Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science' (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

Also in this series

Iain D. Craig
Object-Oriented Programming Languages: Interpretation
978-1-84628-773-2

Max Bramer
Principles of Data Mining
978-1-84628-765-7

Hanne Riis Nielson and Flemming Nielson
Semantics with Applications: An Appetizer
978-1-84628-691-9

Michael Kifer and Scott A. Smolka
Introduction to Operating System Design and Implementation: The OSP 2 Approcah
978-1-84628-842-5

Phil Brooke and Richard Paige
Practical Distributed Processing
978-1-84628-840-1

Frank Klawonn
Computer Graphics with Java
978-1-84628-847-0

David Salomon

A Concise Introduction to Data Compression

Springer

Professor David Salomon (emeritus)
Computer Science Department
California State University
Northridge, CA 91330-8281, USA
email: david.salomon@csun.edu

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

Library of Congress Control Number: 2007939563

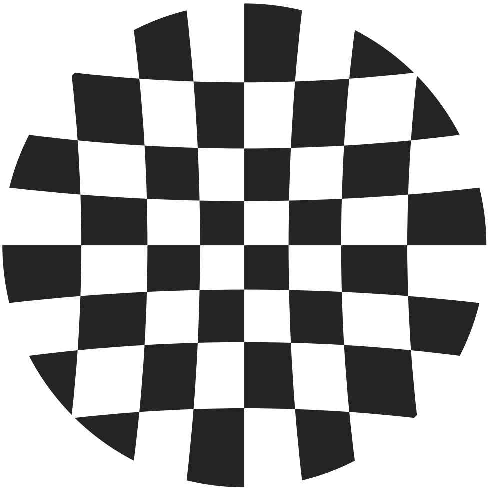
Undergraduate Topics in Computer Science ISSN 1863-7310
ISBN 978-1-84800-071-1 e-ISBN 978-1-84800-072-8

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

This book is dedicated to you, the reader!



Nothing is more impossible than to write
a book that wins every reader's approval.
—Miguel de Cervantes

Preface

It is virtually certain that a reader of this book is both a computer user and an Internet user, and thus the owner of digital data. More and more people all over the world generate, use, own, and enjoy digital data. Digital data is created (by a word processor, a digital camera, a scanner, an audio A/D converter, or other devices), it is edited on a computer, stored (either temporarily, in memory, less temporarily, on a disk, or permanently, on an optical medium), transmitted between computers (on the Internet or in a local-area network), and output (printed, watched, or played, depending on its type).

These steps often apply mathematical methods to modify the representation of the original digital data, because of three factors, time/space limitations, reliability (data robustness), and security (data privacy). These are discussed in some detail here:

The first factor is time/space limitations. It takes time to transfer even a single byte either inside the computer (between the processor and memory) or outside it over a communications channel. It also takes space to store data, and digital images, video, and audio files tend to be large. Time, as we know, is money. Space, either in memory or on our disks, doesn't come free either. More space, in terms of bigger disks and memories, is becoming available all the time, but it remains finite. Thus, decreasing the size of data files saves time, space, and money—three important resources. The process of reducing the size of a data file is popularly referred to as *data compression*, although its formal name is *source coding* (coding done at the source of the data, before it is stored or transmitted).

In addition to being a useful concept, the idea of saving space and time by compression is ingrained in us humans, as illustrated by (1) the rapid development of nanotechnology and (2) the quotation at the end of this Preface.

The second factor is reliability. We often experience noisy telephone conversations (with both cell and landline telephones) because of electrical interference. In general, any type of data, digital or analog, sent over any kind of communications channel may become corrupted as a result of channel noise. When the bits of a data file are sent over a computer bus, a telephone line, a dedicated communications line, or a satellite connection, errors may creep in and corrupt bits. Watching a high-resolution color image or a long video, we may not be able to tell when a few pixels have wrong colors, but other types of data require absolute reliability. Examples are an executable computer program, a legal text document, a medical X-ray image, and genetic information. Change one bit in the executable code of a program, and the program will not run, or worse, it may run and do the wrong thing. Change or omit one word in a contract and it may reverse its meaning. Reliability is therefore important and is achieved by means of error-control codes. The formal name of this mathematical discipline is *channel coding*, because these codes are employed when information is transmitted on a communications channel.

The third factor that affects the storage and transmission of data is security. Generally, we do not want our data transmissions to be intercepted, copied, and read on their way. Even data saved on a disk may be sensitive and should be hidden from prying eyes. This is why digital data can be encrypted with modern, strong encryption algorithms that depend on long, randomly-selected keys. Anyone who doesn't possess the key and wants access to the data may have to resort to a long, tedious process of either trying to break the encryption (by analyzing patterns found in the encrypted file) or trying every possible key. Encryption is especially important for diplomatic communications, messages that deal with money, or data sent by members of secret organizations. A close relative of data encryption is the field of data hiding (steganography). A data file `A` (a payload) that consists of bits may be hidden in a larger data file `B` (a cover) by taking advantage of "holes" in `B` that are the result of redundancies in the way data is represented in `B`.

Overview and goals

This book is devoted to the first of these factors, namely data compression. It explains why data can be compressed, it outlines the principles of the various approaches to compressing data, and it describes several compression algorithms, some of which are general, while others are designed for a specific type of data.

The goal of the book is to introduce the reader to the chief approaches, methods, and techniques that are currently employed to compress data. The main aim is to start with a clear overview of the principles behind this field, to complement this view with several examples of important compression algorithms, and to present this material to the reader in a coherent manner.

Organization and features

The book is organized in two parts, basic concepts and advanced techniques. The first part consists of the first three chapters. They discuss the basic approaches to data compression and describe a few popular techniques and methods that are commonly used to compress data. Chapter 1 introduces the reader to the important concepts of variable-length codes, prefix codes, statistical distributions, run-length encoding, dictionary compression, transforms, and quantization. Chapter 2 is devoted to the important Huffman algorithm and codes, and Chapter 3 describes some of the many dictionary-based compression methods.

The second part of this book is concerned with advanced techniques. The original and unusual technique of arithmetic coding is the topic of Chapter 4. Chapter 5 is devoted to image compression. It starts with the chief approaches to the compression of images, explains orthogonal transforms, and discusses the JPEG algorithm, perhaps the best example of the use of these transforms. The second part of this chapter is concerned with subband transforms and presents the WSQ method for fingerprint compression as an example of the application of these sophisticated transforms. Chapter 6 is devoted to the compression of audio data and in particular to the technique of linear prediction. Finally, other approaches to compression—such as the Burrows–Wheeler method, symbol ranking, and SCSU and BOCU-1—are given their due in Chapter 7.

The many exercises sprinkled throughout the text serve two purposes, they illuminate subtle points that may seem insignificant to readers and encourage readers to test their knowledge by performing computations and obtaining numerical results.

Other aids to learning are a prelude at the beginning of each chapter and various intermezzi where interesting topics, related to the main theme, are examined. In addition, a short summary and self-assessment exercises follow each chapter. The glossary at the end of the book is comprehensive, and the index is detailed, to allow a reader to easily locate all the points in the text where a given topic, subject, or term appear.

Other features that liven up the text are puzzles (indicated by 🧩, with answers at the end of the book) and various boxes with quotations or with biographical information on relevant persons.

Target audience

This book was written with undergraduate students in mind as the chief readership. In general, however, it is aimed at those who have a basic knowledge of computer science; who know something about programming and data structures; who feel comfortable with terms such as *bit*, *mega*, *ASCII*, *file*, *I/O*, and *binary search*; and who want to know how data is compressed. The necessary mathematical background is minimal and is limited to logarithms, matrices, polynomials, calculus, and the concept of probability. This book is not intended as a guide to software implementors and has few programs.

The book's web site, with an errata list, BibTeX information, and auxiliary material, is part of the author's web site, located at `http://www.ecs.csun.edu/~dsalomon/`. Any errors found, comments, and suggestions should be directed to `dsalomon@csun.edu`.

Acknowlegments

I would like to thank Giovanni Motta and John Motil for their help and encouragement. Giovanni also contributed to the text and pointed out numerous errors.

In addition, my editors at Springer Verlag, Wayne Wheeler and Catherine Brett, deserve much praise. They went over the manuscript, made numerous suggestions and improvements, and contributed much to the final appearance of the book.

Lakeside, California David Salomon
August 2007

> To see a world in a grain of sand
> And a heaven in a wild flower,
> Hold infinity in the palm of your hand
> And eternity in an hour.
>
> —William Blake, *Auguries of Innocence*

Contents

Part II: Advanced Techniques 121

> The content of most textbooks is perishable, but the
> tools of self-directness serve one well over time.
>
> —Albert Bandura