Teaching Computer Vision to Computer Scientists: Issues and a Comparative Textbook Review

Author: Bruce A. Maxwell

Affiliation: University of North Dakota, Grand Forks, ND 58202-9015

Abbreviated Title: Teaching Computer Vision to Computer Scientists

Abstract

Computer vision is a broad-based field of computer science that requires students to understand and integrate knowledge from numerous disciplines. Computer science [CS] majors, however, do not necessarily have an interdisciplinary background. In the rush to integrate, we can forget, or fail to plan for the fact that our students may not possess a broad undergraduate education. To explore the appropriateness of our education materials, this paper begins with a discussion of what we can expect CS majors to know and how we can use that knowledge to make a computer vision course a more enriching experience. The paper then provides a review of a number of the currently available computer vision textbooks. These texts differ significantly in their coverage, scope, approach, and audience. This comparative review shows that, while there are an increasing number of good textbooks available, there is still a need for new educational materials. In particular, the field would benefit from both an undergraduate computer vision text aimed at computer scientists and from a text with a stronger focus on color computer vision and its applications.

Keywords: undergraduate education, computer vision, textbook review

1 Introduction

Computer vision, the study of enabling computers to understand images, is a broad-based, interdisciplinary, and applied field in computer science. It brings together mathematics, electrical engineering, signal processing, optics, physics, psychophysics, and computational theory and algorithms. As such, it is an excellent learning tool for teaching undergraduates to integrate and use their acquired knowledge. It melds their educational experience into a single exciting and motivating application.

Unfortunately, all too often we get blank looks when we start talking about Nyquist's theorem, Fresnel reflection, or singular value decomposition [SVD]. We're happy to have engineers take computer vision because we don't have to explain Fourier transforms. We're happy to get physicists in our classes because they understand optics and EM theory; and we're happy to have mathematicians take computer vision because they know about SVD and set theory. We, the professors, have had to master these diverse topics as part of our graduate education. What we can forget is that the average computer science major is not a physicist, nor a mathematician, nor an electrical engineer. In fact, they may not have had any significant academic experience with these specialized topics.

Instead, we can expect computer scientists to be good at algorithmic concepts, data structures, file formats, and logically converting a set of directions into code. We can expect them to have a calculus level of mathematical sophistication, and we can rely somewhat on a culture of experimentation. In order to successfully teach the multidisciplinary science of computer vision to computer scientists, we must understand and take advantage of these skills and attributes.

It seems, however, that the computer vision community has geared much of its educational materials towards an audience other than undergraduate computer scientists. This group of students deserves a second look as they form a core audience in undergraduate computer vision education. In particular, the explosion of digital imagery on the world-wide web and the presence of ubiquitous image manipulation software means that computer scientists have a new-found interest in computer vision. Furthermore, as computer vision increasingly mixes with computer graphics, human-computer interfaces, networking, operating systems, and architecture, it is our CS students who will possess the other half of the interdisciplinary equation.

This paper explores some of the issues raised when teaching computer vision to computer scientists. The broad issues raised in this paper, however, apply equally well to other disciplines, as no single discipline contains all of the fundamental tools of computer vision. Section 2 begins with observations about the relative strengths and weaknesses of computer science undergraduates with respect to traditional topics in computer vision. This section also suggests how to present concepts in a manner that takes advantage of the knowledge that CS majors possess. Given this base of knowledge, Section 3 looks at some topics a computer vision course for CS majors needs to address to enable them to gain a strong understanding of the fundamentals of computer vision. Finally, Section 4 is a review of a number of currently available textbooks with a focus on their intended audience, their level and scope, and their method of presentation. The concluding section ties these threads together and suggests where improvements in educational materials are still needed in the field of computer vision.

2 Using CS Students' Knowledge

When presenting topics in computer vision, or any other field, we benefit greatly as instructors if we take advantage of students' existing knowledge and abilities. We should not present the Fourier transform to a group of CS majors in the same way we present it to an electrical engineering [EE] course. Limitations on time mean we cannot deal with subtleties or tangents, and CS students don't have a semester of linear systems analysis behind them. The latter fact is an important advantage for EE students because it allows them to place Fourier theory in a context with other system analysis tools. How then, do we present Fourier transforms and similar topics to CS majors without losing them in a stream of words and symbols?

It turns out that this problem is not unique to computer vision. Speech recognition, for example, presents similar challenges to CS students as it depends heavily on signal processing and statistical modeling methods. Computer graphics can also be a challenge since it requires students to use linear algebra for transformations, sampling theory and filtering for anti-aliasing, and physics and electro-magnetic [EM] theory for realistic illumination models.

A systematic analysis begins by looking at the topics we can expect the average junior CS major knows well enough that we can rely on previous instruction. This analysis is based upon an accredited B.S. degree in computer science, but will be different for different schools. A list of courses a typical junior CS major will have completed include Data Structures, Computer Architecture, Computer Organization (Assembly), an introductory programming series, and possibly a second programming language. By this time they may also have taken one or two computer science electives such as operating systems and automata theory, depending upon the structure of prerequisites and required courses. Based on this curricular background, we can expect them to have a working knowledge of data structures, possess reasonable programming skills, be able to create and manipulate multi-dimensional arrays, work with code libraries & packages, and be able to follow discussions of low-level computer architecture and computer system components.

The next category of skills are those we can hope for, but which not all students in a given class will necessarily possess. For example, senior and advanced junior CS majors will have additional mathematics courses like Multivariable Calculus or Linear Algebra. In addition, some students may have taken computer graphics, VLSI design, Neural Networks, AI, or Expert Systems. These courses will give them a stronger background in applied math, 3D transformations, low-level hardware, and connectionist and symbolic reasoning.

There are many topics in computer vision, however, that do not arise in a CS curriculum. Examples of these topics include: Fourier transforms (filtering), convolution (filtering), quaternions (calibration), set theory (2D operators), moments (feature extraction), Karhunen-Loeve transforms (pattern matching), and physical illumination models (physics-based vision). As noted in parentheses, these topics relate to fundamental areas of computer vision. It is also interesting to note that these topics, in particular filtering, feature extraction, and pattern matching, are commonly the *first* topics covered in a computer vision course. Thus, we are asking our CS students to immediately step into the topics that are furthest removed from their background.

There are, at least, three courses of action we can take as instructors. First, we could radically change the order of presentation so that we ease students into the more difficult topics. For example, we could focus the course on object recognition, introducing other areas of computer vision as needed to support this task. Second, we can change our method of presentation to take advantage of the skills CS students possess. In my opinion the first suggestion contains merit and should be considered within the computer vision education community. However, there are currently no educational materials supporting such a move so it is left for future discussion. The second course of action, which does not exclude the first, can immediately impact student learning using current materials.

A third course of action we could take as instructors is to require more courses as a prerequisite for a computer vision course (i.e. linear systems analysis, basic physics, image processing). However, this option will reduce access to the course at a time of growing interest and demand. It may also be overkill to require CS students to take a linear systems analysis course before taking computer vision, where they use that knowledge for one or two weeks out of fifteen. The first two options, which tailor the CV course to the student audience, ensure that those students who are interested can both get into the course and get something out of it.

Based on experience and the previous analysis of CS students’ backgrounds, the following are observations about techniques that seem to work or that certainly don't work.

First, equations out of context don't work except for a select few students. Simple formulas, like the thin lens law,

$$\frac{1}{Z_{in}} + \frac{1}{Z_{out}} = \frac{1}{f} \tag{1}$$

can be demonstrated geometrically in class and reinforced on homework or quizzes. Convolution, however, is more difficult to present as an equation since it represents a process. Beyond convolution, the continuous Fourier transform equation explains a transformation of an infinite signal onto an infinite basis space. The equation is often the first thing we present, and yet to CS students *i* and *j* are loop variables.

What does seem to work is to have students use the equations in a computational context. This gives them a sense of the process and purpose represented by the equations. Students in both the vision and speech courses readily understood convolution after implementing a simple Gaussian filter. In fact, several students, on their own initiative, wrote general convolution algorithms. For the Fourier transform, having students use pre-written code modules on well-chosen inputs allowed them to intuitively grasp the function of the transform. Thus, it is important for students to either implement the equations as part of a guided exercise or use them as prepackaged modules.

This observation leads to the following: **present concepts as algorithmically as possible to CS majors**. Convolution is a prime example of a process that can be represented as a continuous equation using integrals, a discrete equation using summations, or as an algorithm. Of the three representations, CS majors seem to find the latter the most understandable. If you give a CS major the integral version, he or she will find it difficult to use that knowledge on an assignment or exam. On the other hand, if you present convolution as a process of placing the inverted filter on the image, multiplying the overlapping elements, and summing the resulting values, students are quite good at turning that into a program. Once they have done this, then it is appropriate to introduce the other versions of convolution to connect them with the mathematical representations of their code.

The presentation of morphological operators is a second area where presenting algorithms first could improve instruction quality and student understanding. When covering dilation, for example, the presentation typically begins with a statement like the following from Haralick & Shapiro [4].

$$A \oplus B = \{c \in E^N \mid c = a + b \text{ for some } a \in A \text{ and } b \in B\} \tag{2}$$

Erosion, thickening, thinning, opening, closing, and other morphological operators are presented in a similar manner. While it is certainly possible for a student to sit down and work through the notation, think in terms of sets, and then understand a few examples, an algorithmic presentation will draw more strongly on CS students’ strengths.

The same concept, for example, can be presented as taking two binary images, one the image to be dilated (A), the other the dilation operator or structuring element (B), and moving the structuring element over the underlying image in a specific manner. To execute dilation, for example, we place the origin of the dilation operator at each “on” pixel in image A and then take the logical OR of overlapping pixels. The resulting image is the dilation of A by B. If the instructor then goes on and presents the set theory representation of this operator, the student has an intuitive grasp of the operator's meaning and can more easily understand the relationship between the action on the image and the mathematical relationship defined above in (2).

The key point is that the instructor should not automatically present the continuous version (or set theoretical version), followed by the discrete version, followed by the algorithm. If a student doesn't understand the concept until the algorithm is presented, then up to two-thirds of the presentation is wasted; the instructor does not go backwards and put the discrete and continuous versions in context. On the other hand, if we present the algorithm first--the presentation most understandable to a CS major--followed by the mathematics as support, then the student is prepared to listen and make the appropriate connections. It is important to realize that this approach does not imply weakening or diluting the material in any way. Instead, reversing the order of presentation allows students to better contextualize the material within their background knowledge.

Unfortunately, all of the textbooks reviewed herein present concepts in the traditional continuous-to-discrete-to-algorithm format. Furthermore, as noted in the reviews, except in a few texts the connection between the math and the algorithm is weak. It is up to us, the instructors, to think about our audience and adapt appropriately.

3 CS Undergraduate Vision Topics

Before developing a list of background topics a CS undergraduate computer vision course should cover, it is worthwhile identifying the main topics of an undergraduate CV course. Rather than identify a single list of topics, however, I want to propose three lists, each corresponding to a different kind of CV course: a classical computer vision course, a comprehensive computer vision and image processing course, and a computer vision course geared towards computer scientists and multimedia applications. Textbooks and example courses for the first two courses exist, but the third is still in the process of definition. The third type of course is of increasing importance because of the explosion of digital imagery and digital image capture technology. The computer vision techniques underlying visual information management are a fast-growing area of CV [3].

Based on experience and a comparison of CV textbooks, a classical computer vision course will generally include: binary image processing, filtering, edges & lines, feature detection, segmentation, optics, calibration, stereo, motion, texture, shading, shape representation, object recognition, color spaces, and color segmentation. In a CV course, as opposed to an image processing course, binary image processing and filtering are generally limited in scope. Binary image processing might consider, for example, binary region features, growing and shrinking, and erosion and dilation. Filtering is generally limited to Gaussian filters, median filters, edge-preserving filters, and edge-finding filters such as the Sobel operator. Image transforms, if they are covered, are not covered in-depth, but presented as tools for implementing other concepts.

A comprehensive computer vision and image processing [CVIP] course, possibly taught over two semesters, will generally cover the classical CV topics listed above plus additional topics more typically taught in an image processing course. These might include: in-depth coverage of image transforms, image compression, image restoration, biologically motivated vision, and pattern classification. A comprehensive CVIP course may also be the model for a graduate one-semester CV course for students with appropriate backgrounds.

For a CS/multimedia CV course the topics change slightly. Topics like binary image processing, shape-from-X, and stereo may receive less in-depth treatment. Topics that need to be added include: content-based image retrieval, object recognition based on color histograms, artificial neural network-based visual processing, 3D model extraction, and computer graphics applications of vision. Examples of the latter topic includes realistic texture synthesis, view-based morphing, 3-D model manipulation, and cinematic composition.

The key to a successful course, whichever kind you choose to teach, is to ensure that students have the necessary background to understand the material and complete the assignments. While it may be frustrating to an instructor to spend time on the basics of linear algebra or linear filters, the time saved when teaching students their application in various aspects of computer vision is worth the effort. In the author's experience teaching vision to CS undergraduates and graduates at Carnegie Mellon University and the University of North Dakota we must expect and plan to present the following topics in a classical CV course.

First, we must spend time presenting basic linear algebra. The concepts of matrix operations, transformations, and eigenvalues are necessary for calibration [6], pose recognition [13], and object recognition [17]. If you are teaching CS majors, it may be worth selecting a textbook with a good appendix on linear algebra fundamentals.

Second, we need to spend time presenting basic signal processing, including the Fourier transform, which is important for compression [15], filtering [1], texture analysis [8], and stereo [11]. We don't need to teach them how to derive it, to code it, or to symbolically find the transform of a function. What we need to do is give them an intuitive understanding of spatial frequency and an algorithmic understanding of the inputs and outputs of functions like the fast Fourier transform.

There is also the possibility of leaving out the Fourier transform, and five of the vision texts support this option (Nalwa; Davies; Haralick & Shapiro; Jain, Kasturi, & Schunk; and Sonka, Hlavac, and & Boyle). These texts do not rely upon the Fourier transform for more than a small portion of their material, if at all. Thus, for instructors teaching a classical CV course or course with a CS/multimedia focus, this is a serious option if there are concerns about time limitations or overloading students with tangential concepts.

Finally, we need to present some basic optics and the physics of light and reflection. In particular, we need to focus on how color is created, how it changes as surfaces change, and how the physical world affects it. Understanding color is important for calibration, segmentation [7][12], object recognition [16], content-based image retrieval [9], and vision applications in computer graphics [10].

For more advanced courses, it may also be necessary to present quaternions [6] and some set theory [4][15] depending upon the depth of coverage. However, an advanced vision course will have more flexibility with regard to prerequisites than an introductory vision course offered as a CS elective.

4 Textbook Reviews

Whatever we decide to include in our course, it is helpful to have a textbook that includes most or all of that material. Until recently, choosing a textbook for a computer vision course was a simple process. If you were teaching an undergraduate course you could choose from B. K. P. Horn's *Robot Vision* or V. Nalwa's *A Guided Tour of Computer Vision*; if you were teaching a graduate course, you used *Computer and Robot Vision* by R. Haralick and L. Shapiro. Since 1994, however, three new, or revised, comprehensive textbooks have become available, and at least two others will appear in 1997-98.

This section reviews six of the currently, or soon to be available texts: *Robot Vision* by B. K. P. Horn, *Computer and Robot Vision* by R. Haralick, and L. Shapiro, *Image Processing, Analysis, and Machine Vision* by M. Sonka, V. Hlavac, and R. Boyle, *Machine Vision* by R. Jain, R. Kasturi, and B. Schunck, *Machine Vision* by E. R. Davies, and *A Guided Tour of Computer Vision* by V. S. Nalwa. In addition, this section includes some general comments on *Image Analysis & Machine Vision*, a textbook in preparation by G. Stockman and L. Shapiro which should appear in the 1998-99 academic year.

Other texts the author is aware of, but which are not included in this review, are *Computer Vision* by D. Ballard and C. Brown, *Intelligence: The Eye, the Brain, and the Computer* by M. Fischler and O. Firschein, and *Practical Computer Vision using C* by J. R. Parker. Ballard and Brown's text, while a valuable algorithmic reference, is over 15 years old. Fischler and Firschein's text does not contain sufficient material on computer vision for a full-semester course on the topic, and Parker's book, from 1994, is too limited in scope. Also not included in this comparative review is *Computer Vision and Image Processing: A Practical Approach using CVIPtools* by S. Umbaugh [18]. Despite the broad title, Umbaugh's book is more appropriate as an image processing textbook and does not cover most of the major computer vision topics listed above.

Worth mentioning separately is the textbook *Digital Image Processing* by K. Castleman. The recent revision of this text contains a set of chapters on higher level image processing, including segmentation, edge detection, line-finding, binary processing, texture and shape analysis, pattern matching, color image processing, and 3D imaging, including a short section on stereo. For a two-semester EE course on image processing and computer vision this would be a good choice. However, when compared to any of the other texts reviewed herein, it does not have the same breadth of computer vision topics or depth within each topic. Furthermore, the presentation of the material makes strong assumptions about student's backgrounds and their facility with filtering and transforms. In essence, what these chapters assume is knowledge of the previous two-thirds of the text. Thus, as a textbook for a stand-alone course in computer vision, in my opinion, *Digital Image Processing* is not a viable option.

In addition to providing a general overview of each text, these reviews attempt to categorize each one according to its coverage, scope, approach, audience, and appropriateness for computer science undergraduates. An instructor should be able to use these categorizations and comparisons either to select a textbook appropriate to their own course, or as impetus to start writing their own.

4.1 Robot Vision, by B. K. P. Horn, MIT Press, 1986.

This is the classic computer vision text written by the pioneer of physics-based vision. As recently as spring 1996, this was the textbook for the Carnegie Mellon University [CMU] undergraduate computer vision course. However, the CMU course was based not around the textbook, but around a set of lecture notes developed by the faculty. The textbook was considered to be a reference only.

Part of the reason the text was used only as a reference is that the coverage in *Robot Vision*, while adequate when published, has been diminished by age. As shown in Table 1, the book presents fundamental concepts in most areas of computer vision. However, advances in algorithms, theory, and implementation require the instructor to present a large amount of material not covered in the book.

The approach Horn takes in *Robot Vision* is to use continuous mathematics to describe the interaction of light, matter, and image sensors. As a reference for the physical and geometrical mathematics of light interacting with surfaces and the camera it is excellent. His apparent audience, however, is upper-level undergraduates and graduate students with either an EE or physics background and a working knowledge of multi-variable calculus.

Besides being written for a different audience, *Robot Vision* has several drawbacks with respect to teaching undergraduate computer scientists. First, there is little discussion of the algorithms or data structures necessary to implement the basic tools of computer vision. Horn derives or provides the equations, as in Chapter 8 where he covers edges and edge finding, but does not discuss implementation issues. For undergraduates who are in the process of learning how to convert equations into code, this presents a daunting first step.

Second, many of the equations are presented in continuous form only. While this is appropriate for characterizing the physical world, the book does not make a strong connection between the discrete digital image and the continuous world. This adds to students' difficulties in translating equations into code and requires the instructor to repeatedly make the connection between continuous and discrete representations.

As a result of the problems with audience, methodology, and age, this book is not a good choice for an undergraduate computer vision course, particularly one aimed at CS majors. This is especially true with the appearance of a the new generation of comprehensive texts.

4.2 Computer and Robot Vision, R. Haralick, and L. Shapiro, Addison-Wesley, 1992.

Before moving to the newer texts, it is still worth considering *Computer and Robot Vision* by Haralick and Shapiro. This two-volume book is an in-depth examination of the fundamentals of computer vision. Its coverage of computer vision is broad, although the age of the text means the instructor needs to fill in some topics and the past few years of developments.

The first volume of this pair presents binary and 2D vision. Topics covered include thresholding and binary algorithms, region analysis, statistical pattern recognition, mathematical morphology, filtering, noise removal, edge and line finding, the facet model, 2-D texture analysis, greyscale segmentation, and arc extraction. The second volume covers 3D vision and includes illumination and physics-based vision, perspective geometry, photogrammetry, motion analysis, image registration, 3D labeling, shape representation, and knowledge-based vision. The amount of information provided on each topic makes the two-volume set appropriate for a two-semester course.

Some of the topics an instructor may wish to augment with other readings include object recognition, motion analysis, and stereo. These topics, in particular, have expanded in new directions since 1990. Object recognition, for example, which is covered in chapter 18, is limited to a few specific representations and model-matching methods. Advances in appearance-based and model-based object recognition, in particular, will need to be added to a comprehensive graduate course.

The approach taken in *Computer and Robot Vision* is rigorously mathematical, although the two volumes do present numerous algorithms and discuss implementation issues and performance. Anyone using this book would be well-served by both a linear algebra and a modern algebra course. For mathematically unsophisticated students, however, this book would pose a major stumbling block simply because of the quantity of equations. An undergraduate CS major, for example, would be quickly overwhelmed and the text would become a hindrance rather than an educational aid. Furthermore, *Computer and Robot Vision* has an extensive bibliography at the end of each section and refers to it heavily. For a graduate student who is used to accessing technical literature this is quite appropriate. For an undergraduate who is learning computer vision for the first time, however, asking them to seek out, read, and understand the primary literature at this scale is inappropriate.

This combination of coverage, audience, and approach makes *Computer and Robot Vision* an appropriate choice for a two-semester graduate seminar. For a one-semester undergraduate course for computer scientists, however, the volumes present too much information at too high a level.

4.3 Machine Vision, by R. Jain, R. Kasturi, and B. G. Schunck, McGraw-Hill, 1995.

In contrast to *Computer and Robot Vision*, the scope and detail of the material in *Machine Vision* is appropriate for undergraduates. *Machine Vision*'s coverage is broad, however, it is not as in-depth or mathematically rigorous as *Computer and Robot Vision*. In addition to binary operations, filtering, edge and line finding, and segmentation, the book also covers motion, object recognition, object representation, stereo, calibration, texture, and shading. In all of these areas, the authors have made an effort to present material that is fundamental to computer vision and not specific to a particular approach or school of thought. Because of this, the text should continue to be useful for some time.

The approach taken is biased towards an EE view of the material, but the authors do present a number of the basic vision tools--e.g. 2D operators, edge-finding, and stereo--in an algorithmic or procedural manner in addition to the mathematical presentation. The text also provides some explicit transitions from continuous mathematics to the discrete representations necessary for working with images. Thus, while the text is more appropriate for engineers, it is not unapproachable by computer scientists.

For CS majors, however, the book is not without its difficulties. The presentation of filters in Chapter 4, for example, is geared more towards a sophomore electrical engineer than a CS major. In this author's experience, EE majors using this book have no trouble grasping it, while the CS majors tend to understand the concepts only after seeing them presented algorithmically in class or as part of an assignment. Also, the presentation of stereo and calibration, which is divided into two disjoint chapters, is not coherent or well-organized. To follow the complete stereo process described in the book--calibration, imaging, registration--requires flipping between chapters which appear to have been written by different authors.

The other difficulty with *Machine Vision* as an undergraduate text is, in this author's opinion, that it is biased towards industrial applications of computer vision using greyscale or binary images. The chapters on optics, shading, and color constitute less than 40 pages of the 550 page text.

Color, however, is central to a number of areas of computer vision: segmentation, object recognition, vision in human-computer interfaces, and vision applications in computer graphics. The latter two topics are also good motivating applications for CS majors because they are connected to mainstream computer science.

This observation is relevant because one of the most important issues when teaching undergraduates is motivation; they need to see some desirable end-result for their labors. Greyscale pictures of unrecognizable assembly-line parts, while perhaps having nice properties for processing, do not encourage undergraduates to study vision. Thus, the short shrift given to color in *Machine Vision* is a hole in the text's coverage. Unfortunately, this criticism can be leveled at all of the other texts as well. The three texts *A Guided Tour of Computer Vision*, *Robot Vision*, and *Computer and Robot Vision* all contain a minimal color component. In the other texts, color computer vision is treated as a special case and given just a chapter or two. The only text that begins to break away from this approach is *Image Processing, Analysis, & Machine Vision*, where the authors have included extensions from greyscale to color images in several chapters.

Thus, given that a weak color component is common to all of the texts, for a one-semester undergraduate course *Machine Vision* is a reasonable choice, particularly for students with an EE background.

4.4 M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision,

2nd ed., PWS Publishers, to appear 1998.

This textbook, which should be available in 1998, fits in between *Machine Vision* and *Computer and Robot Vision*. In its breadth it covers more topics than any other computer vision text considered in this paper. Furthermore, its depth in some topics approaches that of *Computer and Robot Vision*, but with an algorithmic and conceptual focus rather than a rigorous mathematical one. Like many other texts, *Image Processing, Analysis, & Machine Vision* begins with filtering, edge-finding, segmentation, and 2-D shape representation. At this point, however, the text looks at various approaches to object recognition, including artificial neural networks, graph matching, and fuzzy logic. One of the strengths of this book is that it provides enough background on these topics for students to follow, although having an artificial intelligence course as prerequisite would improve students' ability to focus on the computer vision applications rather than the tools.

After object recognition, *Image Processing, Analysis, & Machine Vision* moves on to 3D vision including chapters on image understanding, 3D vision (calibration, stereo, and physics-based vision), and motion analysis. The book does not follow the same order as other texts, however, and later chapters also include mathematical morphology, linear discrete transforms, and image compression. This is actually convenient for an undergraduate instructor, because these later chapters contain much of the material that is challenging for CS majors. Their placement later in the book, and the resulting implication that previous chapters do not depend on them for understanding, means an instructor can more easily pick and choose which of these topics to cover.

The real strength of *Image Processing, Analysis, & Machine Vision* is its comprehensive, in-depth coverage of computer vision. The material is well-written and, while the mathematical formulation of methods is still the focus of the narrative, the text includes algorithms for many of the methods it covers. The text also seems to contain more example images and image comparisons than other texts, making it easier to obtain an intuitive understanding of the material. Its approach relies on a mixture of EE and CS concepts, and there is even a chapter on data structures for image analysis. Thus, with intelligent topic selection and adequate instruction this text is not inappropriate for CS majors.

The book is written at a higher level than *Machine Vision*, however, and instructors should carefully consider their prerequisites for a computer vision course before selecting this text. Based on its level and breadth, *Image Processing, Analysis, & Machine Vision* will be more accessible to students who have already had an artificial intelligence course that introduces them to artificial neural networks, search, and fuzzy logic. Students without this background could be overwhelmed with a range of new concepts that, while they are useful in computer vision, are not specific to the field.

One final issue with *Image Processing, Analysis, & Machine Vision* is that the coverage, while broad, is uneven. Looking at the relative depth with which topics are covered, one is left with the perception that the personal research topics of the authors receive significantly more emphasis. For example, chapter 9 is a monolithic chapter on 3D vision that includes stereo, calibration, shape-from-shading, reflectance models, photometric stereo, and range-imaging. In all of the other texts these topics are divided into 2-3 separate chapters.

A second departure from other textbooks is that some methods that receive central coverage in others are de-emphasized, not mentioned, or placed in different contexts in *Image Processing, Analysis, & Machine Vision*. For example, the Hough transform is covered in the chapter on segmentation rather than within the chapter covering edge and line-finding. While line-finding is presented as an application of the Hough transform, this presentation comes after the line-finding chapter and in the middle of a discussion of segmentation. These issues are not a major drawback to the text, but an instructor should be aware of them and should structure the lectures and reading assignments appropriately.

Overall, *Image Processing, Analysis, & Machine Vision* is a good compromise between *Machine Vision* by Jain, Kasturi, and Schunck and *Computer and Robot Vision* by Haralick & Shapiro. Through strategic selection of chapters, it can be appropriate for either a senior level CS or EE undergraduate computer vision course, or an advanced graduate level course.

4.5 Machine Vision: Theory, Algorithms, Practicalities, by E. R. Davies, Academic Press, 1997.

Davies' *Machine Vision* is an interesting contrast to *Machine Vision* by Jain, Kasturi, and Schunck and *Image Processing, Analysis, & Machine Vision*. Where the previous texts have tried to give broad coverage of computer vision and provide the fundamentals in most of its subfields, Davies' text focuses on robust, general-purpose tools and algorithms. Topics like color, stereo, object recognition, and model-based vision are either mentioned in passing or left out altogether. Thus, its coverage is more limited than the other texts reviewed herein.

The topics that are covered by Davies' *Machine Vision*, however, are covered in good detail in an accessible manner. The approach is fundamentally algorithmic, with mathematics used as a tool to supplement the conceptual presentation. Because of this, the audience for this text is fairly broad, and includes students with both computer science and EE backgrounds. Certain sections of the book, for example the section detailing the Fourier transform, are included as chapter appendices and are not required in order to understand the rest of the chapter.

Davies' book is also the only one of the five texts to include a full chapter on the use of artificial neural networks [ANN] in vision tasks. Both *Image Processing, Analysis, & Machine Vision* and *Digital Image Processing* present ANNs, but as part of a larger chapter. Davies presents an overview of ANNs, and then discusses their use in both filtering and pattern recognition.

On a more specific note, chapter 2 of this book is one of the best introductions to the practical aspects of computer vision algorithms this author has seen. Chapter 2 of *Image Processing, Analysis, & Machine Vision* is similar in nature, but focuses more on the data structures. Davies touches on many of the issues that students stumble across and have problems with during their first vision course. For example, it walks the reader through the reason they have to use two images in order to correctly convolve a Gaussian filter with an image. If students read nothing else of this book, reading chapter 2 will help them avoid numerous visits to the instructor's office with questions about why their assignment won't work.

Overall, if the focus of your course is on industrial vision applications, Davies' *Machine Vision* would be an excellent choice for a textbook. If, however, your focus is more on human-computer interfaces, robotics, navigation, or computer graphics applications, using this text would require you to provide a significant amount of other reading.

4.6 A Guided Tour of Computer Vision, by Vishvjit S. Nalwa, Addison Wesley, 1993.

Nalwa's text differentiates itself from the other four by not concerning itself with implementation details. The focus of this book is on the geometrical and conceptual foundations of computer vision. Like Davies, Nalwa is concerned about and suspect of the robustness of most vision algorithms. Where Davies' solution is to focus on techniques that have been shown to work in industrial applications, Nalwa takes the opposite approach and focuses on the fundamental concepts underlying the various sub-fields.

The coverage of *A Guided Tour of Computer Vision* is fairly broad, and touches on most of the standard topics, as shown in Table 1. Like Davies, however, Nalwa steers away from object recognition as being too context dependent. Unlike Davies, the text does cover the fundamentals of object representation.

Because Nalwa does not focus on implementation details, his text does not, and probably will not, suffer as much from age as some of the others. His focus on the concepts makes the book approachable to undergraduates of all backgrounds, so long as they have a basic mathematical dexterity. However, because of the lack of implementation details an instructor using this book would have to supplement the text with other readings in order for students to undertake practical assignments. With supplementary materials, this text would be appropriate for a general undergraduate computer vision course.

Table 1: Textbook Content Comparison

Topic	Horn	Haralick & Shapiro	Sonka, Hlavac, & Boyle	Jain, Kasturi, & Schunck	Davies	Nalwa
Filtering	Y	**Y***	Y	Y	Y	N
Image Transforms	Y	S	**Y***	Y	Y	Y
Binary Image Processing	Y	**Y***	Y	Y	Y	N
Edges & Lines	Y	Y	Y	Y	Y	Y
Segmentation (thresholding)	Y	Y	Y	Y	Y	S
Segmentation (color, physics-based)	N	N	S	S	N	N
Feature Detection	Y	Y	Y	Y	**Y***	S
Optics	Y	Y	Y	Y	Y	Y
Calibration	Y	Y	Y	Y	Y	S
Stereo	Y	Y	Y	Y	S	Y
Motion	Y	Y	Y	Y	Y	Y
Texture	Y	Y	Y	Y	Y	Y
Color & Color Spaces	S	N	Y	Y	N	N
Shading	Y	Y	Y	Y	N	Y
Biological System	N	N	N	N	N	**Y***
Biologically Motivated Vision	N	N	**Y***	N	**Y***	N
Shape Representation	Y	Y	Y	Y	N	Y
Object Recognition	Y	Y	Y	Y	N	N
Line-Drawing Interpretation	S	**Y***	S	N	N	**Y***
Image Compression	N	N	**Y***	N	N	N
Image Restoration	N	N	**Y***	N	N	N

Y - covered in detail as a chapter or a significant part of one

N - not covered

S - covered somewhat, but not in detail

* - strong or unique coverage of the topic

approach to presenting the material, like *Machine Vision: Theory, Algorithms, Practicalities*, but also needs to emphasize applications such as object recognition, vision as part of human-computer interfaces, image databases, and computer graphics. These topics are more closely tied to the field of computer science and the opportunities available to CS graduates.

In conclusion, we must realize that we tend to teach as we have been taught. While this may be appropriate for students who have been taught like us, we must recognize our audience and tailor our styles and material appropriately. CS majors possess certain types of knowledge and abilities. We, as educators, can use their existing knowledge to more easily pass on the concepts and ideas of computer vision.

6 Acknowledgments

The author would like to thank Kevin Bowyer, Louise Stark, and George Stockman for organizing both this issue of IJPRAI and the 1997 Workshop on Undergraduate Education and Image Computation, held at CVPR-97. That workshop was the inspiration and motivation for this work. In addition, their comments and observations have been invaluable in making this a useful reference for computer vision educators.

7 References

[1] Castleman, K. *Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[2] Davies, E. R., *Machine Vision: Theory, Algorithms, Practicalities, 2nd Ed*., Academic Press, San Diego, CA, 1997.

[3] Gupta, A., S. Santini, and R. Jain, “In Search of Information in Visual Media”, Communications of the ACM, Vol. 40, No. 12, pp. 35-42, 1997.

[4] Haralick, R., and L. Shapiro, *Computer and Robot Vision*, Addison-Wesley, Reading, Mass., 1992.

[5] Horn, B. K. P., *Robot Vision*, MIT Press, Cambridge, 1985.

[6] Jain, R., R. Kasturi, and B. G. Schunck, *Machine Vision*, McGraw-Hill, New York, 1995.

[7] Klinker, G., *A Physical Approach to Color Image Understanding*, Ph.D. thesis, Computer Science, Carnegie Mellon University, CMU-CS-88-161.

[8] Krumm, J., *Space Frequency Shape Inference and Segmentation of 3D Surfaces*, Ph.D. Thesis, Robotics Institute, Carnegie Mellon University, CMU-RI-TR-93-32, 1993.

[9] D. Lee, R. Barber, and W. Niblack, “Indexing for Complex Queries on a Query-By-Content Image Database,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 142-146, 1994.

[10] Levoy, M., and P. Hanrahan, “Light Field Rendering,” in *Proc. SIGGRAPH `96*, pp. 31-42, 1996.

[11] Maimone, M., “Modeling foreshortening in stereo vision using local spatial frequency,” *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 1995, pp. 519-24 vol.1.

[12] Maxwell, B. A. and S. A. Shafer, “Physics-Based Segmentation of Complex Objects Using Multiple Hypotheses of Image Formation,” in *Computer Vision and Image Understanding*, Vol. 65, No. 2, February, pp. 269-295, 1997.

[13] Murase, H. and S. K. Nayar, “Visual Learning and Recognition of 3-D Objects from Appearance,” *Int'l Journal of Computer Vision*, vol. 14, pp. 5-24, 1995.

[14] Nalwa, V., *A Guided Tour of Computer Vision*, Addison-Wesley, Reading, Mass., 1993.

[15] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision, 2nd edition*, PWS Publishers, to appear spring 1998. (1st edition published by Chapman and Hall, London, 1993, sold out and out of print).

[16] Swain, M., D. Ballard, “Color Indexing,” *Int'l Journal of Computer Vision*, vol. 7, no. 1, pp. 11-32, 1991.

[17] Turk, M. and A. Pentland, “Face Recognition Using Eigenfaces,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586-591, 1991.

[18] Umbaugh, S., *Computer Vision and Image Processing: A Practical Approach Using CVIPtools*, Prentice-Hall PTR, Upper Saddle River, NJ, 1998.