Integrating Teaching with Research Using Computer Vision Examples

George Bebis and Dwight Egbert
Computer Vision Laboratory
Department of Computer Science
University of Nevada
Reno, NV 89557, USA
(bebis,egbert)@cs.unr.edu

Abstract

College education has historically, for the most part, focused on the products of scientific investigation. Many students today acquire their learning of a particular field through structured courses which restrict the students to passive watching rather than active doing. Preparing our students to be thinking individuals and motivating as many of them as possible to pursue graduate studies is absolutely vital to our society. Student research is an invaluable tool to be used in this process. This paper reviews recent and on-going innovative efforts to integrate teaching with research using computer vision examples. In particular, we emphasize our efforts at the University of Nevada, Reno (UNR) to integrate computer vision examples in several undergraduate core courses. These efforts are part of a larger project called "Combined Research-Curriculum Development (CRCD) in Computer Vision" which is funded by the National Science Foundation (NSF).

Keywords: university education, computer vision, teaching, research.

1. Introduction

There has been a strong movement lately to involve students in research [1][2]. Immersing students into research is a key factor for enhancing their critical thinking, creativity, self-confidence, ability to work on collaborative projects and motivating them to pursue graduate studies and engage in lifelong learning. Current science education standards [3], however, reflect increased emphasis on teaching science "facts" and decreased emphasis on teaching how those facts are generated, how they are used to understand complex systems, and how they may be applied to new and complex situations. Exposing students to real data and scientific inquiry experiences can help them realize that science is a process instead of a collection of facts to memorize. Student research is an invaluable tool to be used in this process.

In this paper, we present a review of recent and on-going innovative efforts for integrating research into the curriculum using computer vision examples. Computer vision is an excellent tool for demonstrating general principles in computer science and engineering, and for improving learning by increasing student motivation. In particular, we emphasize our efforts at UNR to integrate computer vision examples in several undergraduate core courses and to develop a more comprehensive program for combining teaching with research.

2. Traditional methods for integrating teaching with research

The three most popular approaches for exposing students to computer vision research are: (1) joining faculty research teams, (2) taking elective courses and independent studies, and (3) getting involved in summer research. These approaches have demonstrated good success; however, they suffer from several drawbacks. The most serious drawback of the first approach is that it targets a rather small number of students. The problem with the second approach is that many students may not have the opportunity to take the elective courses or independent studies as a result of constraints imposed by their degree programs. The last approach tries to alleviate these problems through intensive summer Seminars; however, it still targets a rather small student population. Another serious problem is that these approaches lack sufficient organization and planning of activities. Involvement in research projects and independent studies are ad-hoc in nature, and these opportunities can change significantly from semester to semester. The summer research programs are more established and consistent; however, they only occur during the summer. Educators who have been involved in such activities (e.g., Research Experiences for Undergraduates (REU)) agree that a research experience that lasts only a few months in the summer does not allow sufficient time to complete a serious project [4][5].

3. Rationale for Using Computer Vision

Computer vision is an ideal area for integrating research with teaching. First, it has an immediate appeal to most students due to their intimate relationship to visual experience and people's fascination with this sense. Students will have the opportunity to literally "see" the results of applying theory to solve practical problems. Thus, computer vision research results can provide a high level of motivation to students. Second, it is interdisciplinary; it relates to computer science, mathematics, physics, and engineering, and, most important, to many important application areas such as medicine, robotics, manufacturing and geology. This will assist students in attaining a higher level of competence in science, mathematics, engineering, and technology areas, issues that have been raised up by a recent advisory committee to NSF [1]. It is also an excellent learning tool for teaching students to integrate and use their acquired knowledge, while at the same time providing a high level of motivation. Finally, computer vision systems will soon become commonplace and there will be strong industry demand for computer vision scientists and engineers. As a result of this experience, many students may consider pursuing careers in computer vision. For the majority of the students who will not pursue a career in computer vision, such a background will prove helpful in other areas such as pattern recognition, graphics, robotics, multimedia, virtual reality and medical imaging.

4. Integrating Research into Computer Vision Courses

In a fast developing field such as computer vision, education and research should not be independent from each other. The major challenge for computer vision educators, however, is how to cover the basic theory, while at the same time expose the students to state-of-the-art research in the field. Usually, research results are integrated in advanced, graduate level, computer vision courses through lectures and assignments. In this case, the integration can be done with relative success and usually leads to student publications [6]. Integrating research results in introductory computer vision courses is usually not an option because of time constraints, or it happens only as part of a final project. Despite the difficulties, there exist several documented efforts to integrate research results in introductory computer vision courses using a *student-centered* learning approach [7][8][9]. The two most common forms of this approach are *problem-based* learning and *seminar-based* learning. Seminar-based approaches focus on group discussion and have less emphasis on implementation. Problem-based approaches, on the other hand, emphasize research and implementation using group work.

The undergraduate computer vision course at California State University at Hayward [8] follows a problem-based approach by integrating experimental-based learning experiences (ELE) [10][11]. The main goal was to transfer effectively to the students both important skills and knowledge. Following the definition given in [8] of an experimental learning experience being the presentation of a problem or task with no solution given, students were required to do "research" in order to solve them. That is, they defined the problem, reviewed the literature, broke down the problem, investigated potential solutions, implemented, tested, and documented the results. The emphasis was not only on imparting to students computer vision knowledge but also on instilling in them investigative skills and appropriate habits that would allow them to tackle unsolved computer vision problems. The course material included some traditional topics from image processing and computer vision (e.g., pixel-based and area-based operations, segmentation, recognition, etc.) and was structured around the ELEs. A sample of three ELE examples are given in [8]. Very similar objectives were defined for a computer vision course taught at Sheffield Hallam University in UK [9]. In contrast to traditional computer vision courses where the goal is to familiarize the students with a variety of techniques, the goal of this course was to instill appropriate habits in the students. The course targeted real-world computer vision applications (e.g., in manufacturing), and the objective was to teach students how to evaluate and identify appropriate prior knowledge in order to design successful machine vision systems. Through a number of inquiry-based, group assignments, the students learned about lighting techniques, cameras, lenses, digital image processing, and interfacing.

A seminar-based approach was compared with a problem-based approach in an undergraduate computer vision course at the University of Otago in New Zealand [12]. Both approaches were found to be more effective than the traditional approach based on lectures. Between the two approaches, the problem-based approach was found to be more effective. In the seminar-based approach, a particular computer vision topic was discussed every week. The students were required to read in advance a "classic" paper in the field, and to find and review a recent, related journal paper on their own. The review was based on a number of questions similar to those found in journal paper review forms. They were also expected to work on a computer vision project chosen on their own and submit a report at the end. The seminar-based approach achieved a number of important objectives, such as teaching the students the essentials of computer vision, showing them how to search the literature to find new techniques, and helping them to develop critical-thinking skills. The main weakness of the seminar-based approach was that the students were not able to achieve an in-depth understanding of the techniques discussed because of time constraints and high-level discussions of the topics.

The problem-based approach attempts to alleviate some of the problems associated with the seminar-based approach by being more focused and offering more hands-on experiences to the students. The course was divided into six units, including the following topics: reconstruction and filtering, edge detection, shape representation, object recognition, stereo correspondence, and optical flow. A classic paper or a textbook chapter was assigned for reading before each session. Students were also required to review a recent, related paper as in the seminar-based approach. For each unit, the students had to implement a classic technique and submit a report. Overall, the problem-based approach was found to be more effective than the seminar-based approach despite the fact that fewer topics were covered. Concentrating on a few techniques, trying to teach the skills of thinking, analysis, and evaluation seems more important than teaching more classic techniques.

Traditionally, computer vision is taught in bottom-up fashion; that is, students are taught the fundamentals (i.e., background knowledge) before being exposed to primary material. Recently, the computer vision course at the University of Otago was taught in a top-down fashion, based on the idea of "using students to teach students" [13][14]. The approach was problem-based again;

however, the students were exposed to material that required knowledge they did not previously have. To cover their deficiencies in background knowledge, they had to work backwards to teach themselves what they needed to know. This scenario was inspired from the way active researchers fill their own gaps in background knowledge. In particular, the students reviewed recent computer vision papers and chose a set of background topics with the help of the instructors. Each student then worked on one background topic and prepared a written tutorial for use by the others. Each student received peer reviews of their tutorials from the other students and the instructors. This "background skills" unit was implemented in a six-week segment that was part of a two-semester computer vision course. The rest of the course followed the problem-based approach. Compared to previous years, students performed better in the course when the top-down approach was used.

5. Integrating Research into Core Courses

Integrating research results in elective courses cannot reach a large student population. The obvious solution to this problem is integrating research results in a number of core courses, thus, giving every student the opportunity to get to know what research is all about. The learning approach employed in these core courses is more inquiry-based than research-based. The fundamental difference between inquiry- and research-based learning is the prior state of knowledge of the broader community. In research it is unknown by all; in inquiry it is only unknown by the learner. Thus, it can be safely argued that from a student perspective, pursuit of inquiry-based learning should be functionally equivalent to conducting research. Within a computer science and engineering curriculum, computer vision research results can be integrated relatively easy in a number of core courses, such as programming, data structures, and algorithms. Assigning challenging programming projects that connect what the students are learning to real applications can capture their minds and motivate them to learn more. We review below a number of efforts based on this this approach.

5.1. Programming

Two projects, one based on edge enhancement and the other on image compression, were used in CS1/CS2 courses at the Polytechnic University of Madrid to demonstrate loops, one and two dimensional arrays, files, recursion, and trees [15]. Contrast enhancement, histogram equalization, and hidden message extraction from images were used in a CS1 course at Northeastern University to demonstrate data formats, file organization, file manipulation and also to help students practice loops, decision statements, assignment statements, arithmetic expressions, and arrays [16]. At Duke University [17], the final programming assignment in a CS1 course was based on image manipulation. In a freshman engineering honors programming course at Purdue University [18], students worked in four-person teams on an image analysis project that involved halftoning, using histogram thresholding, and edge detection, using the Laplacian mask. The objectives were to reinforce programming concepts, group work, and the ability to handle large, open-ended problems.

Computer vision concepts were also used in a C++ programming course at Michigan State University (MSU) that was designed especially for non-major graduate students in science and engineering [19]. The course was designed for mature students who wanted to learn programming for their research, and the course capitalized upon their good math background and research interests to motivate them to learn C++ using computer vision concepts. Programming concepts emphasized by projects included data representation, file input/output, nested for-loops, 2D arrays, recursion, and class reusability. In another course offered at the Harvard University Extension School [20], case studies from some practical computer science application areas, including image processing and computer vision, were used to teach introductory applied computing to general-education students. The emphasis was on problem solving and design techniques rather than programming. In each case study, the students were first introduced to the subject and then asked to use previously implemented systems to perform design and problem-solving tasks. The studies related to computer vision were based on image enhancement and face recognition.

Computer vision examples have also been integrated in the Introduction to Computer Science (CS 201) course at UNR. This typical CS1 course is the first programming course for most students. The introduction of computer vision in this course comes at the end of the semester with two lectures and an image processing term project [21][22]. One lecture covers research principles and computer vision basics, followed by a second lecture with questions about the project. Image data are introduced as an example of two dimensional arrays. Basic concepts such as edges in images being changes in brightness, are explained so that students can understand the processing they will perform. The project consists of a menu-driven, image-processing program with several functions, including file read and write, negative, rotate, threshold, and basic filter. The necessary programming does not involve anything beyond what was covered in the course. Students are given a function to read a black and white PGM file and use this function as a model on which to base the writing of their own file write function. All other functions simply involve array manipulations. One of the image processing functions is an operation of the students' own choice. Many students were very creative in their choice of operations, and most students enjoyed the project. The visual feedback made them feel they could really accomplish much with only one programming course.

5.2. Data structures

Solving a problem efficiently requires knowledge of efficient algorithmic techniques. Implementing the solution efficiently, however, requires knowledge of efficient data structures. Introducing data structures to students through computer vision examples can provide a very clear demonstration of these issues in the context of a real application. Computer vision deals with algorithms that process and extract important information from images. Knowledge of the theoretical basis of these algorithms is not enough because one has to implement these algorithms efficiently using appropriate data structures. Many computer vision tasks offer a natural way to introduce basic data structures, such as arrays, queues, stacks, lists, trees, and graphs.

The data structures course at the University of South Florida (USF) has been taught several times using computer vision examples [23]. Students taking the course are not required to have any prior knowledge in image processing or analysis. To make up for the students, lack of background in image-related computation, one lecture is dedicated to discussing the fundamentals of image generation, representation, and processing. Seven programming assignments are described in [23], each one emphasizing a different data structure. Students work in groups to complete the assignments.

A similar approach has been used to integrate computer vision examples in the data structures course at UNR [24]. The main difference from the USF approach is that the assignments have been augmented with inquiry-based material to motivate the students to extend and improve the algorithms to be implemented. The goal was to improve the students' critical thinking skills and give them a taste of what research is all about. To motivate them, extra credit was offered for interesting ideas. Students were not expected to have any prior knowledge of image processing or manipulation. In the beginning of the semester, they were introduced to the fundamentals of computer vision and image processing with two lectures. The first lecture emphasized the area of computer vision and discussed promising applications, accompanied by video-based demonstrations to spark their interest. The second lecture was more technical and discussed image generation, representation, and manipulation. Four programming assignments were given to the students. The goal of the first assignment was to help them practice dynamically allocated arrays, constructors, destructors, copy-constructors, and operator overloading. The objective of the assignment was to implement an image-processing package having the capability to perform some simple image-processing operations, including addition, subtraction, negation, translation, scaling, and rotation. Examples of inquiry-based questions include implementing scaling more effectively (i.e., the basic scaling function they are asked to implement is based on sub-sampling or up-sampling) or eliminating the holes from the rotated image (i.e., the basic rotation function they are asked to implement is based on the forward transformation). Only gray-scale images are used in the first assignment. The goal of the second assignment was to help them improve their skills in using templates. The objective was to extend the capabilities of the image-processing package so that it can handle both grayscale and color images. In the third and fourth assignments, the students were asked to implement a simple system to recognize US coins from images. The third assignment illustrated stacks, queues, recursion, and running times, while the fourth assignment emphasized lists. Examples of inquiry-based questions in this case included how to use the histogram of the image to automate thresholding or how to improve the performance of coin recognition.

5.3. Hardware-related courses

Few efforts have arisen to integrate computer vision concepts in computer vision courses. One of them is the Microprocessor Engineering (CS/EE 336) course at UNR which introduces the details of microprocessor architecture, assembly language, and interfacing to an embedded single board computer (SBC) [25]. As part of the interface programming experience the students must learn about command and data transfer protocols and several file formats. The concepts of computer vision and image processing were easily incorporated through the analogy of sound (which has been covered previously) as a one-dimensional signal compared to an image as a two-dimensional signal. The computer vision lecture module for this course includes a review of the scientific research method and introduces computer vision concepts in the context of the course. For example, a digital camera is one instance of an embedded system with a microprocessor controller and a menu interface. The camera menu is compared to the SBC menu written by the students. The computer vision laboratory session is used to reinforce student understanding of the serial interface hardware, the software command protocol, and the file formats which have always been included as part of this course. The lab is structured to demonstrate the research process by having the students "discover" the transfer protocol between the computer and camera from experimental data with the help of a WEB document which describes how others are also learning the protocol through experimenting. This experiment is directly related to the course's study of SBC interfacing as well as providing an exciting application which allows the students to understand that digital images are simply streams of data.

Computer vision concepts have also been integrated in an embedded computing course [26]. Today's modern embedded computers are more sophisticated and can process still images, video, and audio. The adoption of vision-related technologies in modern consumer products makes embedded computing courses a good fit for computer vision examples. At Clemson University [26], the emphasis of the embedded computing course is on teaching students how to construct multimedia embedded computing devices. Among other things, the students also learn how to construct and program hardware to operate as a digital video camera.

5.4. Algorithms

At least one effort has been documented to integrate image-related computation concepts into algorithms [27]. In this effort, the emphasis was on the implementation of specific algorithms and data structures, empirical analysis of their performance, and comparisons with their theoretical time complexity. Four programming assignments were designed to allow the students to acquire a hands-on experience with algorithms and grasp difficult concepts. The first assignment was on representation of images. The purpose was mainly to familiarize the students with image encoding; however, they also implemented some simple functions (i.e., compute mean gray level value), measured the running time of each function, and determined the dependency of the time on the image size. The second assignment was on median filtering which requires fast sorting algorithms. The students implemented and compared various sorting techniques by varying the window size in median filtering. In the third assignment, they implemented connected components, using graph operations and disjoint-set structures. The objective was to compare the performance of connected components using a linked-list representation of disjoint sets and a disjoint-set forest. The last assignment was on image compression using Huffman codes. The purpose was to familiarize students with greedy algorithms. The running time and compression rate of the implemented compression algorithm was measured and compared using Unix's "gzip" utility.

6. An overview of the CRCD program in computer vision at UNR

Traditional student research participation programs (e.g., joining research teams, performing summer research, taking advanced courses etc.) have demonstrated considerable success in exposing students to current research paradigms. However, it is becoming more and more evident that comprehensive instructional programs, which offer systematic and constant research experiences to students, will be more effective in involving a larger body of students in research and retaining their interest and enthusiasm. Making research an integral part of the curriculum and embedding research pedagogy within the curriculum will offer a more balanced and effective educational experience.

UNR is currently supported by NSF to develop a CRCD program in computer vision. The overall goal of this project is to integrate the results of recent and ongoing research in computer vision into the computer science and engineering curriculum. In contrast to common approaches that propose integration at the senior level through the offering of advanced courses or research projects, our model attempts to achieve integration of teaching with research at all levels, leading to a comprehensive instructional program, offering systematic and constant research experiences for as many students as possible. The project seeks to accomplish this goal by immersing students into research through systematic and structured activities starting at the freshman year and continuing until graduation and graduate school, making research an integral part of each student's education.

The key idea of our model is "injecting" research results into core courses throughout the curriculum. This approach forms the "skeleton" of the UNR model, around which more traditional approaches are integrated. Computer vision research results have been incorporated into one freshman-level (i.e., introductory programming) and two junior-level core courses (i.e., data structures and microprocessors). Second, to set a strong foundation for enabling student research in computer vision, a new, junior-level course in mathematical methods for computer vision is being developed. Current and ongoing research results are being integrated into this course to make it another research experience. A new, senior and introductory graduate-level course is also being developed based on state-of-the-art research results in the area of object recognition. Finally, a "distributed" model of summer research experiences for undergraduate and graduate students is being implemented. During the summer, students (junior, senior, and graduate level) have the opportunity to do research at various research laboratories all over the country.

The main advantage of our model is that it will allow the majority of the students to be exposed to meaningful motivating, real-world topics, making them progressively more involved until they are ready to do their own research projects. For the students who decide not to pursue research more extensively, our model ensures that they will at least have some basic exposure to research experiences. For the students who choose to become more involved, we complement our model with more traditional approaches such as advanced courses, independent studies, involvement in faculty research projects, summer research, and industrial training.

We have been running this program for the last two years with great success. Specifically, a large number of UNR students have learned about computer vision in courses such as Computer Science I, Data Structures, Microprocessors, and have solved some simple "research" problems such as implementing a system to recognize US coins from gray-scale images [24][25]. Some of the teaching-research modules developed at UNR have also been transfered to community colleges in the Reno area [22]. To increase awareness of our project and motivate more people to consider integrating teaching with research, we also organized the first *IEEE Workshop on Combined Research-Curriculum Development in Computer Vision* in 2001 (in conjunction with the Computer Vision and Pattern Recognition (CVPR) conference). We have established strong collaborative links with several academic, national, and industry labs. Our students' research has produced several journal and conference publications, one patent, and has enabled us to obtain new funding. More information about UNR's CRCD program is available from *http://wwww.cs.unr.edu/CRCD*.

7. Conclusions

This paper reviewed some interesting ideas and approaches for integrating research into the curriculum using computer vision. In particular, we emphasized our activities at UNR. This review is by no means comprehensive and is based largely on results that have been reported in the literature and on information found on the World Wide Web. It is certain that more information exists that has not been reported in some publication or report. Hopefully, this review will encourage more people interested in this area to step forward and share their ideas and approaches with the rest of the community.

Acknowledgement

This work was supported by NSF under CRCD grant #0088086.

References

[1] Advisory Committee to the National Science Foundation, Directorate for Education and Human Resources, Melvin D. George (Chairman), *Shaping the Future: New Expectations for Undergraduate Education in Science, Mathematics, Engineering, and Technology, May, 1996.*

[2] D. Coppula, "Integrating teaching and research", *ASEE Prism*, December 1997.

[3] National Research Council [NRC], "National Science Education Standards, National Academy Press, Washington DC, 1996.

[4] M. Shah and K. Bowyer, "Mentoring Undergraduates in Computer Vision Research", *IEEE Workshop on Undergrad Education and Image Computation*, Puerto Rico, 1997.

[5] M. Shah and K. Bowyer, "Mentoring Undergraduates in Computer Vision Research", *IEEE Transactions on Education*, vol. 44, no. 3, pp. 252-257, 2001.

[6] B. Maxwell, "A survey of computer vision education and text resources", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 5, pp. 757-773, 2001.

[7] A. Sanchez, J. Velez, A. Moreno, and J. Esteban, "Introducing algorithm design techniques in undergraduate digital image processing courses", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 5, pp. 789-803, 2001.

[8] L. Grewe, "Effective computer vision instruction through experimental learning experiences", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 5, pp. 805-821, 2001.

[9] T. Pridmore and W. Hales, "Understanding images: an approach to the university teaching of computer vision", *Engineering Science and Education Journal*, col. 4, no. 4, pp. 161-166, 1995.

[10] N. Glasgow, *A guide to student-centered, problem-based learning*, Corwin Press, 1997.

[11] P. Ram, "Problem-based learning in undergraduate education", *J. Chem. Edu.*, vol. 76, no. 8, pp. 1122-1126, 1999.

[12] K. Novins and B. Mccane, "Incorporating primary source material into the undergraduate computer vision curriculum", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 5, pp. 775-787, 2001.

[13] M. Donelan and J. Wallace, "Peer assisted learning: a truly co-operative initiative", in *Students Supporting Students* (J. Dolan and A. Castely eds.), pp. 11-22, Staff and Educational Development Association 1998.

[14] R. Ploetzner, P. Dillenbourg, M. Praier, and D. Traum, "Learning by explaining to oneself and to others", in *Collaborative Learning: Cognitive and Computational Approaches* (P. Dillenbrourg ed.), pp. 103-121, Elsevier, Oxford, 1999.

[15] R. Jimenez-Peris, Sami-Khuri, and M. Patino-Martinez, "Adding breadth to CS1 and CS2 courses through visual and interactive programming projects", *SIGCSE'99*, pp. 252-256, 1999.

[16] H. Fell and V. Proulx, "Exploring martian planetary images; C++ exercises for CS1", *SIGCSE'97*, pp. 30-34, 1997.

[17] O. Astrachan and S. Rodger, "Animation, visualization, and interaction in CS1 assignments", *SIGCSE'98*, pp. 317-321, 1998.

[18] R. Montgomery, "Image analysis: a group assignment in programming with breadth", *ASEE/IEEE Conference on Frontiers in Education*, 1995.

[19] G. Stockman and R. Enbody, "Teaching advanced students C++ with computer vision", *IEEE Workshop on Combined Research-Curriculum Development in Computer Vision", Kauai, Hawaii, 2001.*

[20] J. Marks, W. Freeman, and H. Leitner, "Teaching applied computing without programming: a case-based introductory course for general education", *32nd SIGCSE Technical Symposium on Computer Science Education*, vol. 33, no. 1, pp. 80-84, 2001.

[21] D. Egbert, G. Bebis, M. McIntosh, N. LaTourrette and A. Mitra, "Computer vision research as a teaching tool in CS1", *ASEE/IEEE Conference on Frontiers in Education*, 2002.

[22] D. Egbert, G. Bebis, and D. Williams, "Computer Vision Research Teaching Modules for Community College Computer Science and Engineering Courses", *42nd ASEE/IEEE Frontiers in Education Conference*, 2003 (accepted).

[23] S. Sarkar and D. Goldgof, "Integrating image computation in undergraduate level data-structure education", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 12, no. 8, pp. 1071-1080, 1998.

[24] G. Bebis, "Data Structures Assignments" (http://www.cs.unr.edu/˜bebis)

[25] D. Egbert, "Microprocessors Assignment" (http://www.cs.unr.edu/˜egbert)

[26] A. Hoover, "Computer vision in undergraduate education: modern embedded computing", *IEEE Workshop on Combined-Research Curriculum in Computer Vision*, Kauai, Hawaii, 2001.

[27] E. Fink and M. Heath, "Image-processing projects for an algorithms course", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 5, pp. 859-868, 2001.