scan techniques, and Fourier transformation head up the list. The development of the use of the Fourier transformation is most interesting and is applied to pattern recognition by Fourier optics. Speech recognition is introduced, and the differences in techniques required are fully discussed. The author does an excellent job in this chapter of relating known concepts to pattern recognition.

CHAPTER X—LEARNING

In chapter X the author discusses unsupervised learning systems to develop the classification characteristics. The whole chapter is devoted to various techniques of teaching the pattern classifier how best to achieve classification.

This book never discusses what Fukunaga [5] considers to be the most important part of pattern recognition, error determination. This would make it more difficult to use this book as text in a higher level course. One possible other shortcoming is the total lack of problems in the book. All in all, for persons interested in achieving an introductory background in pattern recognition, this book is unsurpassed.

REFERENCES


ROBERT C. GILLESPIE

West Virginia Tech.

Montgomery, W. Va.


This book will become a lethal weapon against skepticism toward the possibility of presenting programming as a precise discipline based on clear and simple principles. It treats what many regard as advanced topics in programming in so straightforward a manner, that there is little doubt that the "advanced topics" are in fact elementary notions. The material is presented simultaneously at many levels; the college sophomore will enjoy a hearty repast of solid conceptual and practical aspects of algorithm design, and the college professor or industrial professional will enjoy many morsels from the philosophy of programming to practical advice on the best uses of each language construction. The material is presented in a fast-moving, unadorned style, reminiscent in places of sketchy lecture notes; indeed, fully fourteen chapters appear in the first 124 pages. Every concept is backed by an example; I counted approximately 45 sample algorithms or programs in the text and about 25 more in the Exercises. The sample programs are masterpieces of clarity, exemplifying how invariant assertions can be embedded in comments to make understanding and verifying the programs as simple as possible. Wirth leaves little doubt as to the power of this technique, not merely by discussing it but by doing it.

There are fifteen chapters. Chapter 1, a one-page introduction, tides the reader over to Chapter 2: the author evidently does not like long introductions. Accordingly, in Chapter 2 he gets down immediately to business, introducing the basic concept of an algorithm as a sequence of statements defining a pattern of behavior on the values of variables. Chapter 3 describes the basic components of a computing machine. Chapter 4 outlines the basic components of a computer system. Chapter 5 simultaneously presents some simple programs and shows how to include in them invariant assertions from which the correctness of the program can be deduced. Chapter 6 shows a simple condition which guarantees the termination of all iterations, one so simple as to have escaped the attention of many a programmer. Chapter 7 introduces basic programming language concepts—expressions, compound statements, conditional statements, repetitive statements, and selective statements. It also restates the rules of inference about program verification used in the previous two chapters. Chapter 8 is a masterful exposition, axiomatically, of the four data types Boolean, integer, character, and real. The difficult aspect of the proposition of a semantic is concisely treated. Chapter 9 discusses some very simple programs based on recurrence relations, showing the power and utility of the while statement. Unfortunately, all the examples are drawn from number theory or numerical analysis and do not illustrate the power of the while statement for other applications. Chapter 10 is a discussion of the data structure array and its use. Chapter 11 is a discussion of the data structure array and its use. Chapter 12 tells about procedures and functions; in ten short pages the author covers the motivation for procedures, scope of names, parameter passing, and function-procedures. The author eliminates the possibility of a student's confusion over the hairline distinction between "begin" blocks and "procedure" blocks (as in Algol or PL/1) by eliminating "begin" blocks as a method of limiting the scope of names. He also eliminates possible confusion over function procedures' side effects by prohibiting assignments to nonlocal variables within such procedures. However, the instructor is going to have to point out to the student that he may be confused by such things when he encounters programming languages other than the one used here (Pascal). Chapter 13 is about number representations and conversions between bases. (It seemed to interrupt the progression of ideas, but I suppose every book on programming contains such a chapter.) Chapter 14 presents some text-processing examples, accomplished via array and file manipulations. Chapter 15 gives four examples of stepwise (sometimes called top-down) program development. The examples show very nicely the ease with which well-structured, nontrivial programs can be presented. The examples are: solving linear equations by Gaussian elimination, finding two solutions for the dio-phantine equation $x = y^2 + z^2$, tabulating the first n prime numbers, and generating character strings in which no two immediately adjacent substrings are equal.

I have two criticisms of this book, each concerning what was left out rather than what is included. First, the programs appearing in the text and Exercises are heavily biased toward numerical and number-theoretic problems. My crude count of the numbers of examples in various categories is shown in the table:

<table>
<thead>
<tr>
<th>Text</th>
<th>Exercises</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>TOTAL</td>
<td>45</td>
</tr>
</tbody>
</table>

(There are also 19 other Exercises which are straight exercises, proofs of simple propositions, or thought problems.) The heavy emphasis on numerical and number-theoretic problems will reduce the appeal of the book to engineering and business-oriented communities. And, except for Gaussian elimination, it is difficult to envision applications even in the scientific community of the examples developed stepwise in Chapter 15; to stress the utility of stepwise program development in practice, it seems, one should use examples of practical utility. However, the author's choice of examples is entirely consistent with his advice; he has stated in the Preface, "exercises and examples have been selected as demonstrations of generally valid problems and methods of solution."

His emphasis on numerical and number problems is also consistent with his intention that "this text is tailored for people who view a course in systematic construction of algorithms as part of their basic mathematical training."

My final criticism is that the book presents only a subset of Pascal, a language of which the author should be proud. In fact, the reader seems to have taken great pains not to mention the name Pascal: there is no mention of it in the Preface, and the only textual reference I could find was in a footnote. I can surmise only that he has deemphasized Pascal because he wants the reader to regard a specific language merely as a tool for implementing the concepts of systematic programming—he specifically wants to...
avoid the accusation he has written a book for the purpose of promoting Pascal. There seems to be little real danger of this accusation's actually being leveled, for the book so clearly emphasizes the concepts dealt with more profoundly and in a broader sense than preceding chapters, would considerably enhance the value of future editions of this book (especially among engineers and businessmen), and would introduce the student to systematic uses of data structures. And, more important, this would emphasize Pascal's data-structuring capabilities, which exempt it from the criticism so often directed at Algol 60 (viz., Algol's utility outside of scientific computing is marginal since it cannot handle structured data).

My remarks regarding two things I should like to have seen included in the book should in no way be interpreted negatively. They are matters of personal taste. Seldom does one find a book so well written that it is possible to think of only two ways of improving it! You will enjoy this book. Read it.

PETER J. DENNING
Dept. of Computer Sciences
Purdue University
West Lafayette, Ind. 47907


This book covers most of the major topics in pattern recognition at a level appropriate to a new student of the field. Therefore it can be very useful as a textbook either for an introductory graduate, or an advanced undergraduate course. The authors have deliberately played down the mathematical formalism (no long or rigorous proofs) and assumed only that background which students in such courses are expected to have. They supply numerous problems and quite a few illustrative and instructive examples. One attractive feature is the critical evaluation of many of the techniques discussed which their own experience enables them to do. I enjoyed reading statements like the one on p. 179 about the literature on linear discriminant functions. Finally each chapter is followed by extensive historical and bibliographical remarks.

The book is divided into two parts as its title implies. The first part consists of six chapters describing the statistical techniques used for pattern recognition: Bayes decision theory (including multivariate normal and independent binary features), maximum likelihood estimation, nonparametric techniques (Parzen windows, nearest neighborhood based estimates), linear discriminant functions (including both the separable and nonseparable cases), linear programming, potential functions, a brief review of stochastic approximation and an extensive treatment of clustering and unsupervised learning. Emphasis is placed on problems which have been of special interest in pattern recognition. Thus a detailed analysis is made of the effect of adding new features in the performance of a pattern classifier. Examples are given illustrating the multiplicity of solutions and the pitfalls of the clustering techniques. The material is well organized and cohesive.

The second part also consists of six chapters dealing with problems inherent in the processing of pictorial data. The title scene analysis does not do justice to the breadth of the coverage since only part of the last chapter deals with that subject which is traditionally referred to under the term. In contrast to the first part, the treatment here is less organized or complete and this is certainly due to the great flux (and associated controversy) of the state of the art in the field of computer processing of pictorial data. Parts which provide background methodology deal with spatial Fourier transform, perspective transformations, projective invariants, and line fitting techniques. There the coverage is quite systematic. Other parts deal with more involved topics and in particular region analysis, shape description, and syntactic description. There one could argue on many points but it is certainly to the authors credit that they even attempted to tackle these subjects in the limited space available. A reasonably complete coverage of these topics is probably outside the scope of a general introductory course. On the other hand I wish the authors had said a few more words than they do in p. 366 about the comparison of the various techniques of boundary shape description (e.g., Fourier coefficients versus the earlier discussed points of maximum curvature). It would have also been nice if the computational complexity of the various schemes had been discussed since it has often been the case that otherwise attractive methods often suffer in this respect.

There are close to 400 references and although they do not cover by any means the enormous literature on the subject they give a fair sampling of it for a beginner to get started.

Overall the book is a very worthwhile addition to the literature, especially because of its breadth of coverage.

THEODOSIOS PAVLIDIS
Dep. of Elec. Eng.
Princeton University
Princeton, N.J. 08540


This volume is one of a series of state of the art reports by Infotech. In some respects it is reminiscent of the software which is discussed within it. Its organization is a good example of top down specification via the method of successive refinement. There are other analogies too which shall be pointed out later.

The organization of the book is as follows:

How To Use This Report: This is a one-page guide to the reading of this volume. It describes the organization of the report in terms of levels. A description such as this is essential for a report of this size. Software system documentors take note.

Forward: The foreword provides, in six pages, a very broad brush discussion of the subject matter of the report plus addition commentary on the organization and use of the report. This is considered to be level 1 of the hierarchically structured report.

Analysis: This section is entitled "Software Engineering" and comprises, in just over 200 pp., a wide ranging overview of software engineering. The subject is organized into the following constituents.

1) Software design.
2) Project control.
3) Documentation.
4) Software production.
5) Correctness and testing.
6) Performance evaluation.
7) The software industry.
8) Education for software engineers.

Though its possible to quibble with this breakdown (everyone has their own list) this is a perfectly reasonable organization of the subject.

In this analysis section, other analogies to large software efforts begin to appear. While the editor's intersperse connecting paragraphs, the meat of this section consists mostly of excerpts from the following section which consists of presentations and invited papers. There is an attempt to simulate a discussion among the participants by juxtaposing these excerpts. While this technique occasionally succeeds, it as often as not does not. The analogy to software is the corresponding lack of integration so frequently found in large systems.

The analysis does succeed in conveying that the subject area is not rigorously enough defined for a common terminology to exist or, at times, for very meaningful things to be said. The essential aspects as well as the problems of software engineering are presented. Its most useful function might well lie in indicating where no adequate methodology exists. However, the disarray of software engineering vis-a-vis so-called ordinary engineering is at least in part overdone. The CS-A and Concorde airplanes and certain power generating nuclear reactors have had project difficulties not unlike large software systems. Even relatively stable fields have difficulties, e.g., witness the frequent automotive call-back campaigns.

Presentations and Invited Papers: There are eighteen presentations or invited papers, covering most of the areas described in the