Undergraduate Computer Vision Curriculum to Complement a Robotics Program

Randy P. Broussard, Jenelle Armstrong Piepmeier
United States Naval Academy
Weapons and System Engineering Department

Abstract
This article discusses a computer vision curriculum, including laboratory exercises, which is suitable for undergraduate engineering students. While classroom and laboratory exercises focus on off-line computation, on-line implementation can be achieved with simple equipment such as web-cams. Exercises include a sidewalk or line following exercise utilizing the Hough transform, a face recognition using eigenfaces, barcode reading, handwriting recognition, and sign language recognition. Data-set development for these exercises is also discussed. MATLAB and the Image Processing Toolbox are utilized to allow students to focus on higher-level understanding of commonly available image processing tools. The use of advanced tools allows students to attempt and finish meaningful examples. This paper focuses on exercises that serve as a useful complement to robotics curriculum and student robotics projects.

1. Introduction

This paper describes a single semester computer vision course tailored to fourth year undergraduate students with strong engineering backgrounds and moderate computer programming skills. The students referred to in this paper are in the Weapons and Systems Engineering department at the United States Naval Academy. They have a strong mathematics background and a good foundation in Fourier transforms and frequency analysis. They have completed one MATLAB and one "C++" programming course, but have no computer vision or image processing background. Within the major, each student is required to complete a year long senior project. A large number of these projects are robotics-related. To support these projects, overviews of advanced topics, such as face recognition and computational intelligence, are included. We've found these topics also serve to foster long-term interest in the area of computer vision. To support the unique mission of the U. S. Naval Academy, this curriculum favors object identification topics. Each pattern recognition approach is compared and contrasted to the target recognition technologies currently used within the military. The state-of-the-art is discussed to give the student an understanding of capabilities and limitations of the technologies they may encounter during their military careers. We use *Computer Vision* by Shapiro and Stockman as a text [1].

2. Background

A comprehensive survey of computer vision education has been compiled by Bebis *et al* in [2]. Bebis correctly points out that the computer vision field has grown rapidly in the past decade, and yet it is not well represented the curriculum most institutions. For over a decade, computer vision has been a part of the robotics curriculum in the Systems Engineering Department at the Naval Academy [3]. The course presented in this article serves as a stand-alone, yet complementary course to the robotic offerings within our department. In a survey of computer vision courses taken by Maxwell [4] five categories were identified: classic image processing, classic computer vision, application oriented, focused, and high-level. Our course would probably be categorized as classic computer vision: a survey course covering a variety of topics. However, military applications are frequently mentioned due to the context of our institution.

3. Curriculum Overview

The curriculum is composed of a single course given to students in the first semester of their senior year. Two primary computer vision topics are covered; image processing and pattern recognition. The image-processing portion addresses image acquisition, filtering, transformation and thresholding. The pattern recognition portion teaches feature computation, feature selection, classifier selection, and classifier usage. The course goal is to develop students who are proficient in selecting and applying various computer vision tools such as image filters and statistical classifiers. The students implement some tools, but most are provided by the professor, or available in MATLAB or MATLAB's Image Processing Toolbox. These tools are used to build larger, more complex computer vision systems. As systems engineers, our students must understand the design and functionality of component pieces, but their overarching role is to build higher-level functionality from those components.

The curriculum is designed to first teach the students the design and functionality of various computer vision tools. As each tool is learned, an experiment is performed to demonstrate the tool's capabilities and limitations. The topics are presented in an order that allows several tools to be grouped together to form a higher-level experiment on real world images. An example of this would be to group edge detection, automatic thresholding, and Hough transforms to build a road detection system that could be used for vehicle navigation.

A broad range of topics are covered within the course. The image processing portion covers the following topics: Image formats, camera physics, camera types, histograms, automatic thresholding, neighborhood operations, noise removal, edge and corner detection, line identification, image filtering, morphology, segmentation. The pattern recognition portion covers these additional topics: Feature computation, feature normalization and invariance, feature selection, object classification, classifier types, Feature-Template-Model based vision, 3D vision, face recognition and sign language recognition.

4. Experiments

Many high level experiments such as parts recognition, road detection, handwriting recognition, and face recognition are used to teach the students to design computer vision systems using standard, readily available tools and functions. Real world images are used whenever possible. MATLAB image processing routines are used only after the students have grasped the basic theory behind the function. The following sections describe the six major experiments performed within the course. Each experiment increases complexity and skill requirements compared to the preceding experiment.

4.1. Bar Code reading

In this experiment, eight 24-bit digital camera images of barcode labels are used to exercise the student's ability to utilize histograms and thresholds. The goal of the exercise is to have the student design an automatic algorithm that cleanly thresholds the image to retain only the bars representing the barcode. The thresholded image is then passed to an instructor written barcode reading function that works only on very clean images. The eight images have been manipulated so that each requires a different threshold value to cleanly retain the barcode information. The manipulation consisted of scale and shifts in brightness, contrast and gamma. Histogram equalization was also used as a manipulation technique. Use of these manipulated images force the student to systematically search the histogram for the desired information and select a threshold to cleanly retain only that information.

Figure 1 Sample of barcode images and their respective histograms used in thresholding algorithm.

4.2. Road Navigation

This two-part lab starts by examining the strengths and weaknesses of various popular edge and corner detecting techniques. The commonly used Roberts and Prewitt operators are compared with the SUSAN edge and corner detector developed by Smith [5]. The student implement the Image Processing Toolbox edge detectors for the former, and construct the algorithm for the later. Simple images containing lines of various widths are processed. Afterwards, a real-world image of building with a sidewalk is used demonstrated the noise rejection ability of each technique.

The second part of the lab examines straight-line detection using the Hough transform. An instructor-processed image is provided to the students to prevent errors from part one carrying forward into the next part of the experiment. The image is an edge image (generated using the SUSAN edge detection method) from the previous lab. The goal of the experiment is to identify the vertically angled lines that represent the edges of the sidewalk. The building contains many horizontal straight that add complexity to the problem. Figures 2-3 show the original image and an edge image generated using the SUSAN edge detector. Specific lines identified using Hough Transform are highlighted.

4.3. Feature Selection techniques and Classifier comparisons

This is another two-part experiment. The first part of the experiment demonstrates the value of various feature selection techniques. The students utilize scatter plots, cross-correlation, and feed forward feature selection to identify salient features. Ten feature sets are constructed with varying degrees of saliency. Several of the features are entirely composed of white noise. The selected features from each technique are processed with a Bayesian classifier to determine saliency. The power and ease of use of each technique is compared and contrasted.

The second part of the experiment compares the strengths and weaknesses of several classifier algorithms. The students compare the classification performance of the K Nearest Neighbor, Bayesian, and Multi-Layer Perceptron (an artificial neural network) algorithms. After gaining a moderate understanding of each algorithm, the students use instructor implemented algorithms of each classifier. Several feature sets are processed using each classifier. Each feature set is designed the weakness of a particular classifier. Strengths, weaknesses, and computation costs are discussed.

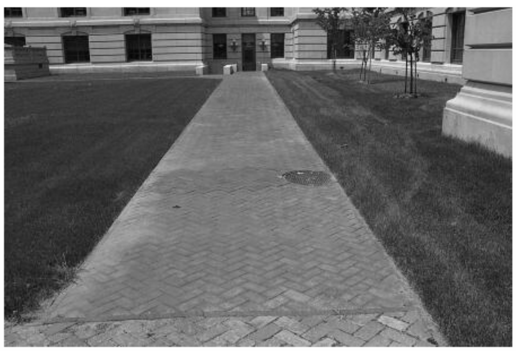

Figure 2 Sidewalk image taken outside of Maury Hall at USNA used for edge detection assignment.

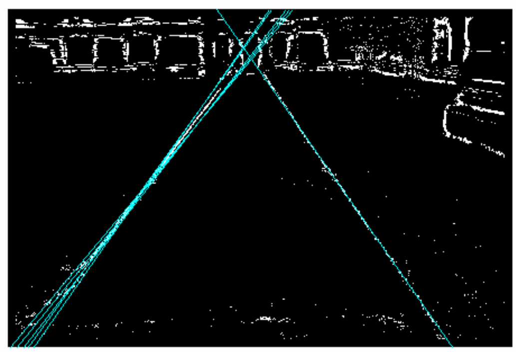

Figure 3 Hough Transform results identifying sidewalk edges using Hough transform.

4.4. Handwriting Recognition Project

In this experiment, the students design and implement a handwriting recognition system. The experiment was designed to utilize all skill taught in the prior weeks of the course. The students were required to perform all image processing, feature computation, feature selection, and classifier tuning. The use of MATLAB functions and instructor provided classifiers and information display functions was allowed. The classification accuracy of the system is tuned and tested on a handwriting database developed by the National Institute for Standard and Technology (NIST)[1]. The subset of the database we used contained ten characters written by twenty individuals. Both a training database and a test database are provided. The student designed systems varied greatly in approach and accuracy. Some students chose to use feature based pattern recognition while others used a template-based paradigm. The template-based approach combined with several feature

---

[1] Available on the World Wide Web at: trec.nist.gov

based measurements typically achieved the best recognition accuracy. A competition atmosphere was created within this experiment. Grades were directly linked to recognition accuracy. The top three students were exempted from turning in a write-up, which enhanced competition.

4.5. Face Recognition

In this unit, the students were given the assignment of reading the article by Turk and Pentland on the use of eigenfaces for face recognition [6]. The students were supplied with a sample database[2] and instructor-provided code. The code takes the database, computes a set of eigenfaces. By projecting the image onto the eigenfaces, a set of numbers is computed for each image. These numbers are used as features that correspond to a face. The numbers are computed for both at training and testing set, and the pattern recognition tools from the previous lab are utilized for identification.

Figure 2 Sample images from the face database. This group constructed a database of 60 images. Inconsistencies in scale caused some recognition difficulty.

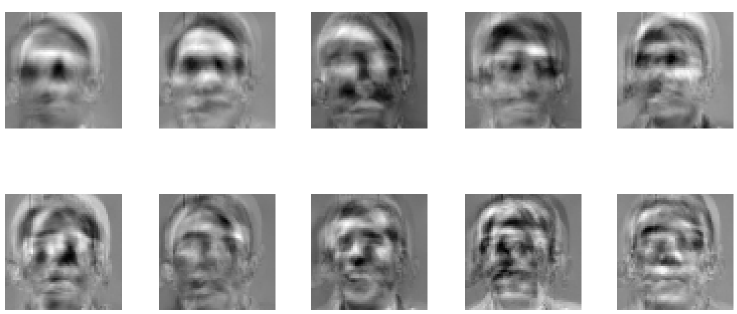

Figure 3 The first ten eigenfaces from the student created database. The students noticed that certain eigenfaces seemed to closely resemble particular individuals. They also noticed how various eigenfaces seemed to concentrate on specific areas such as the forehead, hair, or eyes.

After testing the code on the provided database, the students were tasked with developing their own database and testing the robustness of the method to various changes such as orientation, scale, or the presence of a hat. Students quickly learned the sensitivity of the

[2] The face database used was the AT&T Laboratories (Cambridge) Face Database. Available on the World Wide Web at http://www.uk.research.att.com/pub/data/att_faces.zip.

method to variations in scale, location, and lighting. This lab was very popular and one group has expressed intentions of implementing it in the senior project. Figure 4 shows sample faces and Figure 5 shows sample eigenfaces.

4.6. Sign Language Recognition

As a final project, students are given a database with ten images. Each image is of the first ten sign language symbols 'a' through 'j'. The symbols are constrained to one row. The students are told to develop code that will isolate each symbol and identify it by the corresponding written character. They use the provided database as a training set. This project requires many of the skills learned throughout the semester. On the day of the final exam, they must use their code to identify a test image. Their code reads in the image, isolates the symbols, classifies the symbol, and displays the results. A report describing their results and the approach they took is a required submission. This assignment is ideal for systems engineers because it forces them to tackle a problem by integrating the tools they have learned throughout the semester into a cohesive product.

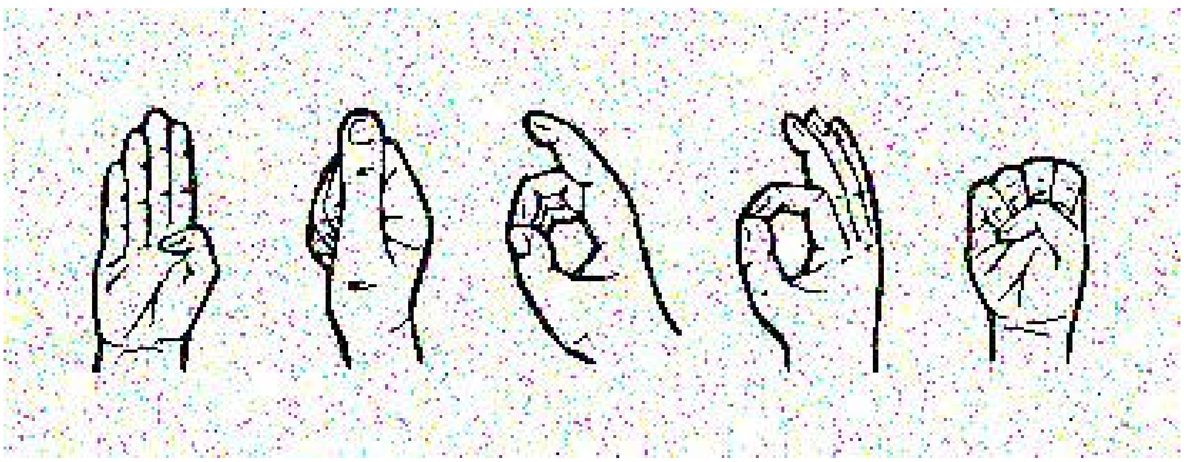

Figure 4 The characters displayed in this image spell the word "badge". Noise has been added to the image.

5. Specialized Functions and Databases

Database construction proved to be the most challenging part of implementing this course. A course of is heavily reliant upon data, image processing tools, and pattern recognition tools. The early portion of the course requires simple images with benign backgrounds to allow the advantages of each image processing technique to be readily apparent. However, the higher level image processing experiments required images that contained multiple processing challenges. When the pattern recognition portion is reached, relatively large databases are required to achieve statistically significant results. In the feature selection and classifier comparison experiment, several hundred observations of multiple objects were used to fill the database. The handwriting recognition experiment utilized over 500 handwritten characters from 20 different people. The course required a database be constructed or obtained for each of the following topics: Edge and corner detection, road navigation, nuts and bolts identification, barcode reading, space clouds segmentation, hamburger bun seed count, feature selection, classifier comparison, handwriting recognition, sign language recognition and human face recognition.

To allow the student to focus on creating higher-level object recognition systems, tools such as statistical classifiers and confusion matrix functions were provided. If the student were to spend time implementing these functions, not enough time would remain in a single semester for the student to build complex systems. Programs that produced scatter plots, confusion matrixes, feature normalization and enhanced sub-sampled images were implemented for student use. A K-Nearest Neighbor classifier, Bayesian classifier, and a Multi-Layer Perceptron artificial neural network were also implemented. The software for the face recognition lab placed the eigenface feature data in a matrix format similar to that used for the object recognition project. This allowed students to reuse the K Nearest Neighbor functions for identification.

6. Conclusions

We have presented an undergraduate level computer vision curriculum that can be covered in a single semester. The experiments develop many skills desired within advanced stationary or mobile robotics programs. Several laboratory exercises have been described including automatic thresholding, edge detection, handwriting recognition, and face recognition. We have delineated and characterized the computer functions and databases required to implement a course of this type. The course covers a considerable number of topics, but due of the visual nature of the material; the students remained engaged and excited. Several senior projects have incorporated topics from the course. An automated vehicle used the Hough transform technique for following a line on the floor. Several projects this year are implementing vision for target tracking on mobile vehicles. Students devising a robotic chess playing device have proposed a face-recognition module that will identify the opponent and tailor the difficulty level to the individual. Two consecutive years of student critiques have described the course as one of the most applicable and interesting course they have taken in their undergraduate curriculum.

References

[1] G. Stockman and L. Shapiro, *Computer Vision.* Englewood Cliffs, NJ: Prentice-Hall 2001.

[2] G. Bebis, D. Egbert, and M. Shah. “Review of computer vision education,” *IEEE Transactions on Education,* (46)1:2-21, Feb 2003.

[3] W. Clement, “An instructional robotics and machine vision laboratory,” *IEEE Transactions on Education.,* 37:87-90, Feb 1994.

[4] B. Maxwell, “Teaching computer vision to computer scientists: Issues and a comparative textbook review,” *Int. J. Pattern Recognition Artificial Intelligence*, (15)5:757-773, 2001.

[5] S.M. Smith and J.M. Brady. “SUSAN - a new approach to low level image processing.” *Int. Journal of Computer Vision*, (23)1:45--78, May 1997.

[6] M. A. Turk and A. P. Pentland. “Face recognition using eigenfaces,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 586 -- 591, June 1991.

RANDY P. BROUSSARD
Randy P. Broussard received a BS in electrical engineering from Tulane University in 1986, an MS in computer engineering from Florida Institute of Technology in 1991, and a PhD in electrical engineering from the Air Force Institute of Technology in 1997. He is currently an assistant professor in the Department of Systems Engineering at the United States Naval Academy where he teaches computer engineering, computer vision, and control systems. His research includes the areas of automatic target recognition, neural networks, computer vision, stock market timing and breast cancer detection.

JENELLE ARMSTRONG PIEPMEIER
Jenelle Armstrong Piepmeier received a Bachelor of Science in Engineering from LeTourneau University in 1993, Master of Science in Mechanical Engineering and Doctor of Philosophy in Mechanical Engineering from Georgia Institute of Technology in 1995 and 1999, respectively. Since 1999, she has been on the faculty of the Systems Engineering Department of The United States Naval Academy as an Assistant Professor. Her research interests include vision-guided robotics and vision-based wave sensing.