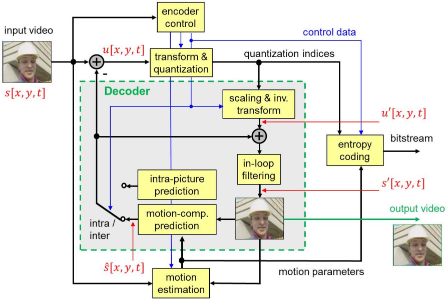
Source Coding and Compression

Heiko Schwarz



Contact:

Dr.-Ing. Heiko Schwarz

heiko.schwarz@hhi.fraunhofer.de

Outline

Introduction

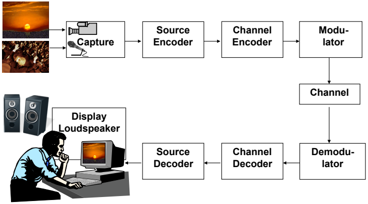
Part I: Source Coding Fundamentals

- Review: Probability, Random Variables and Random Processes
- Lossless Source Coding
- Rate-Distortion Theory
- Quantization
- Predictive Coding
- Transform Coding

Part II: Application in Image and Video Coding

- Still Image Coding / Intra-Picture Coding
- Hybrid Video Coding (From MPEG-2 Video to H.265/HEVC)

Introduction



Motivation for Source Coding

- **Source coding** or **compression** is required for efficient **transmission** or **storage**, leading to one or both of the following benefits:
  - Transmit more data given throughput (channel capacity or storage space)
  - Use less throughput given data
- Typically, source coding or compression are considered enabling technologies, i.e., technologies that enable an application
- Examples for source coding applications:
  - gzip, compress, winzip, ...
  - Mobile voice, audio, and video transmission
  - Internet voice, audio, and video transmission
  - Digital television
  - MP3 and portable video players (iPod, ...)
  - Digital Versatile Discs (DVDs) and Blu-Ray Discs

Practical Source Coding Problems

- **File compression** (text file, office document, program code, ...)
  - Example: Example 80 MByte down to 20 MByte (20%)
- **Audio compression**
  - Stereo with sampling frequency of 44.1 kHz
  - Each sample being represented with 16 bits

  $\Longrightarrow$ Raw data rate: $44.1 \times 16 \times 2 = 1.41$ Mbit/s

  $\Longrightarrow$ Typical data rate after compression: 64 kbit/s (4.5%)
- **Image compression**
  - Original picture size: 3000×2000 samples (6 MegaPixel)
  - 3 color components (red, green, blue) and 1 byte (8 bit) per sample

  $\Longrightarrow$ Raw file size: $3000 \times 2000 \times 3 = 18$ MByte

  $\Longrightarrow$ Typical compressed file size: 1 MByte (5.6%)
- **Video compression**
  - Picture size of 1920×1080 pixels and frame rate of 50 Hz
  - Each sample being digitized with 8 bit
  - 3 color components (red, green, blue)

  $\Longrightarrow$ Raw data rate: $1920 \times 1080 \times 8 \times 50 \times 3 = 2.49$ Gbit/s

  $\Longrightarrow$ Typical compressed data rate: 12 Mbit/s (0.5%)

Source Coding in Practice

- Source coding often **enables applications**:
  - Digital television (DVB-T)
  - Internet video streaming (YouTube)
- Source coding **makes applications economically feasible**
  - Distribution of digital images
  - High definition television (HDTV) over IPTV
- Many applications use source coding techniques
  - Software is often distributed in compressed form
  - Audio data are typically compressed (MP3, AAC)
    - Mobile audio players (IPod,...) and mobile phones
    - Audio download (ITunes) and streaming services (Internet radio)
  - Digital images are typically compressed (JPEG)
    - Compression is often done in camera
    - Picture found on web sites are compressed
  - Digital video data are typically compressed (MPEG-2, H.264/AVC)
    - Output of video cameras, optical discs
    - Video streaming (Youtube, Internet TV)
  - About 70% of the bits in the Internet are compressed video data

Pulse-Code Modulation

Analog-to-Digital Conversion: Pulse-Code Modulation

- Pulse-code modulation (PCM) is based on following principles
  - Sampling (obeying SHANNON-NYQUIST sampling theorem)
  - Quantizing sample values

- Sampling theorem asserts that a time-continuous signal $s(t)$ that contains only frequencies less than Ω Hz, can be recovered from a sequence of its sample values using

$$s(t) = \sum_{n=-\infty}^{\infty} s(t_n)\, \psi(t - t_n) \tag{1}$$

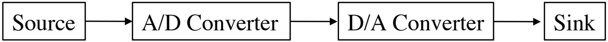
  where $s(t_n)$ is value of nth sampling instant $t_n = \frac{n}{2\Omega}$ and $\psi(\cdot)$ is given as

$$\psi(t) = \frac{\sin(2\pi\Omega t)}{2\pi\Omega t} \tag{2}$$

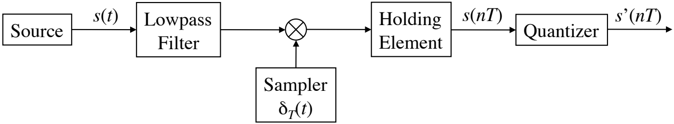
- The signal values $s(t_n)$ can be quantized allowing only an approximate reconstruction of $s(t)$

Analog-to-Digital Conversion: Overview

- Analog-to-digital and digital-to-analog conversion

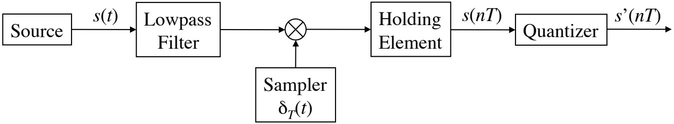


- Source and analog-to-digital converter



- Analog-to-digital converter turns analog signal into a discrete signal
  - **Analog signal**: Continuous-time and continuous-amplitude signal
  - **Discrete signal**: Discrete-time and discrete-amplitude signal

Analog-to-Digital Conversion



- Sample and hold operator turns continuous-time into discrete-time signal
- Low-pass filter ensures that signal is band-limited
- Quantizer turns continuous-amplitude signal into discrete-amplitude signal
  - A simple method is to quantize signal $s(nT)$ by mapping it to $K = 2^k$ possible amplitude values
  - A simple quantization rule is

$$s'(nT) = \lfloor s(nT) \times 2^k + 0.5 \rfloor / 2^k \tag{3}$$

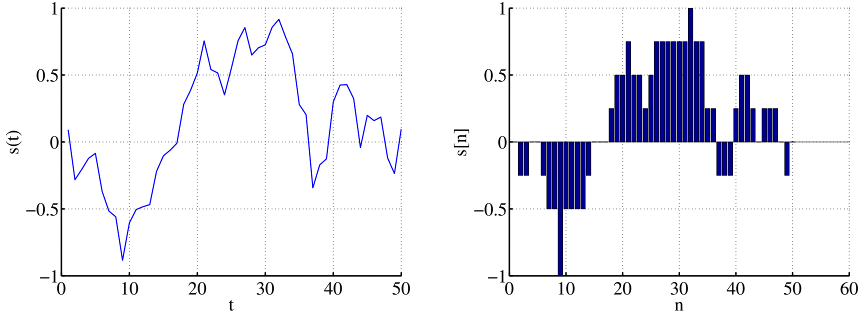
- We use the notation for the discrete signal $s[n]$ as an abbreviation for $s'(nT)$ with T being the sampling interval
- Digital values $s[n]$ are in practice numbers that are stored in a computer

Why Analog-to-Digital Conversion?

- Required for processing data with a computer
- All compression methods discussed here are computer programs:
  - Encoder: Mapping of $s[n]$ into a bit stream b
  - Decoder: Mapping of the bit stream b into the discrete decoded signal $s'[n]$
- Although we will also discuss compression of analog signals in theory, in practice all algorithms will assume discrete versions of these analog signals that are very close approximation of these analog signals

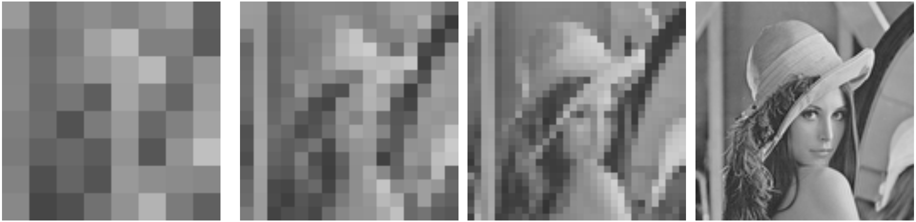
One-Dimensional Signal Example

- Speech and audio signals are typically one-dimensional temporal signals
- Discrete signal below is temporally sampled and its amplitude is represented using $k = 3$ bits, i.e., $K = 8$ different values
- Note: Reconstruction value of -0.75 is not present in example, allowing us to represent this signal with $K = 8$ instead of $K = 9$ reconstruction values



Two-Dimensional Signal Example

- Pictures are two-dimensional spatial signals
- Videos are three-dimensional spatio-temporal signals
- Below sampling of picture **Lena** with different spatial sampling rates
  - 8×8, 16×16, 32×32, and 128×128 samples (from left to right)
  - Each sample is represented with $n = 8$ bits
  - Each square represents average of luminance values it covers



Two-Dimensional Signal Example

- Below quantization of picture **Lena** with different bits/sample
  - $k = 1, 2, 4,$ and 8 bits/sample (from left to right)
  - The spatial sampling rate is fixed to 128×128




Three-Dimensional Signal Examples

- Below, format, sampling rate and sampling method for different video signals yield corresponding PCM data rates

Picture format	Luma signal	Chroma signal	Sampling	Frames/s	Data rate
Common Intermediate Format (CIF)	352×288 (352×240)	$2 \times 176 \times 144$ $(2 \times 176 \times 120)$	progressive 8 bit	25 (30)	
ITU-R BT.601 Format ("Standard Television")	720×576 (720×480)	$2 \times 360 \times 576$ $(2 \times 360 \times 480)$	interlaced 8 bit	25 (30)	
ITU-R BT.709: 720p ("High Definition TV")	1280×720	$2 \times 640 \times 720$	progressive 8 bit	50 (60)	
ITU-R BT.709: 1080i ("Full HDTV")	1920×1080	$2 \times 960 \times 1080$	interlaced 8 bit	25 (30)	
ITU-R BT.2020: UHD-1 ("Ultra HDTV 4k")	3840×2160	$2 \times 1920 \times 1080$	progressive 10 bit	50 (60)	
ITU-R BT.2020: UHD-2 ("Ultra HDTV 8k")	7680×4320	$2 \times 3840 \times 2160$	progressive 12 bit	50 (60)	

Basic Communication Problem

- The **basic communication problem** may be posed as

  > **Conveying source data with highest fidelity possible**
  > **within an available bit rate**

  or, equivalently, as

  > **Conveying source data using lowest bit rate possible**
  > **while maintaining a specified reproduction fidelity**

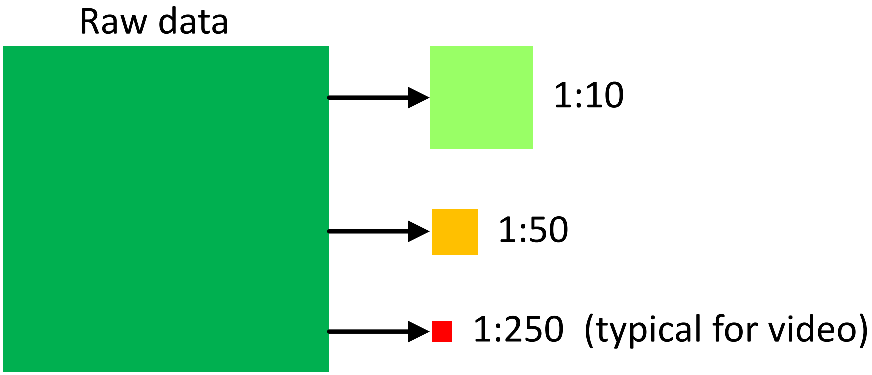
- In either case, a fundamental trade-off is made between bit rate and fidelity
- The ability of a source coding system to make this trade-off well is called its **coding efficiency** or **rate-distortion performance**, and the coding system itself is referred to as a **source codec**
- **Source codec**: a system comprising a **source co**der and a source **dec**oder

Example: JPEG (1:10 Compression)

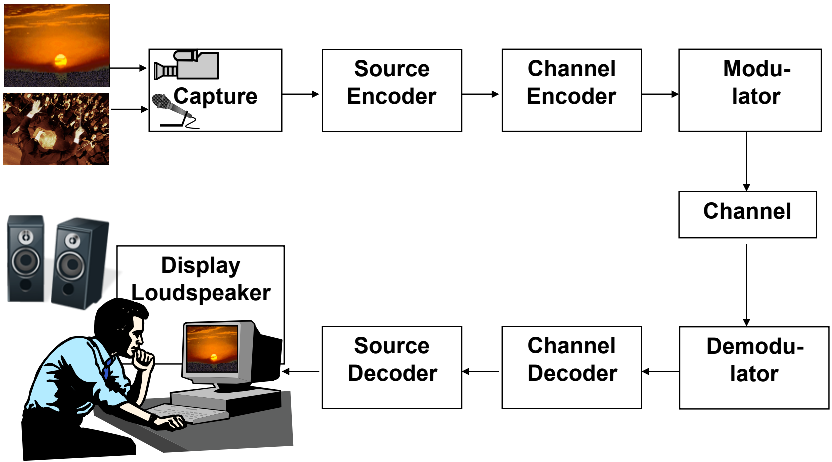
Example: JPEG (1:50 Compression)

Example: H.265/HEVC (1:50 Compression)

Geometrical Interpretation



Transmission System



Practical Communication Problem

- Source codecs are primarily characterized in terms of:
  - **Throughput of the channel**, a characteristic influenced by
    - transmission channel bit rate and
    - amount of protocol and error-correction coding overhead incurred by transmission system
  - **Distortion of the decoded signal**, which is primarily induced by
    - source encoder and
    - by channel errors introduced in path to source decoder
- The following additional constraints must also be considered
  - **Delay (start-up latency and end-to-end delay)** include
    - processing delay, buffering,
    - structural delays of source and channel codecs, and
    - speed at which data are conveyed through transmission channel
  - **Complexity (computation, memory capacity, memory access)** of
    - source codec,
    - protocol stacks, and network

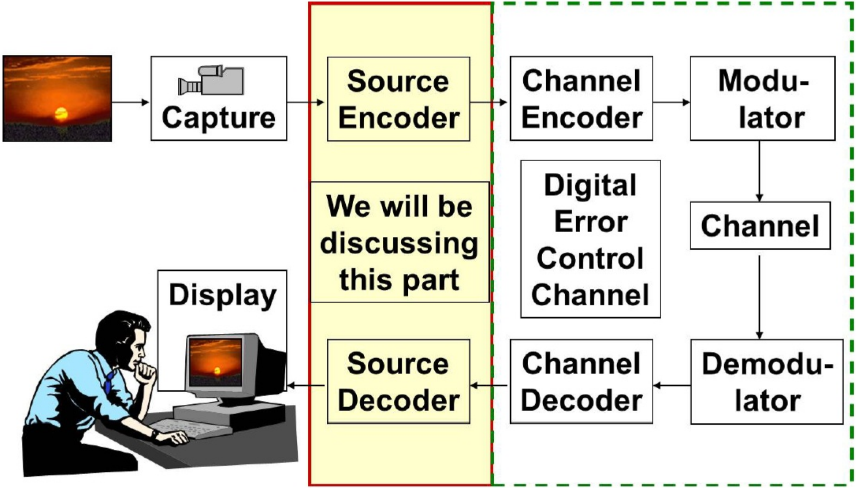
Formulation of the Practical Communication Problem

- The practical source coding design problem is posed as follows:

Given a maximum allowed delay and a maximum allowed complexity, achieve an optimal trade-off between bit rate and distortion for the transmission problem in the targeted applications.

- Here, we will concentrate on source codec only
  - Delay is only evaluated for source codec
  - Complexity is also only assessed for the algorithm used in source codec

Scope of This Course



Transmission Channels and Optical Storage Media

- Fixed transmission lines:
  - ISDN line: 64 kbit/s
  - ADSL: 6 Mbit/s
  - VDSL: 25 Mbit/s or 50 Mbit/s
- Mobile networks:
  - GSM: 15 kbit/s
  - EDGE: 474 kbit/s (max)
  - HSDPA: 7.2 Mbit/s (peak)
  - LTE: 300 Mbit/s (peak)
- Broadcast channels
  - DVB-T: 13 Mbit/s (16QAM)
  - DVB-S: 38 Mbit/s (QPSK)
  - DVB-C: 38 Mbit/s (64QAM)
- Optical storage media
  - Compact Disc (CD): 650 MByte with 1.41 Mbit/s (12 cm)
  - Digital Versatile Dics (DVD): 4.7 GByte with 10.5 Mbit/s (DVD-5-SS-SL)
  - Blu-Ray Disc (BRD): 50 GByte with 36 Mbit/s (12 cm, DS-DL)

Types of Compression

- **Lossless coding**:
  - Uses redundancy reduction as the only principle and is therefore reversible
  - Also referred to as noiseless or invertible coding or data compaction
  - Well known use for this type of compression for data is Lempel-Ziv coding (gzip) and for picture and video signals JPEG-LS is well known
- **Lossy coding**:
  - Uses redundancy reduction and irrelevancy reduction and is therefore not reversible
  - It is the primary coding type in compression for speech, audio, picture, and video signals
  - The practically relevant bit rate reduction that is achievable through lossy compression is typically more than an order of magnitude larger than with lossless compression
  - Well known examples are for audio coding are the MPEG-1 Layer 3 (mp3), for still picture coding JPEG, and for video coding H.264/AVC

Distortion Measures

- The use of lossy compression requires the ability to measure distortion
- Often, the distortion that a human perceives in coded content is a very difficult quantity to measure, as the characteristics of human perception are complex
- Perceptual models are far more advanced for speech and audio codecs than for picture or video codecs
- In speech and audio coding,
  - Perceptual models are heavily used to guide encoding decisions
  - Listening tests are used to determine subjective quality of coding results
- In picture and video coding,
  - Perceptual models have limited use to guide encoding decisions (mainly focusing on properties of the human visual system)
  - Viewing tests are used to determine subjective quality of coding results
- This lecture: Use of objective distortion measures such as MSE and SNR

Mean Squared Error (MSE)

- **Speech and audio:** (N: duration in samples)

$$u[n] = s'[n] - s[n] \tag{4}$$

$$\text{MSE} = \frac{1}{N} \sum_{n=0}^{N-1} u^2[n] \tag{5}$$

- **Pictures:** (X: picture height, Y: picture width):

$$u[x,y] = s'[x,y] - s[x,y] \tag{6}$$

$$\text{MSE} = \frac{1}{X \cdot Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} u^2[x,y] \tag{7}$$

- **Videos:** (N: number of pictures, MSE_n: MSE of picture n):

$$\text{MSE} = \frac{1}{N} \sum_{n=0}^{N-1} \text{MSE}_n \tag{8}$$

Signal-to-Noise Ratio

- **Speech:**

$$\text{SNR} = 10 \cdot \log_{10}\left(\frac{\sigma^2}{\sigma_u^2}\right) \tag{9}$$

$$\text{with} \quad \sigma^2 = \frac{1}{N}\sum_{n=0}^{N-1}(s[n]-\mu_s)^2 \;\text{ and }\; \mu_s = \frac{1}{N}\sum_{n=0}^{N-1} s[n] \tag{10}$$

$$\sigma_u^2 = \frac{1}{N}\sum_{n=0}^{N-1}(u[n]-\mu_u)^2 \;\text{ and }\; \mu_u = \frac{1}{N}\sum_{n=0}^{N-1} u[n] \tag{11}$$

- **Pictures:** (k: bit depth of samples)

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{(2^k-1)^2}{\text{MSE}}\right) \tag{12}$$

- **Videos:** (N: number of pictures, PSNR_n: PSNR of picture n)

$$\text{PSNR} = \frac{1}{N}\sum_{n=0}^{N-1} \text{PSNR}_n \tag{13}$$

Recommended Literature

Source Coding

- T. M. Cover and J. A. Thomas, “Elements of Information Theory,” John Wiley & Sons, New York, 1991.
- Gersho, A. and Gray, R. M. (1992), “Vector Quantization and Signal Compression,” Kluwer Academic Publishers, Boston, Dordrecht, London.
- Jayant, N. S. and Noll, P. (1994), “Digital Coding of Waveforms,” Prentice-Hall, Englewood Cliffs, NJ, USA.
- Wiegand, T. and Schwarz, H. (2010). Source Coding: Part I of Fundamentals of Source and Video Coding, Foundations and Trends in Signal Processing, vol. 4, no. 1-2. (http://iphome.hhi.de/wiegand/assets/pdfs/VBpart1.pdf)

Image and Video Coding

- W. Pennebaker and J. Mitchell, “JPEG Still Image Data Compression Standard,” Van Nostrand Reinhold, New York, 1993.
- D. S. Taubman and M. W. Marcellin, “JPEG 2000 – Image Compression Fundamentals, Standards, and Practice,” Kluwer Academic Publishers, 2002.
- Y. Wang, J. Ostermann, Y.-Q. Zhang, “Video Processing and Communications,” Prentice-Hall, 2002.
- J.-R. Ohm, “Multimedia Communication Technology. Representation, Transmission and Identification of Multimedia Signals,” Springer, Heidelberg/Berlin, 2004.

Organization

Lecture:	Tuesday 10:30-12:00 & 12:15-13:45 Room 1.16
Lecturer:	Dr.-Ing. Heiko Schwarz Head, Image & Video Coding Group Image Processing Department Fraunhofer Heinrich Hertz Institute heiko.schwarz@hhi.fraumhofer.de http://iphome.hhi.de/schwarz
Course weights:	Quizzes: 20% Project: 20% Midterm exam: 25% Final exam: 35%

Copies of slides and solutions of exercises can be downloaded at:

http://iphome.hhi.de/schwarz/GUC-SourceCoding.htm