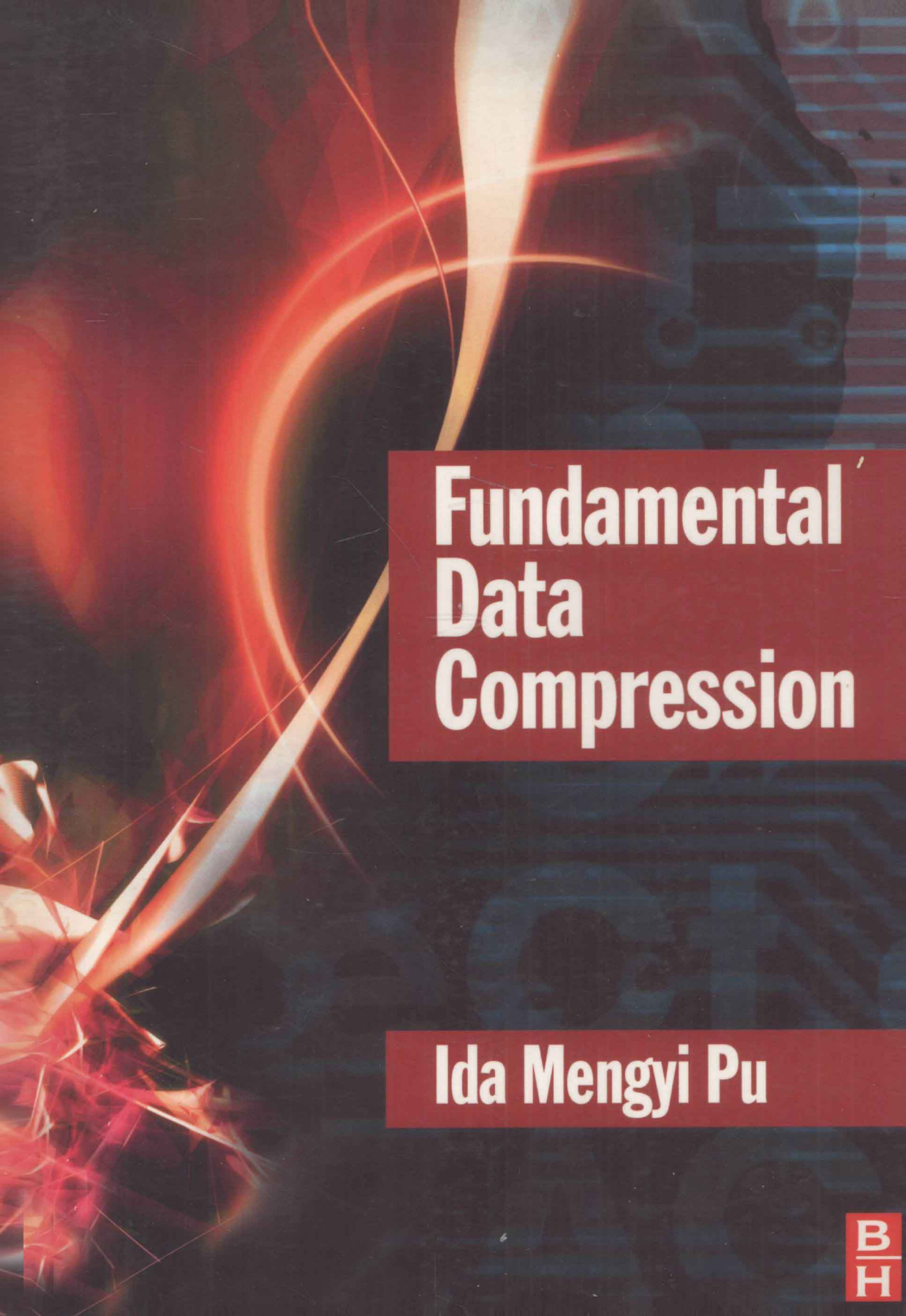
Fundamental
Data
Compression
Ida Mengyi Pu
B
H

Fundamental Data Compression

Ida Mengyi Pu

ELSEVIER

Amsterdam • Boston • Heidelberg • London • New York • Oxford
Paris • San Diego • San Francisco • Singapore • Sydney • Tokyo
Butterworth-Heinemann is an imprint of Elsevier

Butterworth-Heinemann is an imprint of Elsevier
Linacre House, Jordan Hill, Oxford OX2 8DP
30 Corporate Drive, Suite 400, Burlington, MA 01803

First published 2006

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication Data
A catalogue record for this book is available from the Library of Congress

ISBN–13: 978-0-7506-6310-6
ISBN–10: 0-7506-6310-3

For information on all Butterworth-Heinemann publications visit our web site at http://books.elsevier.com

Printed and bound in Great Britain

06 07 08 09 10 10 9 8 7 6 5 4 3 2 1

Fundamental Data Compression

To my students, family and friends

Preface

This book aims to introduce you to data compression, a fascinating research area in computer science as well as in telecommunication and computer networks of electronic engineering.

The book sets out a sequence for you to study the subject efficiently. It lays a good foundation for your professional work or advanced studies in the future. Topics include various useful compression techniques for text, audio, image, video data and international standards. I hope to show you the significance of some compression ideas and technologies, and, more importantly, to stimulate and encourage you to design your own compression algorithms.

We shall, in this text,

- discuss important issues in data compression
- study a variety of data compression techniques for compression of binary, text, sound and image data in digital forms
- learn how to design and implement compression algorithms.

We shall study some existing compression standards and compression utilities available. You will not only broaden the knowledge of compression techniques but also appreciate the mathematical impact on the evolution of the technology. You will also benefit from the development of your own transferable skills such as problem analysis and problem solving. Your skills in algorithm design and programming can also be improved by attempting the exercises, laboratory work and assessment questions at the end of each chapter.

I write this text book for students reading for their degrees as well as for anyone who is interested in data compression. Data compression is such a fruitful research area that the literature is extremely rich. However, I want to keep the contents of this book concise so the coverage and length of studies are suitable for a university single academic term (30 hours in 10 weeks) module. I want to give the reader many examples and illustrations to ease the learning process. I also want to include various exercises and implementation tasks to encourage readers' creative activities.

This book, therefore,

- looks at various topics in data compression from an algorithm designer's point of view

- is focused on the algorithmic issues which were overlooked in the past
- invites the reader to engage more in the teaching/learning process
- provides opportunities for the reader to apply the skills and knowledge from other modules, such as data structures, algorithm design, programming, software developments, internet computing, to name just a few.

Other considerations and arrangements include:

- following an order from simple to complex for sections and chapters
- covering certain theoretical foundations for data compression
- introducing not only the data compression techniques but also the ideas behind them
- focusing on algorithms and stimulating new ideas
- including exercises, laboratory problems, implementation hints, bibliography for further reading and assessment questions for each chapter.

Each data compression technique is viewed as a solution to a certain algorithmic problem in the book. The learning process can be viewed as a process of learning how to derive algorithmic solutions to various compression problems.

Bibliography

There are few textbooks on data compression for teaching purposes. However, a huge number of research papers and websites are dedicated to the subject.

To focus on the fundamentals, only a selective list is provided in the book for further reading. However, it should not be difficult for interested readers to find more detailed information on the Internet. Otherwise, the following key words can be used to start your search for various topics:

```
data compression
compression algorithm
information theory
run-length
Huffman coding
arithmetic coding
LZ77, LZ78, LZW
Burrows-Wheelers transform
```

Web page for the book

There will be an auxiliary website to the book (contact the publisher for details). It is a good idea to visit the page from time to time for updated information.

In addition to the usual teaching materials such as the texts or news, you may find our demonstration and experiment pages interesting. You may also check your understanding on certain concepts or verify your exercise results.

Get involved

The best way to learn a new subject is to get involved as much as possible. For example, you may like to share your own demonstration programme with others in the world. Check the book web page on how to contribute.

Prerequisites

This book is fairly self-contained. Appendices provide you with further mathematics background. However, there are some prerequisites including knowledge of elementary mathematics and basic algorithmics. You may find the issues in this book easier if you have reviewed certain topics in mathematics at undergraduate level, such as *sets, probability theory, basic computation on matrices* and *simple trigonometric functions* (e.g. $\sin(x)$ and $\cos(x)$, where $x \in [0, 1]$), and topics in *algorithm design*, such as *data structures* and *computational complexity.* It would be advantageous if you are fluent with one of computer programming languages.

For those who have neither the background nor the time to prepare themselves for the subject, we recommend that you follow each chapter closely since necessary mathematics or algorithmic foundations are discussed anyway.

Study methods

The highlight of each university academic year is to share the joy of my students' success in their studies. While different methods work better for different people, some methods seem to work for most people.

For example, one effective way to study compression algorithms is to trace the steps in each algorithm and attempt an example yourself. It is even better if you can follow the algorithmic ideas and try to invent your own.

Another effective way to study compression algorithms is to implement them and run experiments. Exercise and laboratory questions at the end of each chapter may direct you to various starting points for your experiments. They all together provide good help to your understanding.

Based on experience, we suggest and recommend the following practice:

1. Spend two hours on revision or exercise for every hour of study on new material.

2. Use examples to increase your understanding of new concepts, issues and problems.

3. Ask the question: 'Is there a better solution to the current problem?'

4. Use the Contents pages to comfort yourself with the scope of the subjects, and refer to the Learning Outcomes at the end of each chapter to clarify the learning tasks.

Experts have predicted that more and more people will be engaged in jobs involving certain multimedia application in the future. As more images and audio data are required to be processed, data compression techniques will continue to grow and evolve. Like any technology, what you have learnt today can become outdated tomorrow. We therefore recommend that you focus on the important principles of the subject and gain a good understanding of the issues in the field of data compression. The experience could be very useful for your future career.

Exercises, laboratory and assessment

The exercises, laboratory and assessment questions at the end of each chapter are set for you to check your understanding and to practise your programming skills.

It is useful for you to have access to a computer so you can implement the algorithmic ideas learnt from the book. There is no restriction on the computer platform nor a requirement for a specific procedural computer language. Our internal students at the University of London have gained experience in implementing various compression algorithms in *Java*, *C*, *C++*, *Python*, *Visual Basic*, *MatLab* or even *Pascal*.

Although implementation of an algorithm remains pretty much an art among university students today, you may like to follow a more systematic approach in your implementation in order to develop a 'good programming style':

1. Analyse and understand the algorithm.
2. Derive a general plan for the implementation.
3. Develop the program blocks.
4. Test or justify the correctness of the programs.
5. Comment on the limitations of the programs.

Your implementation details should be documented including a section on each of the above stages of work.

How to use this book

The book provides guidance for further development of your interests as well as further reading. However, you are not expected to read every item in the Bibliography section to enable individual topics to be studied in depth.

This book is written to invite you to get involved. The learning process requires the *input* of your own experiments and experience. Therefore, you are encouraged to, if possible, ask questions, pursue articles in research journals, browse the relative websites, attend conferences or trade shows etc., and in

general pay attention to what is happening in the computing world. Best of all, try your own experiments and invent your own algorithms.

The rest of this book is organised as follows:

Chapter 1 discusses essentials, definitions and algorithmic concepts.

Chapter 2 introduces the information theory and enhances the concepts and issues in data compression.

Chapter 3 introduces an intuitive compression method: run-length coding. This simple algorithm serves as an example of how to design a compression algorithm in a systematic way.

Chapter 4 discusses the preliminaries of data compression and reviews the main idea of Huffman coding and Shannon-Fano coding. This serves as an example to demonstrate how to apply the information theory to analyse the compression algorithm, and how to address the efficient implementation issues.

Chapter 5 introduces adaptive Huffman coding.

Chapter 6 studies issues of arithmetic coding.

Chapter 7 covers dictionary-based compression techniques.

Chapter 8 introduces prediction and transforms. This serves as a foundation for the next three chapters.

Chapter 9 discusses one-dimensional wave signals. This serves as an application of prediction and transforms in the previous chapter.

Chapter 10 discusses image data and still image compression. This serves as an application of the prediction and transform techniques on two-dimensional data.

Chapter 11 introduces video compression methods.

Appendix A highlights the milestones in the area of data compression.

Appendix B reviews the basics on matrix operations.

Appendix C covers briefly the necessary mathematics for Fourier transforms.

Appendix D provides the guidance on the pseudocode in algorithm design.

And finally,

Appendix E gives a list of notations used in the book.

Your comments

If you have any comments about this book, either general or specific, favourable or unfavourable, you are very welcome to send them to `i.pu@gold.ac.uk`.

Good luck!

I.M. Pu
London
September 2005

Acknowledgements

I would like to take the opportunity to thank the many people who helped me in preparation of the book: David Hatter for initialising this project and encouragement; the reviewers Professors Maxime Crochemore, Bill Smyth, and an anonymous professional for valuable comments; the proof reader Dr Yuji Shen for helpful comments and suggestions; the editors and publishing team Alfred Waller, Melissa Read, Hayley Salter, Deb Puleston, and Jodi Cusack, for constant support.

I would also like to thank all the researchers working on original ideas of data compression for making this research area extremely interesting and challenging.

Contents