemi, A., and Pauley, L., "Using Matlab to Teach Introductory Computer-Programming Course for Engineers", *Proceedings of the 2004 ASEE Annual Conference*, Salt Lake City, Utah.
13. Azemi, A., and Pauley, L., "Teaching the introductory computer programming course for engineers using Matlab and some exposure to C," *Proceedings of the 2006 ASEE Annual Conference*, Chicago, Illinois.
14. Bualan, R., "Teaching Computer Programming Skills to First Year Engineering Students Using Fun Animation in Matlab" *Proceedings of the 2006 ASEE Annual Conference*, Chicago, Illinois.
15. Nagchaudhuri, A., Mitra, M., Brooks, C., Earl, T.J., Ladd, G., and Bland, G., "Integration of Mechatronics, Geospatial Information Technology, and Remote Sensing in Agriculture and Environmental Stewardship", *Proceedings of 2006 International Mechanical Engineering Congress and Exposition (IMECE'06)*, November 5-10, Chicago, IL., 2006
16. Nagchaudhuri, A., William, M., Mitra, M., Conry, R.W., and Bland G., "Vertical Integration of Students and Mentoring Activities Pave the Way for Phase-II of UMES-NASA Collaborative Experiential Learning Project", Session 2470, Proceedings of the 2004 ASEE Annual Conference, Salt Lake City, June 2004, CD ROM
17. Nagchaudhuri, A., " Integration of Modern Software Tools for Virtual Prototyping and Intelligent Control of Mechanical Systems", Proceedings of 2003 International Mechanical Engineering Congress and Exposition (IMECE'03), Washington D.C., Nov. 16-21, 2003
18. Nagchaudhuri, A., Srinivasan, S.S, Wood, J. and Stockus, A., "Mechatronics Laboratory at UMES: A Platform to Promote Synergy in Education and Research Across Disciplinary Boundaries", Proceedings of 2003 International Mechanical Engineering Congress and Exposition (IMECE'03), Washington D.C., November 16-21, 2003
19. Nagchaudhuri, A., and Bland, G., " UMES-AIR: A NASA-UMES Collaborative Project to Promote Experiential Learning & Research in Multidisciplinary Teams for SMET Students", *Journal of SMET Education: Innovations and Research*. July –Dec, 2002.
20. Multispec (Freeware) Available @ URL - <http://cobweb.ecn.purdue.edu/~biehl/MultiSpec/>
21. Ladd, G., Nagchaudhuri, A., Earl, T.J., Bland, G, Mitra, M., "Rectification, Georeferencing, and Mosaicking of Images Acquired with Remotely Operated Aerial Platforms, Proceedings

- of Annual Conference of American Society for Photogrammetry and Remote Sensing, Reno, Nevada, May, 2006. (CD ROM)
22. Mosaicking Tutorial (Matlab) <http://www.pages.drexel.edu/~sis26/MosaickingTutorial.htm>
 23. Clay, D.E., Kim, K-I, et. al., “Characterizing Water and Nitrogen Stress in Corn Using Remote Sensing”, Agronomy Journal, 98(3), pp., 579-587, April 11, 2006.
 24. Terrahawk Imaging System <http://www.terraverdetech.com>
 25. HSV Color Space, “Wikipedia the Free Encyclopedia”, Available @ URL http://en.wikipedia.org/wiki/HSV_color_space
 26. RGB Color Space, “Wikipedia the Free Encyclopedia”, Available @ URL <http://en.wikipedia.org/wiki/RGB>
 27. Matlab :- Mathworks Inc. <http://www.mathworks.com>