

The background of the book cover features a dark blue field with vertical columns of binary code (0s and 1s) in a lighter blue font. Overlaid on this is a network diagram consisting of numerous small, semi-transparent orange spheres connected by thin, light blue lines, creating a web-like structure.

# DISTRIBUTED SOURCE CODING

THEORY AND PRACTICE

SHUANG WANG  
YONG FANG  
SAMUEL CHENG



WILEY

# Distributed Source Coding

Theory and Practice

*Shuang Wang, Yong Fang, and Samuel Cheng*

WILEY

This edition first published 2017

© 2017 John Wiley & Sons Ltd

Registered office John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at [www.wiley.com](http://www.wiley.com).

The right of Shuang Wang, Yong Fang, and Samuel Cheng to be identified as the authors of this work has been asserted in accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act 1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The publisher is not associated with any product or vendor mentioned in this book.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. It is sold on the understanding that the publisher is not engaged in rendering professional services and neither the publisher nor the author shall be liable for damages arising herefrom. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

*Library of Congress Cataloging-in-Publication data applied for*

ISBN: 9780470688991

A catalogue record for this book is available from the British Library.

Cover design by Wiley

Cover image: Top numbers image: a\_Taiga/Gettyimages

Bottom molecular image: KTSDESIGN/SCIENCE PHOTO LIBRARY/Gettyimages

Set in 10/12pt WarnockPro by SPi Global, Chennai, India

Printed and bound in Malaysia by Vivar Printing Sdn Bhd

10 9 8 7 6 5 4 3 2 1

## Preface

This book intends to provide fundamental knowledge for engineers and computer scientists to get into the topic of distributed source coding, and is targeted at senior undergraduate or first-year graduate students. It should also be accessible for engineers with a basic knowledge of calculus and applied probability. We have included short chapters on information theory, channel coding, and approximate inference in the appendix. The goal is to make the content as self-contained as possible.

Excluding the appendix, the book is divided into three parts. First-time readers who would like to get into applications quickly should work through Chapters 1, 2, 3, 5, and 7, and then go straight to Part III, the application chapters. Chapters 4 and 6 are more advanced and may require more mathematical maturity from readers.

Distributed source coding is an active research area and the authors have not been able to cover every aspect of it. We have included materials that are geared more toward our own research interests. However, we have striven to include the most interesting applications in this book.

## Acknowledgment

Book writing is a long endeavor. It is especially so in this case. This book project grew out of several distributed source coding tutorials presented in the US, Taiwan and Switzerland around 2009 by Dr. Vladimir Stankovic, Dr. Lina Stankovic, and the third author. Many materials in this book reflect the immense influence of these tutorials. Drs. Stankovic have been very kind in offering extensive unconditional help and allowed us to include many of their ideas and even their own words in this book. For this we wish to express our most sincere and deepest gratitude.

The third author would also like to express special thanks to his Master and PhD advisor, Dr. Zixiang Xiong, who taught him not just distributed source coding, but also everything relating to research and academia.

Over the years, the authors have received help from many colleagues, friends, and students, including Angelos Liveris, Costas Georgiades, Yang Yang, Zixin Liu, Zongmin Liu, Qian Xu, Shengjie Zhao, Nikoloas Deligiannis, Xiaoqian Jiang, Adrian Munteanu, Peter Schelkens, Andrei Sechelea, Feng Chen, and Lijuan Cui. In particular, we would like to offer special thanks to Liyan Chen and Xiaomin Wang for generating some of the figures. We could not possibly include everyone in the list above, and we would like to ask for forgiveness for any omission. Finally, the authors would like to acknowledge support from the National Science Foundation, the National Institutes of Health (NHGRI K99HG008175), and the National Nature Science Foundation of China.

## About the Companion Website

Don't forget to visit the companion website for this book:



[www.wiley.com/go/cheng/dsctheoryandpractice](http://www.wiley.com/go/cheng/dsctheoryandpractice)

There you will find valuable material designed to enhance your learning, including:

- Example codes

Scan this QR code to visit the companion website



## Contents

**Preface** *xiii*

**Acknowledgment** *xv*

**About the Companion Website** *xvii*

### **1 Introduction** *1*

- 1.1 What is Distributed Source Coding? *2*
- 1.2 Historical Overview and Background *2*
- 1.3 Potential and Applications *3*
- 1.4 Outline *4*

## **Part I Theory of Distributed Source Coding** *7*

### **2 Lossless Compression of Correlated Sources** *9*

- 2.1 Slepian–Wolf Coding *10*
  - 2.1.1 Proof of the SW Theorem *15*
    - Achievability of the SW Theorem *16*
    - Converse of the SW Theorem *19*
- 2.2 Asymmetric and Symmetric SW Coding *21*
- 2.3 SW Coding of Multiple Sources *22*

### **3 Wyner–Ziv Coding Theory** *25*

- 3.1 Forward Proof of WZ Coding *27*
- 3.2 Converse Proof of WZ Coding *29*
- 3.3 Examples *30*
  - 3.3.1 Doubly Symmetric Binary Source *30*
    - Problem Setup *30*
    - A Proposed Scheme *31*
    - Verify the Optimality of the Proposed Scheme *32*
  - 3.3.2 Quadratic Gaussian Source *35*

|          |   |           |
|----------|---|-----------|
|          | Problem Setup   | 35        |
|          | Proposed Scheme   | 36        |
|          | Verify the Optimality of the Proposed Scheme                                    | 37        |
| 3.4      | Rate Loss of the WZ Problem   | 38        |
|          | Binary Source Case  | 39        |
|          | Rate loss of General Cases  | 39        |
| <b>4</b> | <b>Lossy Distributed Source Coding</b>  | <b>41</b> |
| 4.1      | Berger–Tung Inner Bound   | 42        |
| 4.1.1    | Berger–Tung Scheme  | 42        |
|          | Codebook Preparation  | 42        |
|          | Encoding  | 42        |
|          | Decoding  | 43        |
| 4.1.2    | Distortion Analysis   | 43        |
|          | $Pr(\mathcal{E}_1) \rightarrow 0$   | 44        |
|          | $Pr(\mathcal{E}_1^C \cap \mathcal{E}_2) \rightarrow 0$                          | 44        |
|          | $Pr(\mathcal{E}_1^C \cap \mathcal{E}_2^C \cap \mathcal{E}_3) \rightarrow 0$     | 44        |
|          | $Pr(\mathcal{E}_4) \rightarrow 0$   | 44        |
|          | $Pr(\mathcal{E}_5), Pr(\mathcal{E}_6) \rightarrow 0$                            | 44        |
| 4.2      | Indirect Multiterminal Source Coding  | 45        |
| 4.2.1    | Quadratic Gaussian CEO Problem with Two Encoders                                | 45        |
|          | Forward Proof of Quadratic Gaussian CEO Problem with Two Terminals              | 46        |
|          | Converse Proof of Quadratic Gaussian CEO Problem with Two Terminals             | 48        |
| 4.3      | Direct Multiterminal Source Coding  | 54        |
| 4.3.1    | Forward Proof of Gaussian Multiterminal Source Coding Problem with Two Sources  | 55        |
|          | Case 1: $(1 - D_2) \leq \rho^2(1 - D_1)$  | 59        |
|          | Case 2: $(1 - D_2) > \rho^2(1 - D_1)$   | 62        |
| 4.3.2    | Converse Proof of Gaussian Multiterminal Source Coding Problem with Two Sources | 63        |
|          | Bounds for $R_1$ and $R_2$  | 64        |
|          | Collaborative Lower Bound   | 66        |
|          | $\mu$ -sum Bound  | 67        |

## Part II Implementation 75

|          |  |           |
|----------|--|-----------|
| <b>5</b> | <b>Slepian–Wolf Code Designs Based on Channel Coding</b> | <b>77</b> |
| 5.1      | Asymmetric SW Coding                                     | 77        |
| 5.1.1    | Binning Idea   | 78        |
| 5.1.2    | Syndrome-based Approach                                  | 79        |



|          |   |           |
|----------|---|-----------|
|          | Hamming Binning                                   | 80        |
|          | SW Encoding                                       | 80        |
|          | SW Decoding                                       | 80        |
|          | LDPC-based SW Coding                              | 81        |
| 5.1.3    | Parity-based Approach                             | 82        |
| 5.1.4    | Syndrome-based Versus Parity-based Approach       | 84        |
| 5.2      | Non-asymmetric SW Coding                          | 85        |
| 5.2.1    | Generalized Syndrome-based Approach               | 86        |
| 5.2.2    | Implementation using IRA Codes                    | 88        |
| 5.3      | Adaptive Slepian–Wolf Coding                      | 90        |
| 5.3.1    | Particle-based Belief Propagation for SW Coding   | 91        |
| 5.4      | Latest Developments and Trends                    | 93        |
| <b>6</b> | <b>Distributed Arithmetic Coding</b>              | <b>97</b> |
| 6.1      | Arithmetic Coding                                 | 97        |
| 6.2      | Distributed Arithmetic Coding                     | 101       |
| 6.3      | Definition of the DAC Spectrum                    | 103       |
| 6.3.1    | Motivations                                       | 103       |
| 6.3.2    | Initial DAC Spectrum                              | 104       |
| 6.3.3    | Depth- $i$ DAC Spectrum                           | 105       |
| 6.3.4    | Some Simple Properties of the DAC Spectrum        | 107       |
| 6.4      | Formulation of the Initial DAC Spectrum           | 107       |
| 6.5      | Explicit Form of the Initial DAC Spectrum         | 110       |
| 6.6      | Evolution of the DAC Spectrum                     | 113       |
| 6.7      | Numerical Calculation of the DAC Spectrum         | 116       |
| 6.7.1    | Numerical Calculation of the Initial DAC Spectrum | 117       |
| 6.7.2    | Numerical Estimation of DAC Spectrum Evolution    | 118       |
| 6.8      | Analyses on DAC Codes with Spectrum               | 120       |
| 6.8.1    | Definition of DAC Codes                           | 121       |
| 6.8.2    | Codebook Cardinality                              | 122       |
| 6.8.3    | Codebook Index Distribution                       | 123       |
| 6.8.4    | Rate Loss   | 123       |
| 6.8.5    | Decoder Complexity                                | 124       |
| 6.8.6    | Decoding Error Probability                        | 126       |
| 6.9      | Improved Binary DAC Codec                         | 130       |
| 6.9.1    | Permutated BDAC Codec                             | 130       |
|          | Principle   | 130       |
|          | Proof of SW Limit Achievability                   | 131       |
| 6.9.2    | BDAC Decoder with Weighted Branching              | 132       |
| 6.10     | Implementation of the Improved BDAC Codec         | 134       |
| 6.10.1   | Encoder   | 134       |
|          | Principle   | 134       |
|          | Implementation                                    | 135       |

|          |   |            |
|----------|---|------------|
| 6.10.2   | Decoder   | 135        |
|          | Principle                                       | 135        |
|          | Implementation                                  | 136        |
| 6.11     | Experimental Results                            | 138        |
|          | Effect of Segment Size on Permutation Technique | 139        |
|          | Effect of Surviving-Path Number on WB Technique | 139        |
|          | Comparison with LDPC Codes                      | 139        |
|          | Application of PBDAC to Nonuniform Sources      | 140        |
| 6.12     | Conclusion                                      | 141        |
| <b>7</b> | <b>Wyner–Ziv Code Design</b>                    | <b>143</b> |
| 7.1      | Vector Quantization                             | 143        |
| 7.2      | Lattice Theory                                  | 146        |
| 7.2.1    | What is a Lattice?                              | 146        |
|          | Examples  | 146        |
|          | Dual Lattice                                    | 147        |
|          | Integral Lattice                                | 147        |
|          | Lattice Quantization                            | 148        |
| 7.2.2    | What is a Good Lattice?                         | 149        |
|          | Packing Efficiency                              | 149        |
|          | Covering Efficiency                             | 150        |
|          | Normalized Second Moment                        | 150        |
|          | Kissing Number                                  | 150        |
|          | Some Good Lattices                              | 151        |
| 7.3      | Nested Lattice Quantization                     | 151        |
|          | Encoding/decoding                               | 152        |
|          | Coset Binning                                   | 152        |
|          | Quantization Loss and Binning Loss              | 153        |
|          | SW Coded NLQ                                    | 154        |
| 7.3.1    | Trellis Coded Quantization                      | 154        |
| 7.3.2    | Principle of TCQ                                | 155        |
|          | Generation of Codebooks                         | 156        |
|          | Generation of Trellis from Convolutional Codes  | 156        |
|          | Mapping of Trellis Branches onto Sub-codebooks  | 157        |
|          | Quantization                                    | 157        |
|          | Example   | 158        |
| 7.4      | WZ Coding Based on TCQ and LDPC Codes           | 159        |
| 7.4.1    | Statistics of TCQ Indices                       | 159        |
| 7.4.2    | LLR of Trellis Bits                             | 162        |
| 7.4.3    | LLR of Codeword Bits                            | 163        |
| 7.4.4    | Minimum MSE Estimation                          | 163        |
| 7.4.5    | Rate Allocation of Bit-planes                   | 164        |
| 7.4.6    | Experimental Results                            | 166        |

**Part III Applications 167**

|          |  |
|----------|--|
| <b>8</b> | <b>Wyner–Ziv Video Coding 169</b>  |
| 8.1      | Basic Principle 169  |
| 8.2      | Benefits of WZ Video Coding 170  |
| 8.3      | Key Components of WZ Video Decoding 171  |
| 8.3.1    | Side-information Preparation 171   |
|          | Bidirectional Motion Compensation 172  |
| 8.3.2    | Correlation Modeling 173   |
|          | Exploiting Spatial Redundancy 174  |
| 8.3.3    | Rate Controller 175  |
| 8.4      | Other Notable Features of Miscellaneous WZ Video Coders 175                            |
| <b>9</b> | <b>Correlation Estimation in DVC 177</b>   |
| 9.1      | Background to Correlation Parameter Estimation in DVC 177                              |
| 9.1.1    | Correlation Model in WZ Video Coding 177   |
| 9.1.2    | Offline Correlation Estimation 178   |
|          | Pixel Domain Offline Correlation Estimation 178  |
|          | Transform Domain Offline Correlation Estimation 180                                    |
| 9.1.3    | Online Correlation Estimation 181  |
|          | Pixel Domain Online Correlation Estimation 182   |
|          | Transform Domain Online Correlation Estimation 184                                     |
| 9.2      | Recap of Belief Propagation and Particle Filter Algorithms 185                         |
| 9.2.1    | Belief Propagation Algorithm 185   |
| 9.2.2    | Particle Filtering 186   |
| 9.3      | Correlation Estimation in DVC with Particle Filtering 187                              |
| 9.3.1    | Factor Graph Construction 187  |
| 9.3.2    | Correlation Estimation in DVC with Particle Filtering 190                              |
| 9.3.3    | Experimental Results 192   |
| 9.3.4    | Conclusion 197   |
| 9.4      | Low Complexity Correlation Estimation using Expectation Propagation 199                |
| 9.4.1    | System Architecture 199  |
| 9.4.2    | Factor Graph Construction 199  |
|          | Joint Bit-plane SW Coding (Region II) 200  |
|          | Correlation Parameter Tracking (Region I) 201  |
| 9.4.3    | Message Passing on the Constructed Factor Graph 202                                    |
|          | Expectation Propagation 203  |
| 9.4.4    | Posterior Approximation of the Correlation Parameter using Expectation Propagation 204 |
|          | Moment Matching 205  |
| 9.4.5    | Experimental Results 206   |
| 9.4.6    | Conclusion 211   |

|           |   |            |
|-----------|---|------------|
| <b>10</b> | <b>DSC for Solar Image Compression</b>  | <b>213</b> |
| 10.1      | Background  | 213        |
| 10.2      | Related Work  | 215        |
| 10.3      | Distributed Multi-view Image Coding   | 217        |
| 10.4      | Adaptive Joint Bit-plane WZ Decoding of Multi-view Images with Disparity Estimation | 217        |
| 10.4.1    | Joint Bit-plane WZ Decoding   | 217        |
| 10.4.2    | Joint Bit-plane WZ Decoding with Disparity Estimation                               | 219        |
| 10.4.3    | Joint Bit-plane WZ Decoding with Correlation Estimation                             | 220        |
| 10.5      | Results and Discussion  | 221        |
| 10.6      | Summary   | 224        |
| <br>      |   |            |
| <b>11</b> | <b>Secure Distributed Image Coding</b>  | <b>225</b> |
| 11.1      | Background  | 225        |
| 11.2      | System Architecture   | 227        |
| 11.2.1    | Compression of Encrypted Data   | 228        |
| 11.2.2    | Joint Decompression and Decryption Design   | 230        |
| 11.3      | Practical Implementation Issues   | 233        |
| 11.4      | Experimental Results  | 233        |
| 11.4.1    | Experiment Setup  | 234        |
| 11.4.2    | Security and Privacy Protection   | 235        |
| 11.4.3    | Compression Performance   | 236        |
| 11.5      | Discussion  | 239        |
| <br>      |   |            |
| <b>12</b> | <b>Secure Biometric Authentication Using DSC</b>                                    | <b>241</b> |
| 12.1      | Background  | 241        |
| 12.2      | Related Work  | 243        |
| 12.3      | System Architecture   | 245        |
| 12.3.1    | Feature Extraction  | 246        |
| 12.3.2    | Feature Pre-encryption  | 248        |
| 12.3.3    | SeDSC Encrypter/decrypter   | 248        |
| 12.3.4    | Privacy-preserving Authentication   | 249        |
| 12.4      | SeDSC Encrypter Design  | 249        |
| 12.4.1    | Non-asymmetric SW Codes with Code Partitioning                                      | 250        |
| 12.4.2    | Implementation of SeDSC Encrypter using IRA Codes                                   | 251        |
| 12.5      | SeDSC Decrypter Design  | 252        |
| 12.6      | Experiments   | 256        |
| 12.6.1    | Dataset and Experimental Setup  | 256        |
| 12.6.2    | Feature Length Selection  | 257        |
| 12.6.3    | Authentication Accuracy   | 257        |
|           | Authentication Performances on Small Feature Length (i.e., $N = 100$ )              | 257        |
|           | Performances on Large Feature Lengths (i.e., $N \geq 300$ )                         | 258        |

|          |   |            |
|----------|---|------------|
| 12.6.4   | Privacy and Security                              | 259        |
| 12.6.5   | Complexity Analysis                               | 261        |
| 12.7     | Discussion  | 261        |
| <b>A</b> | <b>Basic Information Theory</b>                   | <b>263</b> |
| A.1      | Information Measures                              | 263        |
| A.1.1    | Entropy   | 263        |
| A.1.2    | Relative Entropy                                  | 267        |
| A.1.3    | Mutual Information                                | 268        |
| A.1.4    | Entropy Rate                                      | 269        |
| A.2      | Independence and Mutual Information               | 270        |
| A.3      | Venn Diagram Interpretation                       | 273        |
| A.4      | Convexity and Jensen's Inequality                 | 274        |
| A.5      | Differential Entropy                              | 277        |
| A.5.1    | Gaussian Random Variables                         | 278        |
| A.5.2    | Entropy Power Inequality                          | 278        |
| A.6      | Typicality  | 279        |
| A.6.1    | Jointly Typical Sequences                         | 282        |
| A.7      | Packing Lemmas and Covering Lemmas                | 284        |
| A.8      | Shannon's Source Coding Theorem                   | 286        |
| A.9      | Lossy Source Coding—Rate-distortion Theorem       | 289        |
| A.9.1    | Rate-distortion Problem with Side Information     | 291        |
| <b>B</b> | <b>Background on Channel Coding</b>               | <b>293</b> |
| B.1      | Linear Block Codes                                | 294        |
| B.1.1    | Syndrome Decoding of Block Codes                  | 295        |
| B.1.2    | Hamming Codes, Packing Bound, and Perfect Codes   | 295        |
| B.2      | Convolutional Codes                               | 297        |
| B.2.1    | Viterbi Decoding Algorithm                        | 298        |
| B.3      | Shannon's Channel Coding Theorem                  | 301        |
| B.3.1    | Achievability Proof of the Channel Coding Theorem | 303        |
| B.3.2    | Converse Proof of Channel Coding Theorem          | 305        |
| B.4      | Low-density Parity-check Codes                    | 306        |
| B.4.1    | A Quick Summary of LDPC Codes                     | 306        |
| B.4.2    | Belief Propagation Algorithm                      | 307        |
| B.4.3    | LDPC Decoding using BP                            | 312        |
| B.4.4    | IRA Codes   | 314        |
| <b>C</b> | <b>Approximate Inference</b>                      | <b>319</b> |
| C.1      | Stochastic Approximation                          | 319        |
| C.1.1    | Importance Sampling Methods                       | 320        |
| C.1.2    | Markov Chain Monte Carlo                          | 321        |
|          | Markov Chains                                     | 321        |

|          |   |            |
|----------|---|------------|
|          | Markov Chain Monte Carlo                              | 321        |
| C.2      | Deterministic Approximation                           | 322        |
| C.2.1    | Preliminaries   | 322        |
|          | Exponential Family                                    | 322        |
|          | Kullback–Leibler Divergence                           | 323        |
|          | Assumed-density Filtering                             | 324        |
| C.2.2    | Expectation Propagation                               | 325        |
|          | Relationship with BP                                  | 326        |
| C.2.3    | Relationship with Other Variational Inference Methods | 328        |
| <b>D</b> | <b>Multivariate Gaussian Distribution</b>             | <b>331</b> |
| D.1      | Introduction  | 331        |
| D.2      | Probability Density Function                          | 331        |
| D.3      | Marginalization                                       | 332        |
| D.4      | Conditioning  | 333        |
| D.5      | Product of Gaussian pdfs                              | 334        |
| D.6      | Division of Gaussian pdfs                             | 337        |
| D.7      | Mixture of Gaussians                                  | 337        |
| D.7.1    | Reduce the Number of Components in Gaussian Mixtures  | 338        |
|          | Which Components to Merge?                            | 340        |
|          | How to Merge Components?                              | 341        |
| D.8      | Summary   | 342        |
|          | Appendix: Matrix Equations                            | 343        |
|          | <b>Bibliography</b>                                   | <b>345</b> |
|          | <b>Index</b>  | <b>357</b> |

## 1

## Introduction

Imagine a dense sensor network consisting of many tiny sensors deployed for information gathering. Readings from neighboring sensors will often be highly correlated. This correlation can be exploited to significantly reduce the amount of information that each sensor needs to send to the base station, thus reducing power consumption and prolonging the life of the nodes and the network. The obvious way of exploiting correlation is to enable neighboring sensors to exchange data with one another. However, communication among sensors is usually undesirable as it increases the complexity of the sensors, which in turn leads to additional cost and power consumption. How then is it possible to avoid information exchange among sensors but still be able to exploit the statistical dependency of the readings in different sensor nodes? The solution lies in *distributed source coding*.

Distributed source coding (DSC) enables correlation to be exploited efficiently without the need for communications among sensors. Moreover, in some specific cases it does not incur any loss compared to the case when the sensors communicate.

DSC has been receiving significant attention since the beginning of the 21st century from academics and industrial researchers in different fields of electrical and computer engineering, and mathematical and computer science. Indeed, special sessions and tutorials dedicated to DSC are given at major communications, signal processing, multimedia, and computer engineering conferences. This is no wonder as DSC has potential applications ranging from wireless sensor networks, ad-hoc networks, surveillance networks, to robust low-complexity video coding, stereo/multiview video coding, high-definition television, and hyper-spectral and multi-spectral imaging. This book is intended to act as a guidebook for engineers and researchers to grasp the basic concepts quickly in order to understand and contribute to this exciting field and apply the emerging applications.

## 1.1 What is Distributed Source Coding?

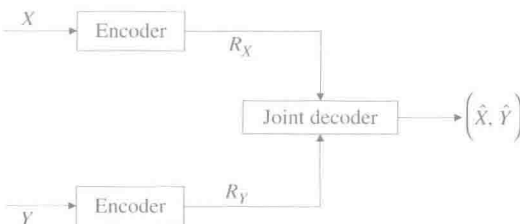
DSC deals with the source coding or compression of correlated sources. The adjective *distributed* stresses that the compression occurs in a distributed or noncentralized fashion. We can, for example, assume that the sources to be compressed are distributed across different nodes in a network. The task is to compress these sources and communicate compressed streams to a decoder for joint decompression. The basis of DSC is that the compressions take place *independently*, that is, the nodes do not exchange their information, whereas decompression is *joint*.

This is illustrated in Figure 1.1 for the simple case of two nodes. Each node has access only to its source, and does not have information about sources present at other nodes. Therefore, “distributed” in this context refers to separate compression at each node. Note that if decompression were also separate for each of the sources, then the problem would boil down to multiple conventional compressions. Throughout the book, distributed compression will always refer to separate encoding and joint decoding, whereas joint compression will refer to joint encoding and joint decoding, if not stated otherwise.

The search of achievable rates of DSC is a major information-theoretical problem and lies in the framework of network information theory, a branch of information theory that tries to find the compression and communication limits of a network of nodes. In the case of discrete sources and perfect reconstruction at the decoder, DSC extends Shannon’s Source Coding Theorem, in information theory, from the point-to-point to multipoint scenario. This is referred to as lossless DCS. When we allow for some distortion in reconstruction, the DSC problem becomes a rate-distortion problem and is referred to as lossy DSC.

## 1.2 Historical Overview and Background

DSC started as an information-theoretical problem in the seminal 1973 paper of Slepian and Wolf [1]. Slepian and Wolf considered lossless separate compression of two discrete sources, and showed that roughly speaking there is no



**Figure 1.1** DSC concept with two separate encoders who do not talk to each other and one joint decoder.  $X$  and  $Y$  are discrete, correlated sources;  $R_X$  and  $R_Y$  are compression rates.



performance loss compared to joint compression as long as joint decomposition is performed.

This remarkable result triggered significant information-theoretical research resulting in solutions – in the form of achievable rate regions – for more complicated lossless setups. In 1976, Wyner and Ziv [2] considered a lossy version, with a distortion constraint, of a special case of the Slepian–Wolf (SW) problem, where one source is available at the decoder as side information. Wyner and Ziv showed that for a particular correlation where source and side information are jointly Gaussian, there is no performance loss due to the absence of side information at the encoder. The lossy case of the generalized SW setup, known as multiterminal (MT) source coding, was introduced by Berger and Tung in 1977 [3, 4].

A possible realization of DSC via the use of conventional linear channel codes to approach the SW bound was known as early as 1973, but due to the lack of any potential application of DSC, work on code designs, that is, how to code the sources to approach given bounds, started only at the end of the last century. The first practical design was reported in 1999 [5], followed by many improved solutions. One key insight of these designs is that conventional channel coding can be used for compression. Indeed, correlation between the sources is seen as a virtual communication channel, and as long as this virtual channel can be modeled by some standard communication channel, for example Gaussian, channel codes can be effectively employed. Capacity-approaching designs [6] based on quantization followed by advanced channel coding, for example with turbo codes [7] and low-density parity-check (LDPC) codes [8], come very close to the bounds for two jointly Gaussian sources.

### 1.3 Potential and Applications

The launch of wireless sensor networks (WSNs) ignited practical DSC considerations in the early years of this century since WSNs naturally call for distributed processing. Closely located sensors are expected to have correlated measurements; thus in theory the DSC setup fulfills the requirement of power-efficient compression for distributed sensor networks. However, many practical problems remain to be solved before DSC is used in mainstream commercial networks. The challenges include the complex correlation structure of real signals, nonGaussian sources, mandatory long codeword lengths, and the complexity of current designs.

Though WSN triggered renewed interest in DSC, another application has emerged: low-complexity video, where the DSC paradigm is used to avoid computationally expensive temporal prediction loop in video encoding. Indeed, loosely speaking, a conventional video encoder needs to find the best matching block to the current one by examining all possible candidates, for