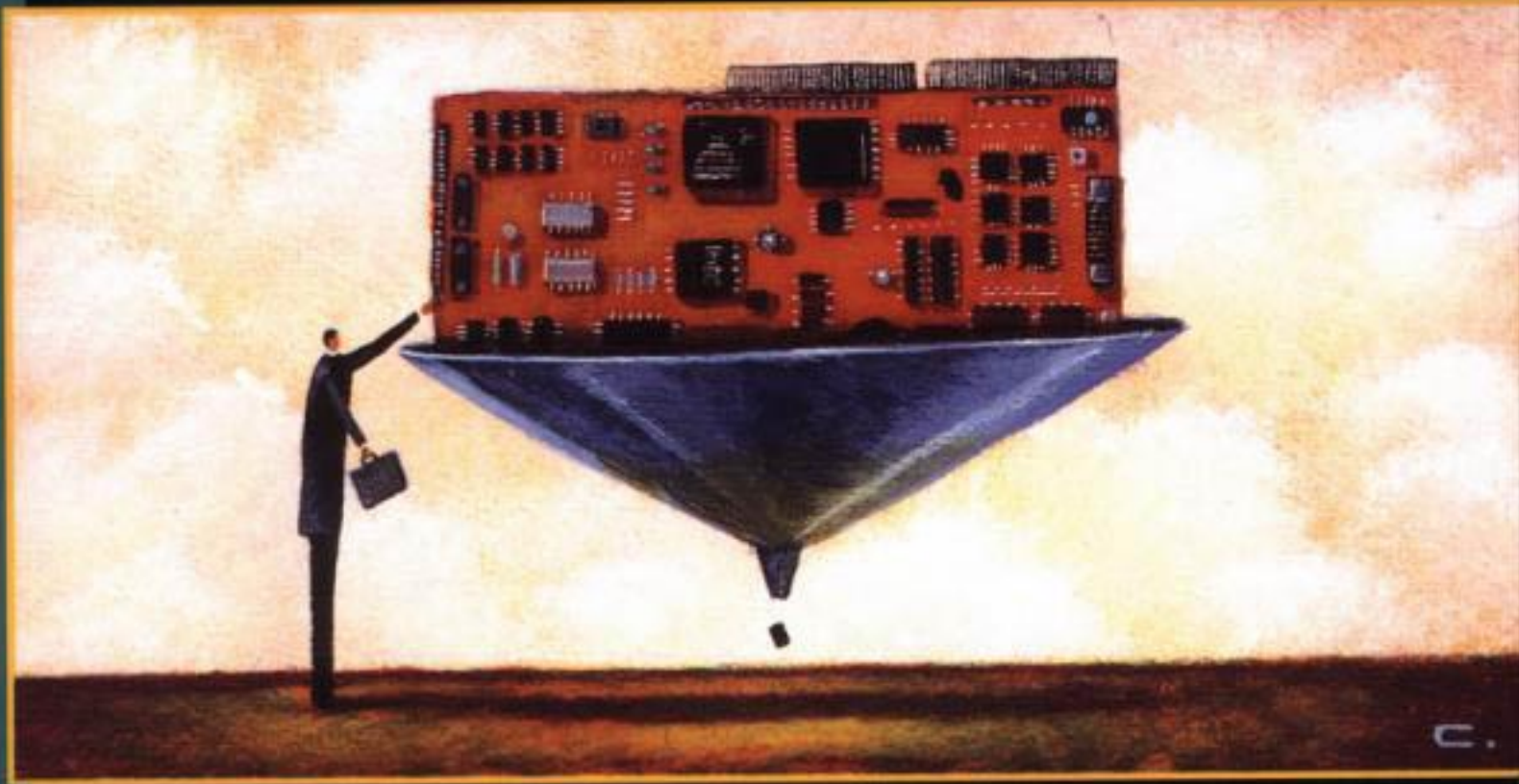


The Data Compression Book

SECOND EDITION



"The best all-around book on the subject."

—Andrew Shulman, *Dr. Dobb's Journal*

"The book hits its target audience right between the eyes."

—David D. Forster, *Dr. Dobbs' Journal*

"One of my favorite books on applied computer technology is *The Data Compression Book*."

—Jeff Proise, *PC Magazine*

Mark Nelson

Afterword

When writing about data compression, I am haunted by the idea that many of the techniques discussed in this book have been patented by their inventors or others. The knowledge that a data compression algorithm can effectively be taken out of the hands of programmers through the use of so-called “intellectual property” law seems contrary to the basic principles that led me and many others into this profession.

I have yet to see any evidence that applying patents to software advances that art or protects the rights of inventors. Several companies continue to collect royalties on patents long after their inventors have moved onto bigger and better things with other companies. Have the patent-holders done anything notable other than collect royalties? Have they advanced the art of computer science?

Making a software product into a commercial success requires innovation, good design, high-quality documentation, and listening to customers. These are things that nobody can steal from you. On the other hand, a mountain of patents can't keep you from letting these things slip away through inattention or complacency. This lesson seems to be lost on those who traffic in intellectual property “portfolios.”

What can you do? First, don't patent your own work, and discourage your peers from doing so. Work on improving your products, not erecting legal obstacles to competition. Secondly, lobby for change. This means change within your company, those you do business with, and most importantly, within the federal government. Write to your congressman and your senator. Write to the ACM. Write to the House Subcommittee on Intellectual Property. And finally, you can join me by becoming a member of the League for Programming Freedom. Write for more information:

League For Programming Freedom
1 Kendall Square #143
P.O. Box 9171
Cambridge, MA 02139

I concluded, we kinotropists must be numbered among Britain's most adept programmers of Enginery of any sort, and virtually all advances on the compression of data have originated as kinotropic applications.

At this point, he interrupted again, asking if I had indeed said "the compression of data," and was I familiar with the term "algorithmic compression"? I assured him I was.

The Difference Engine

William Gibson and Bruce Sterling

Why This Book Is For You

If you want to learn how programs like PKZIP and LHarc work, this book is for you. The compression techniques used in these programs are described in detail, accompanied by working code. After reading this book, even the novice C programmer will be able to write a complete compression/archiving program that can be ported to virtually any operating system or hardware platform.

If you want to include data compression in other programs you write, this book will become an invaluable tool. It contains dozens of working programs with C code that can easily be added to your applications. In-depth discussions of various compression methods will help you make intelligent decisions when creating programs that use data compression.

If you want to learn why lossy compression of graphics is the key factor in enabling the multimedia revolution, you need this book. DCT-based compression like that used by the JPEG algorithm is described in detail. The cutting edge technology of fractal compression is explained in useful terms, instead of the purely theoretical. Working programs let you experiment with these fascinating new technologies.

The Data Compression Book provides you with a comprehensive reference to this important field. No other book available has the detailed description of compression algorithms or working C implementations for those algorithms. If you are planning to work in this field, The Data Compression Book is indispensable.

Afterword
Why This Book Is For You

Chapter 1—Introduction to Data Compression

The Audience
Why C?
Which C?
 Issues in Writing Portable C
Keeping Score
The Structure

Chapter 2—The Data-Compression Lexicon, with a History

The Two Kingdoms
Data Compression = Modeling + Coding
The Dawn Age
Coding
 An Improvement
Modeling
 Statistical Modeling
 Dictionary Schemes
Ziv and Lempel
 LZ77
 LZ78
Lossy Compression
Programs to Know

Chapter 3—The Dawn Age: Minimum Redundancy Coding

The Shannon-Fano Algorithm
The Huffman Algorithm
Huffman in C
 BITIO.C
A Reminder about Prototypes
MAIN-C.C AND MAIN-E.C
 MAIN-C.C
 ERRHAND.C
Into the Huffman Code
 Counting the Symbols
 Saving the Counts
 Building the Tree
 Using the Tree
The Compression Code
Putting It All Together
 Performance

Chapter 4—A Significant Improvement: Adaptive Huffman Coding

Adaptive Coding
Updating the Huffman Tree
 What Swanning Does

- The Algorithm**
- An Enhancement**
- The Escape Code**
- The Overflow Problem**
- A Rescaling Bonus**
- The Code**
 - Initialization of the Array**
 - The Compress Main Program**
 - The Expand Main Program**
 - Encoding the Symbol**
 - Updating the Tree**
 - Decoding the Symbol**
- The Code**

Chapter 5—Huffman One Better: Arithmetic Coding

- Difficulties**
- Arithmetic Coding: A Step Forward**
 - Practical Matters**
 - A Complication**
 - Decoding**
 - Where’s the Beef?**
- The Code**
 - The Compression Program**
 - The Expansion Program**
 - Initializing the Model**
 - Reading the Model**
 - Initializing the Encoder**
 - The Encoding Process**
 - Flushing the Encoder**
 - The Decoding Process**
- Summary**
 - Code**

Chapter 6—Statistical Modeling

- Higher-Order Modeling**
- Finite Context Modeling**
- Adaptive Modeling**
 - A Simple Example**
 - Using the Escape Code as a Fallback**
 - Improvements**
- Highest-Order Modeling**
 - Updating the Model**
 - Escape Probabilities**
 - Scoreboarding**
 - Data Structures**
 - The Finishing Touches: Tables –1 and –2**
 - Model Flushing**
 - Implementation**
- Conclusions**
 - Enhancement**
- ARITH-N Listing**

Chapter 7—Dictionary-Based Compression

- An Example**

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.