Image and Video Coding: Introduction

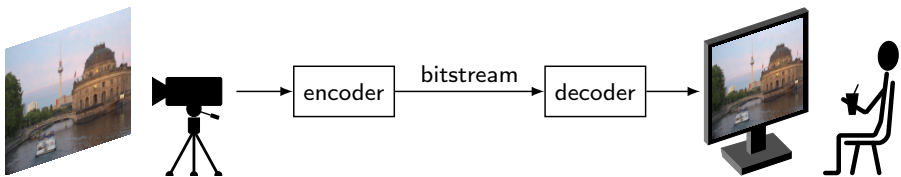
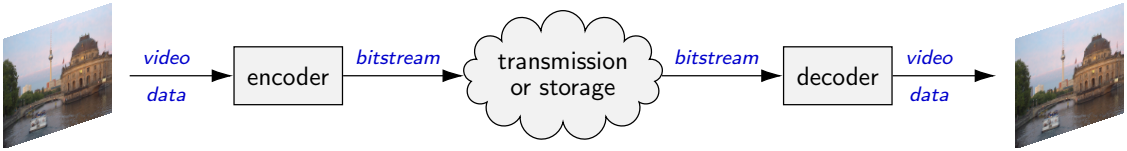


Image and Video Coding



Main Goal of Image and Video Coding

- Efficient transmission or storage of images and videos

➡ Reduce the bit rate for a given amount of video data

Image and Video Coding: Enabling Technology

- Enables new applications or makes them economically feasible
  - ➡ Distribution and storage of images and video
  - ➡ Digital television
  - ➡ Video streaming
  - ➡ Video conferencing

Important Image and Video Coding Standards of ITU-T and ISO/IEC

JPEG (1992) [ISO/IEC 10918-1 | ITU-T Rec. T.81]

➜ Storage and distribution of digital images

MPEG-2 Video (1995) [ITU-T Rec. H.262 | ISO/IEC 13818-2]

➜ Standard definition (SD): Storage (DVD-Video) and digital television broadcast (DVB-T)

H.264 | AVC : Advanced Video Coding (2003) [ITU-T Rec. H.264 | ISO/IEC 14496-10]

➜ High definition (HD): Storage (Blu-ray) and digital television broadcast (DVB-S)

➜ Video streaming

H.265 | HEVC : High Efficiency Video Coding (2013) [ITU-T Rec. H.265 | ISO/IEC 23008-2]

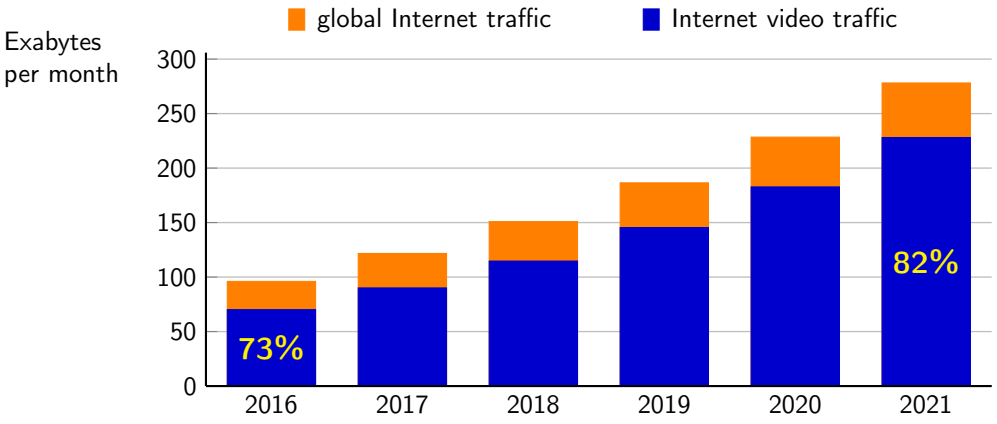
➜ Ultra-high definition (UHD) video storage (4k Blu-ray), Image storage (BPG, HEIF)

➜ Broadcast (HD: DVB-T2, UHD: DVB-S2), UHD video streaming

H.266 | VVC : Versatile Video Coding (2020) [ITU-T Rec. H.266 | ISO/IEC 23090-3]

➜ ...

Estimated Global Internet and Video Traffic



[Cisco: "The Zettabyte Era: Trends and Analysis", 2017]

➜ **Still Need Better Video Coding**

Single-Component Image

- Matrix of integer samples

$$s[x, y] \quad \text{with} \quad \begin{aligned} x &= 0, 1, \cdots, W-1 \\ y &= 0, 1, \cdots, H-1 \end{aligned}$$

- Each sample can take values in a given range

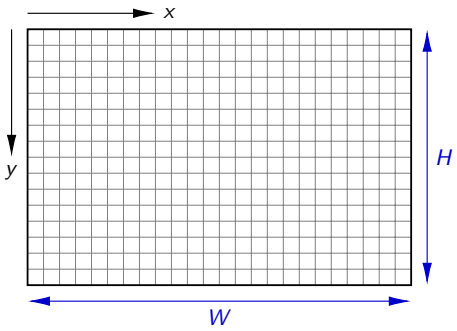
$$s[x, y] \in \left[0, 2^B - 1 \right] \quad \text{with} \quad B = \text{bit depth}$$

➡ Single-component image is characterized by

- Image width W
- Image height H
- Sample bit depth B

➡ Number of bits for raw image data

$$N_{bits} = W \cdot H \cdot B$$



Gray-Level Image Example: Impact of Image Size (Spatial Resolution)

400×300 samples

200×150 samples

100×75 samples

50×38 samples

Gray-Level Image Example: Impact of Image Size (Spatial Resolution)

400×300 samples

200×150 samples

(interpolated to 400×300)

100×75 samples

(interpolated to 400×300)

50×38 samples

(interpolated to 400×300)

Gray-Level Image Example: Impact of Sample Bit Depth

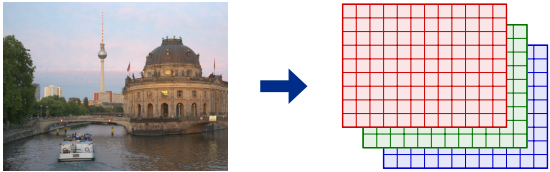
8 bits per sample

6 bits per sample

4 bits per sample

2 bits per sample

Color Images



- Human eye has three types of color receptors
  - ➡ Require three color components
  - ➡ Cameras/displays typically use red, green, blue

➡ A color image is represented as **three matrices of samples** (one for each color component)

➡ Number of bits for raw image data

$$N_{bits} = 3 \cdot W \cdot H \cdot B$$

red, green, blue

sample values

YCC Color Formats

- RGB (red,green,blue) format unsuitable for compression: Large amount of statistical dependencies

➡ YCC formats: Transform samples at the same spatial position (x, y)

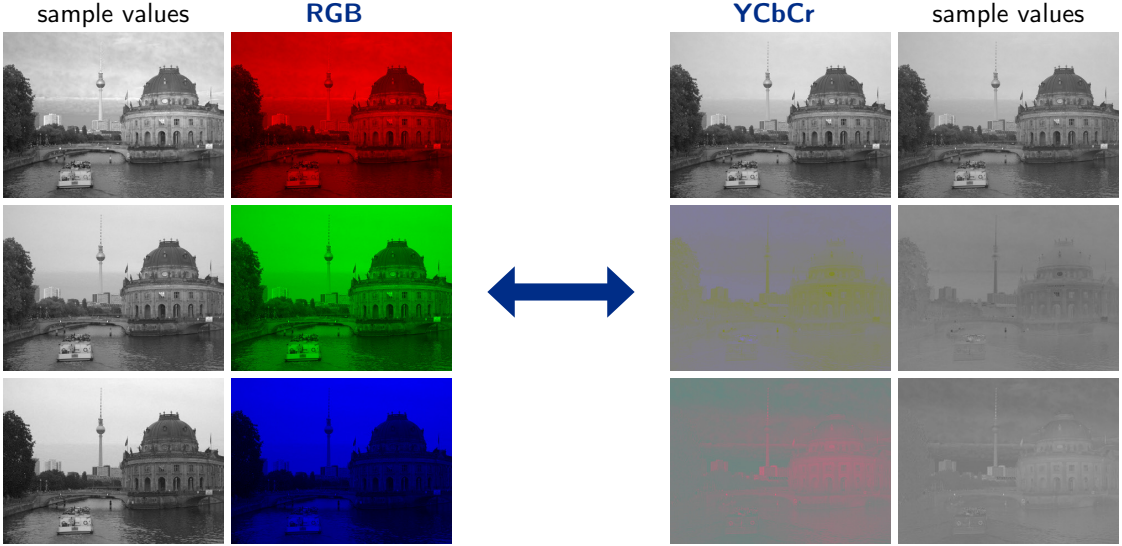
$$\text{RGB} \mapsto \text{YCC}: \quad \begin{bmatrix} Y[x,y] \\ C_1[x,y] \\ C_2[x,y] \end{bmatrix} = \text{round}\left(\begin{bmatrix} & & \\ & \boldsymbol{M}_{3\times 3} & \\ & & \end{bmatrix} \cdot \begin{bmatrix} R[x,y] \\ G[x,y] \\ B[x,y] \end{bmatrix} + \begin{bmatrix} 0 \\ 2^{B-1} \\ 2^{B-1} \end{bmatrix} \right)$$

$$\text{YCC} \mapsto \text{RGB}: \quad \begin{bmatrix} R[x,y] \\ G[x,y] \\ B[x,y] \end{bmatrix} = \text{round}\left(\begin{bmatrix} & & \\ & \boldsymbol{M}_{3\times 3}^{-1} & \\ & & \end{bmatrix} \cdot \left(\begin{bmatrix} Y[x,y] \\ C_1[x,y] \\ C_2[x,y] \end{bmatrix} - \begin{bmatrix} 0 \\ 2^{B-1} \\ 2^{B-1} \end{bmatrix} \right) \right)$$

(transform matrix $\boldsymbol{M}_{3\times 3}$ depends on actual YCC format and RGB color space)

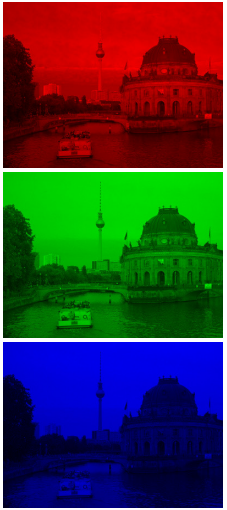
- Most common format in image and video coding: **YCbCr format**
  - **Y** : Luma component (representing brightness)
  - **Cb** : Scaled difference between blue and luma
  - **Cr** : Scaled difference between red and luma

Example: Comparison of RGB and YCbCr format



Color Sampling Formats

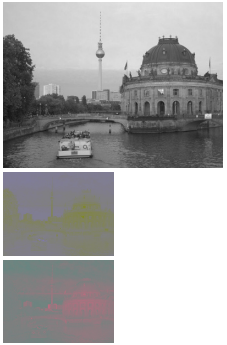
RGB



YCbCr 4:4:4

YCbCr 4:2:2

YCbCr 4:2:0



most common color format

[half the number of RGB samples]

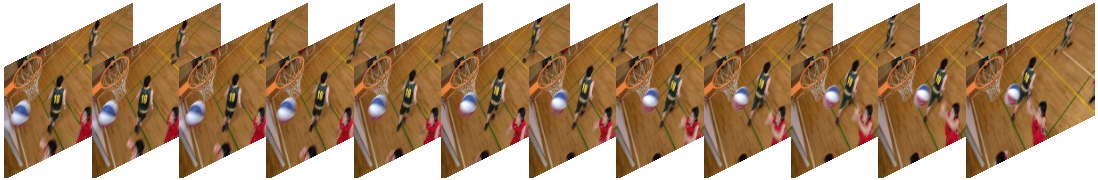
Video: Sequence of Images

Video

- Sequence of images

➡ Characterized by

- Image size $W \times H$
- Sample bit depth B
- Color format (typically, YCbCr 4:2:0)
- **Frame rate** F (pictures per second)



Video Example: Impact of Frame Rate

$F = 50\,\text{Hz}$ (50 pictures per second)

$F = 5\,\text{Hz}$ (5 pictures per second)

Raw Video Data Rate

Raw Video Data Rate: Bit rate of raw video data

$$R_{raw} = (\text{samples per time unit}) \cdot (\text{bit depth per sample})$$
$$= (\text{frame rate } F) \cdot (\text{image size } W \cdot H) \cdot (\text{color format factor } C) \cdot (\text{bit depth } B)$$

Example 1: Ultra High Definition (UHD) Video

- 60 pictures per second (US, Japan)
- 3840×2160 luma samples, YCbCr 4:2:0 color format, 10 bits per sample

➡ **raw data rate**: $R_{raw} = 60\,\text{Hz} \cdot 3840 \cdot 2160 \cdot (3/2) \cdot 10\,\text{bits} \approx \mathbf{7.5\,Gbits/s}$

➡ two-hour video: file size $\approx$ **6.7 TByte** (without audio)

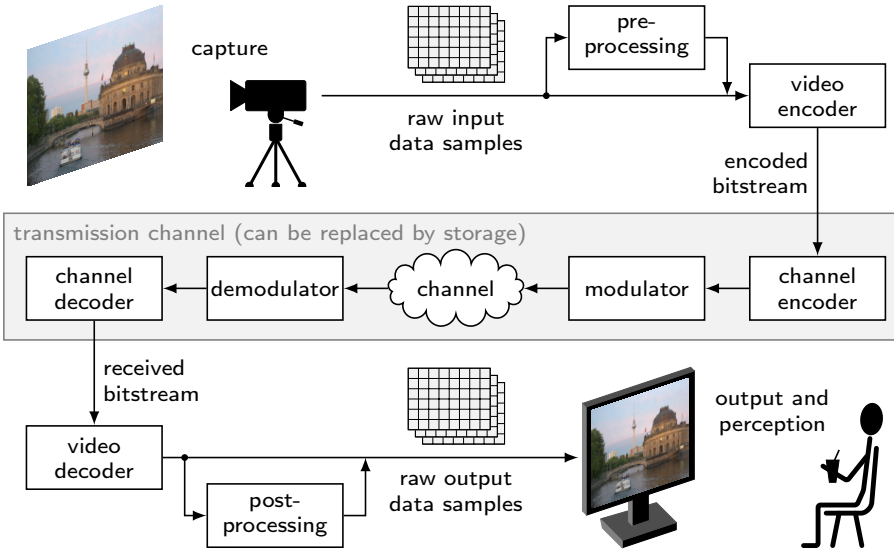
Example 2: Full HD Lecture Video

- 60 pictures per second
- 1920×1080 luma samples (full HD), RGB color format (screen capture), 8 bits per sample

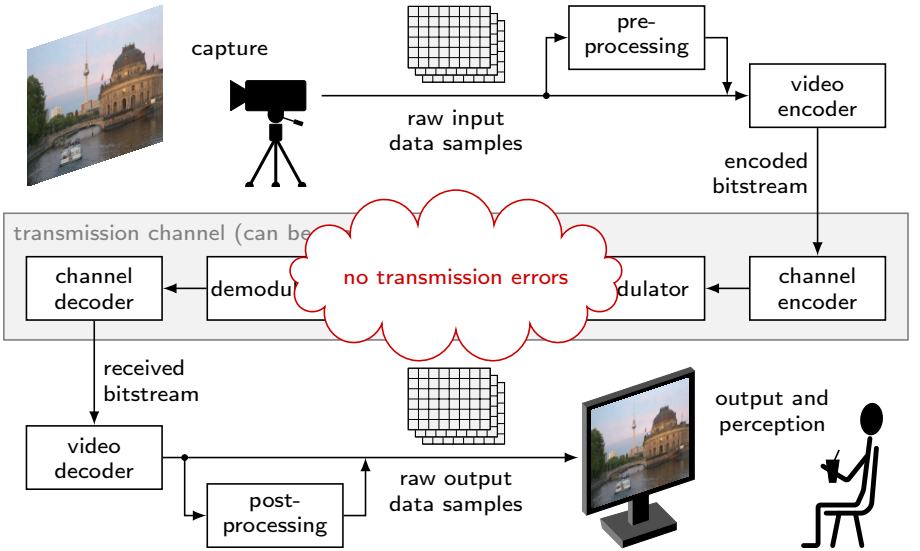
➡ **raw data rate**: $R_{raw} = 60\,\text{Hz} \cdot 1920 \cdot 1080 \cdot 3 \cdot 8\,\text{bits} \approx \mathbf{2.986\,Gbits/s}$

➡ 90 min. lecture: file size $= 2.986\,\text{Gbits/s} \cdot 90\,\text{min} \cdot 60\,\text{s/min} \,/\, (8\,\text{bit/byte}) \approx$ **2 TByte**

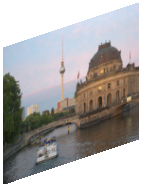
Typical Video Communication Scenario



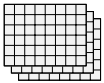
Typical Video Communication Scenario



Typical Video Communication Scenario



capture

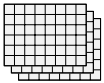


raw input
data samples

video
encoder

bitstream

video
decoder



raw output
data samples

output and
perception

Application Examples

	HD movie on Blu-ray Disc	**UHD broadcast over DVB-S2**	**Video conference over the Internet**
raw video format	1920×1080 luma samples YCbCr 4:2:0 color format 8 bits per sample 24 frames per second	3840×2160 luma samples YCbCr 4:2:0 color format 10 bits per sample 60 frames per second	1280×720 luma samples YCbCr 4:2:0 color format 8 bits per sample 50 frames per second
raw data rate	ca. **600 Mbit/s**	ca. **7.5 Gbit/s**	ca. **550 Mbit/s**
channel bit rate	36 Mbit/s (read speed)	58 Mbit/s (8PSK 2/3)	depends on connection
video bit rate	ca. **20 Mbit/s**	ca. **15 Mbit/s**	ca. **1 Mbit/s**
required compression	ca. **30 : 1**	ca. **500 : 1**	ca. **500 : 1**

Types of Image and Video Compression

Lossless Compression

- Invertible / reversible: Original input data can be completely recovered

➡ Examples:
- PNG, JPEG-LS for images
- H.265 | HEVC lossless for video

➡ Achievable compression ratios typically in range from 2:1 to 3:1

Lossy Compression

- Not invertible: Only **approximation of original input data** can be recovered
- Achieves **much higher compression ratios**

➡ Examples:
- JPEG, JPEG-2000 for images
- MPEG-2, H.264 | AVC, H.265 | HEVC, H.266 | VVC for video

➡ **Dominant form of compression for images and video**

The Basic Image and Video Coding Problem

Image and Video Coding Problem

- Two equivalent formulations:

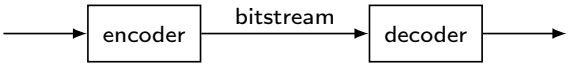
Representing images/videos with the highest fidelity possible within an available bit rate

and

Representing images/videos using the lowest bit rate possible while maintaining a specified reproduction quality

Image/Video Codec

- **Codec**: System of en**co**der and **dec**oder



Video Coding in Practice

Characteristics of Video Codecs

- Bit rate: Throughput of the communication channel
- Quality: Fidelity of the reconstructed signal
- Delay: Start-up latency, end-to-end delay
- Complexity: Computational complexity, memory requirement, memory access requirements

Practical Video Coding Problem

Given a maximum allowed complexity and a maximum delay, achieve an optimal trade-off between bit rate and reconstruction quality for the transmission problem in the targeted application

In this course:

➡ Will concentrate on basic video codec

➡ Ignore aspects of transmission channel (e.g., transmission errors)

Intermediate Summary: Goal of Image and Video Coding

Raw Data Format for Images and Videos

- Gray-Level Images: Matrix of samples (image size, bit depth per sample)
- Color Images: Three arrays of samples (typically YCbCr 4:2:0)
- Video: Sequence of images (frame rate)

➡ **Extremely large raw data rate** (for example: 7.5 GBits/s for UHD 60Hz)

Image and Video Coding

- Represent image/video data using much smaller bit rate (suitable for certain application)

➡ Require **lossy compression** (approximation of original input data)

➡ Main goal of image and video:

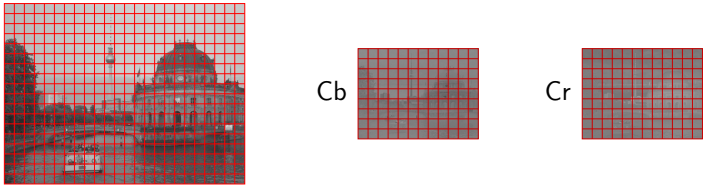
Best possible quality for given bit rate, or

Smallest possible bit rate for given quality

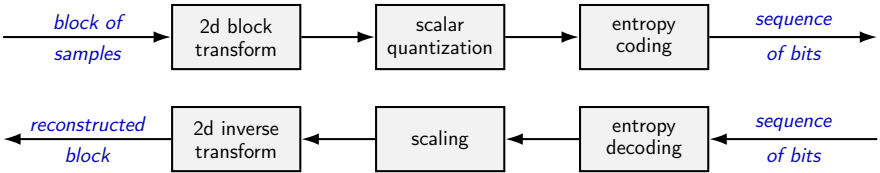
- In practice: Take into account delay and complexity

Image Compression Example: JPEG Baseline

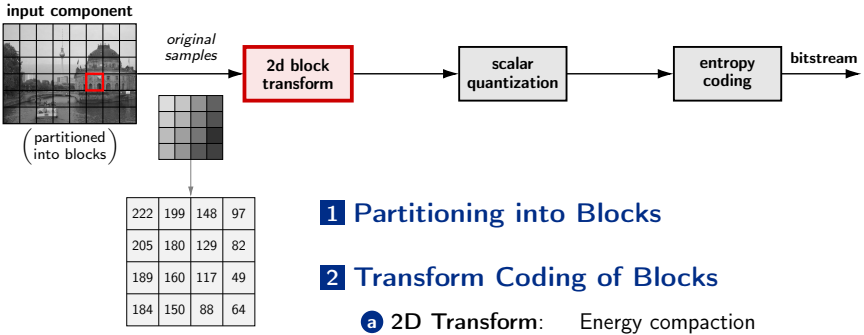
YCbCr 4:2:0
color format



- Partition color components (Y, Cb, Cr) into blocks of 8×8 samples
- Transform coding of 8×8 blocks of samples



JPEG Principle — Transform Coding of Sample Blocks



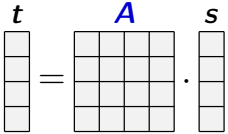
1 Partitioning into Blocks

2 Transform Coding of Blocks

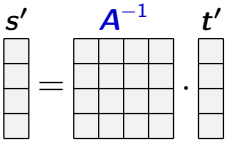
- **a** **2D Transform**: Energy compaction
- **b** **Quantization**: Approximate signal (remove invisible details)
- **c** **Entropy Coding**: Represent data with as little bits as possible

Orthogonal Transform of Sample Vectors

forward transform:



inverse transform:



Linear Transform of a Vector of Samples

- Consider vector $\boldsymbol{s}$ of neighboring samples (e.g., row or column of a block)
- Forward transform and inverse transform: Matrix-vector multiplications

$$\boldsymbol{t} = \boldsymbol{A} \cdot \boldsymbol{s} \qquad \boldsymbol{A} : \text{transform matrix}$$

$$\boldsymbol{s}' = \boldsymbol{A}^{-1} \cdot \boldsymbol{t}' \qquad \boldsymbol{t} : \text{vector of transform coefficients}$$

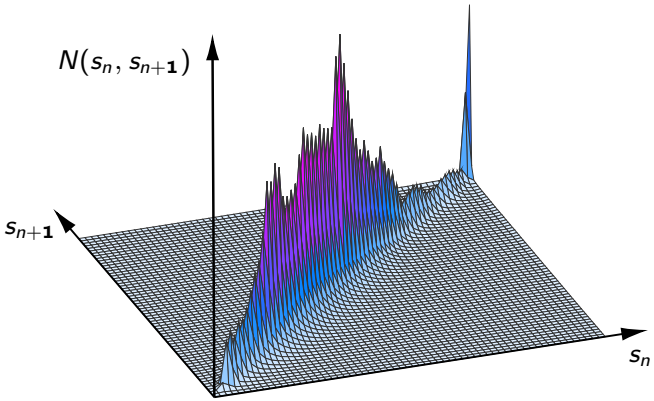
Orthogonal Transform

- Transform matrix $\boldsymbol{A}$ has the property: $\boldsymbol{A}^{-1} = \boldsymbol{A}^{\mathrm{T}}$ (rotation/reflection in signal space)

➡ Same mean squared error (MSE) in sample and transform domain

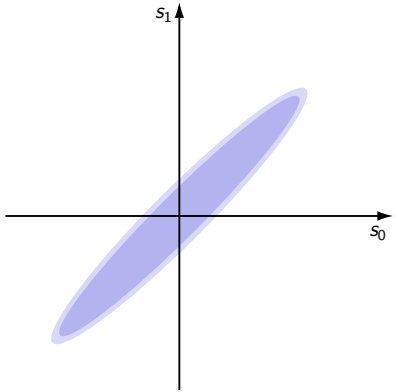
Example: 2D Histogram for Natural Gray-Level Images

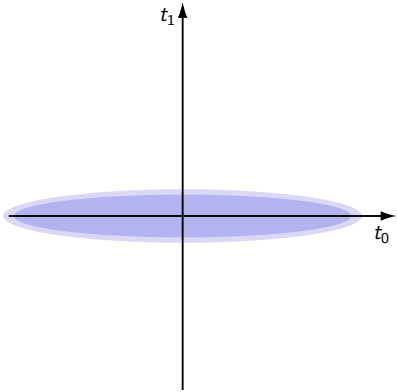
joint histogram
of two horizontally
adjacent samples



15 test images (each 768×512)

Effect of Transform for Correlated Sample Vectors





s_0	s_1

Orthogonal transform:

$$\boldsymbol{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

concentration of signal energy into few coefficients

Transforms in Image and Video Coding

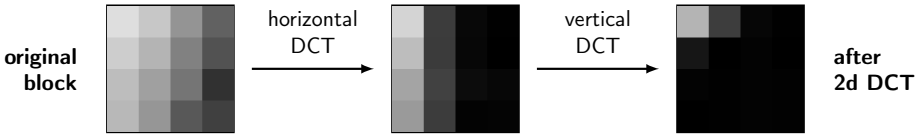
$$\boldsymbol{t} = \boldsymbol{A}\,\boldsymbol{s}\,\boldsymbol{A}^{\mathrm{T}} \quad \text{and} \quad \boldsymbol{s}' = \boldsymbol{A}^{\mathrm{T}}\,\boldsymbol{t}'\,\boldsymbol{A}$$

Separable Orthogonal Block Transforms

- Transform of rows and columns of a block

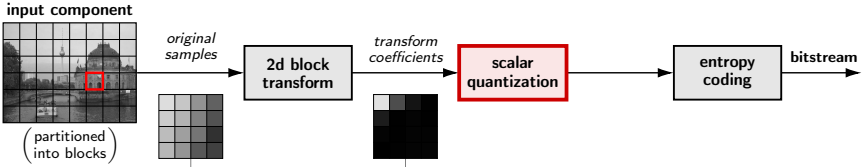
Transforms in Image and Video Coding

- Most often: **Discrete Cosine Transform (DCT)** or approximation thereof

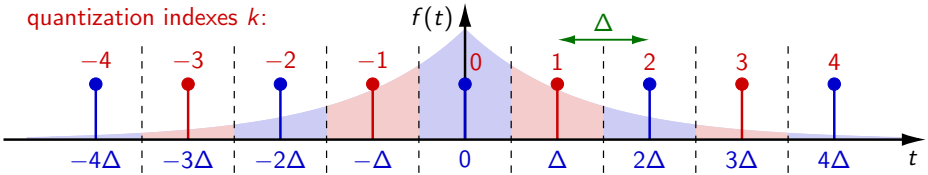


➡ **Effect of Transform: Compaction of Signal Energy**

JPEG Principle — Transform Coding of Sample Blocks



Scalar Quantization



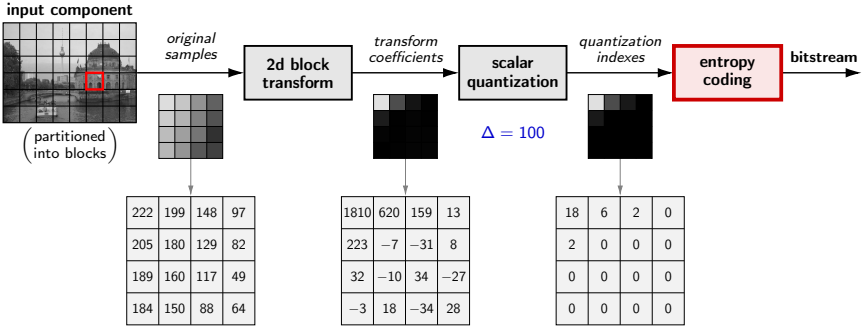
Typical Scalar Quantizer: Uniform Reconstruction Quantizer

- Reconstruction levels: Uniformly spaced and centered around zero (quantization step size Δ)
- **Simple decoder operation**: $t' = k \cdot \Delta$ (k: quantization index)
- **Encoder**: Freedom to adapt decision to source and entropy coding
  - ➡ Simplest encoder: $k = \text{round}\,(t/\Delta)$

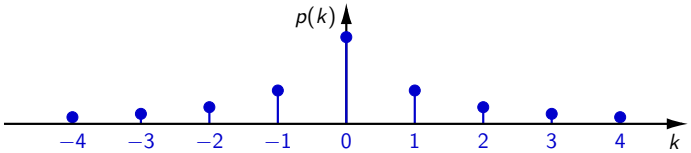
➡ **Effect of Quantization: Approximation of Original Signal**

- ➡ Quantization step size Δ determines trade-off between quality and bit rate

JPEG Principle — Transform Coding of Sample Blocks



Entropy Coding of Quantization Indexes



Lossless Coding of Quantization Indexes

- Simplest approach: Codeword table
- Consider symbol probabilities

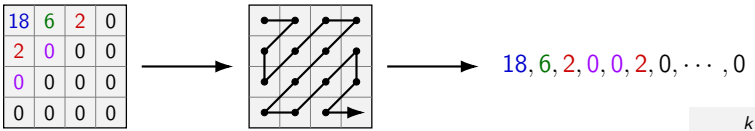
Example

- indexes k: $18, 6, 2, 0, 2, 0, \cdots, 0$ (16 values)
- ➡ simple: $7 + 7 + 2 \cdot 7 + 12 \cdot 6 = 100$ bits
- ➡ better: $10 + 6 + 2 \cdot 4 + 12 \cdot 1 = 36$ bits

➡ **Entropy Coding: Minimize Number of Bits**

k	simple codewords	better codewords
0	000000	0
±1	000001 s	10 0 s
±2	000010 s	10 1 s
±3	000011 s	110 00 s
±4	000100 s	110 01 s
±5	000101 s	110 10 s
±6	000110 s	110 11 s
±7	000111 s	1110 000 s
±8	001000 s	1110 001 s
...	...	...
±14	001110 s	1110 111 s
±15	001111 s	11110 0000 s
±16	010000 s	11110 0001 s
±17	010001 s	11110 0010 s
±18	010010 s	11110 0011 s
...	...	...
±63	111111 s	1111110 000000 s

Entropy Coding: Exploitation of Dependencies



Typical Blocks of Quantization Indexes

- Most zeros are at high-frequency locations

Simple Entropy Coding Improvement

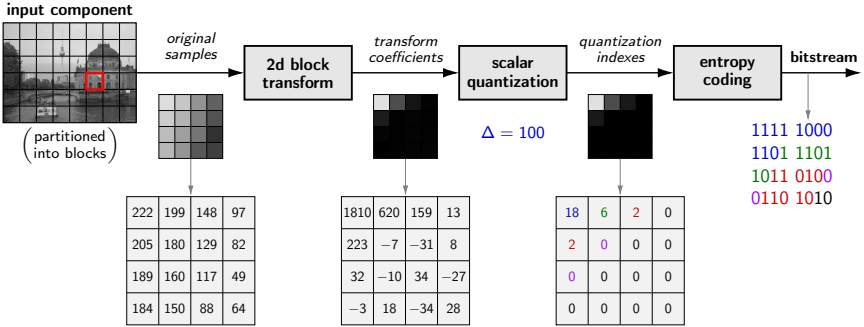
- Scan quantization indexes using zig-zag scan
- Include end-of-block symbol (eob) in code

Example

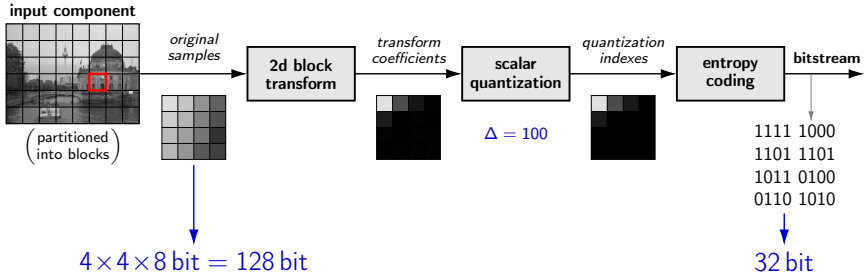
- scanned k: $18, 6, 2, 0, 0, 2, 0, \cdots, 0$ (16 values)
- ➡ new code: $11 + 7 + 5 + 2 \cdot 1 + 5 + 2 = 32$ bits

k	code with eob
0	0
(eob)	10
±1	110 0 s
±2	110 1 s
±3	1110 00 s
±4	1110 01 s
±5	1110 10 s
±6	1110 11 s
±7	11110 000 s
±8	11110 001 s
...	...
±14	11110 111 s
±15	111110 0000 s
±16	111110 0001 s
±17	111110 0010 s
±18	111110 0011 s
...	...
±63	11111110 000000 s

JPEG Principle — Transform Coding of Sample Blocks



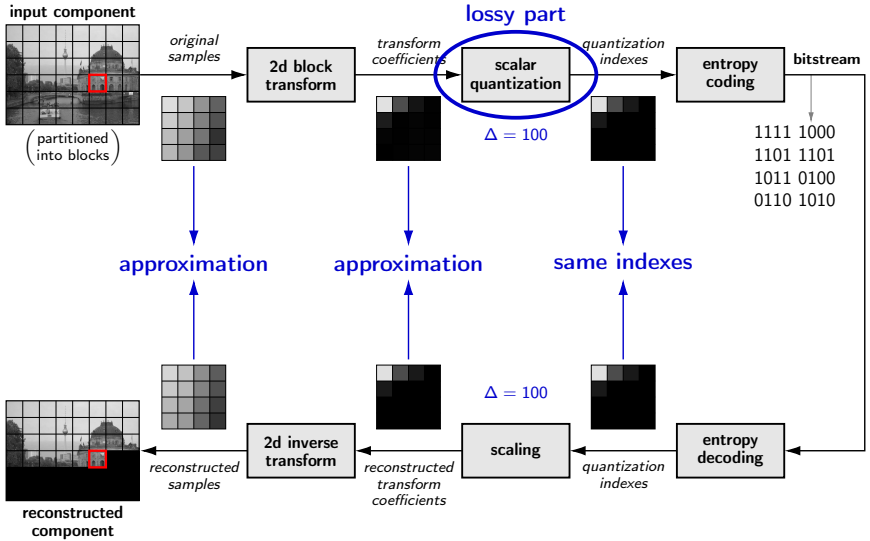
JPEG Principle — Transform Coding of Sample Blocks



➡ Reduction to 25% of raw data size

➡ Compression factor: 4.0

JPEG Principle — Transform Coding of Sample Blocks



JPEG Principle — Transform Coding of Sample Blocks

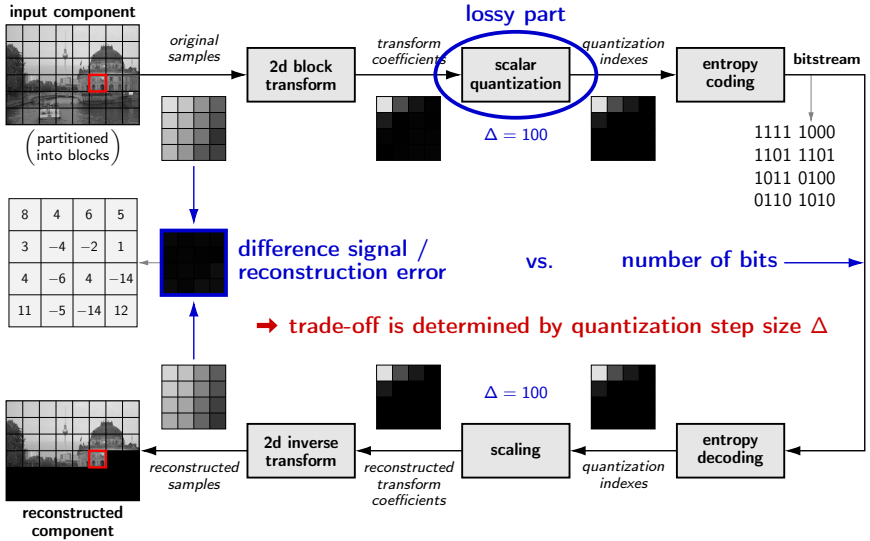


Image Compression: Quality versus Compression Ratio

Original Image (1024 × 640 image points, 1966 KB)

100 %

Image Compression: Quality versus Compression Ratio

Lossy Compressed: JPEG (Quality 95)

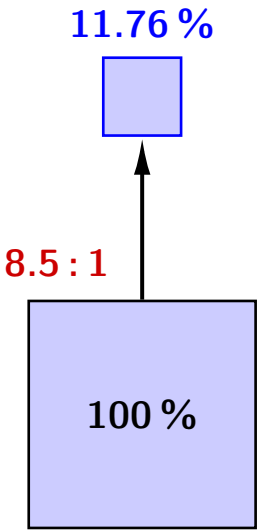


Image Compression: Quality versus Compression Ratio

Lossy Compressed: JPEG (Quality 75)

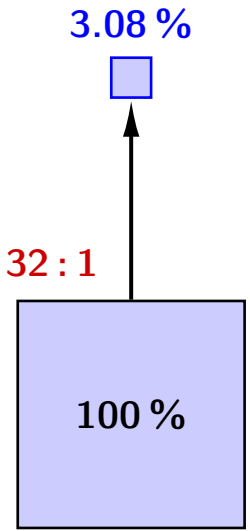


Image Compression: Quality versus Compression Ratio

Lossy Compressed: JPEG (Quality 50)

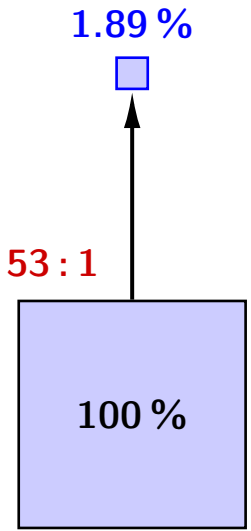


Image Compression: Quality versus Compression Ratio

Lossy Compressed: JPEG (Quality 25)

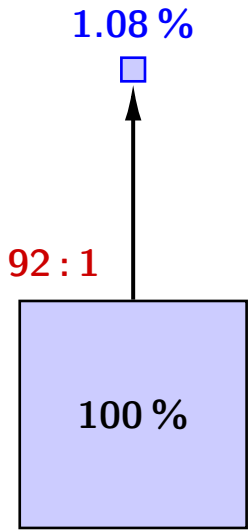
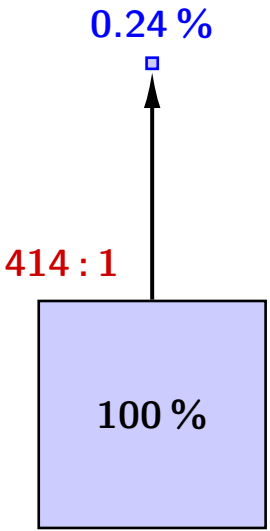


Image Compression: Quality versus Compression Ratio

Lossy Compressed: JPEG (Quality 1)



Intermediate Summary: Basic Principle of JPEG

JPEG Baseline

- Coding of color images in YCbCr 4:2:0 format
- All color components are partitioned into 8×8 blocks of samples
- Each 8×8 block is coded using transform coding

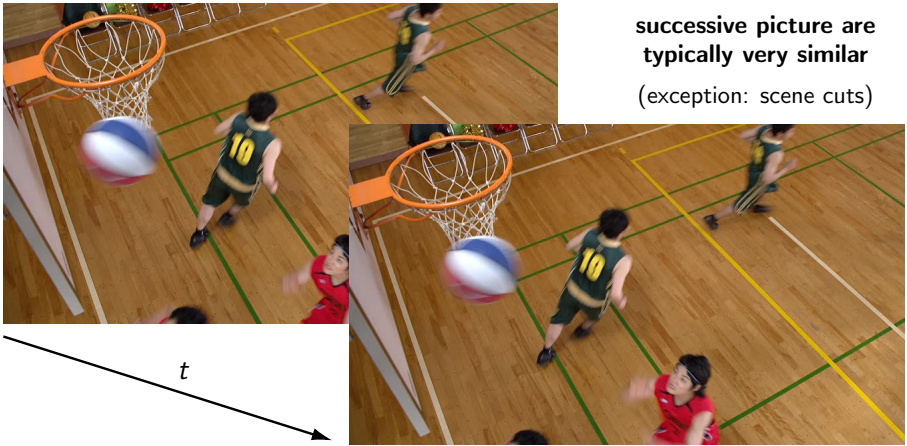
Transform Coding of Sample Blocks

1. Orthogonal Block Transform
   - ➡ Compaction of signal energy (exploit strong dependencies between neighboring samples)
2. Scalar Quantization of Transform Coefficients
   - ➡ Approximation of transform coefficients (divide by quantization step size, rounding)
3. Entropy Coding of Quantization Indexes
   - ➡ Represent quantization indexes (integer numbers) with as little bits as possible

➡ Quantization step size controls trade-off between bit rate and quality

Example Video (832×480 @ 50 Hz)

Successive Pictures in Video Sequences



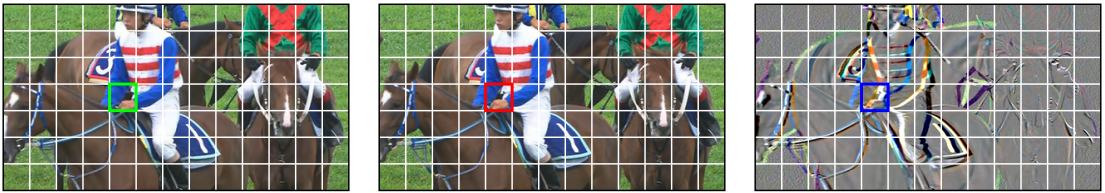
successive picture are typically very similar

(exception: scene cuts)

Important: Utilize large amount of dependencies between video picture

➡ **Basic Idea**: Predict current picture from already coded previous picture

Simple Variant: Frame Difference Coding



- Partition current picture s_n into rectangular blocks
- Block-wise coding of current picture s_n
  1. Get prediction error: $u_n[x, y] = s_n[x, y] - s'_{n-1}[x, y]$
  2. Transform coding: $u_n[x, y] \mapsto u'_n[x, y]$ (similar as in JPEG)
  3. Transmit in bitstream: Quantization indexes $\{k\}$
  4. Reconstruction: $s'_n[x, y] = s'_{n-1}[x, y] + u'_n[x, y]$

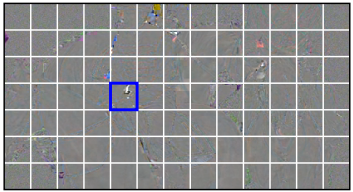
➡ **Problem: Ineffective for moving regions**

Important Improvement: Motion-Compensated Prediction

rec. previous picture s'_{n-1}

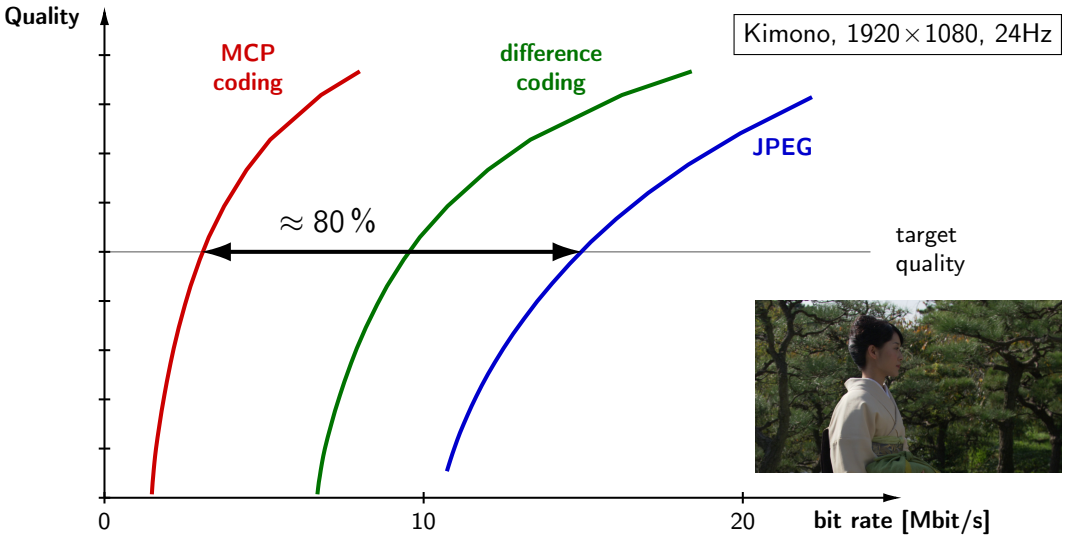
current original picture s_n

prediction error u_n (MCP)



- Estimate motion of blocks in current picture relative to previous picture s'_{n-1}
  - ➡ Motion is specified by displacement / motion vector (m_x, m_y)
- Block-wise coding of current picture s_n
  1. Get prediction error: $u_n[x, y] = s_n[x, y] - s'_{n-1}[x + m_x, y + m_y]$
  2. Transform coding: $u_n[x, y] \mapsto u'_n[x, y]$ (similar as in JPEG)
  3. Transmit in bitstream: Motion vector (m_x, m_y) and quantization indexes $\{k\}$
  4. Reconstruction: $s'_n[x, y] = s'_{n-1}[x + m_x, y + m_y] + u'_n[x, y]$

Efficiency of Motion-Compensated Prediction



Video Compression: Quality versus Compression Ratio

H.264 | AVC @ 2.7 Mbit/s (90 : 1)

H.264 | AVC @ 400 kbit/s (600 : 1)

Original: 240 Mbits/s (832×480, 50Hz, YCbCr 4:2:0, 8 bit)

Measuring Coding Efficiency: Average Bit Rate

Core Properties of a Bitstream

- Average bit rate (in practice: distribution over time matters also)
- Quality of reconstructed image or video

Average Bit rate

images: $$R = \frac{\text{number of bits in bitstream for the image}}{\text{number of luma samples in the image}}$$

video: $$R = \frac{\text{number of bits in bitstream for the video sequence}}{\text{nominal duration of the video sequence}}$$

Reconstruction Quality

- Ideally: Quality as perceived by human being
- In practice: Often use **Peak Signal-to-Noise Ratio (PSNR)** (based on square error)

Measuring Coding Efficiency: Peak Signal-To-Noise Ratio (PSNR)

Distortion Measure: Mean Squared Error (MSE)

$$\textit{single component}: \quad \mathrm{MSE} = \frac{1}{N} \sum_{\forall x,y} \Big(s'[x,y] - s[x,y] \Big)^2$$

$$\textit{color image}: \quad \mathrm{MSE} = \frac{1}{N} \sum_{\forall c} \sum_{\forall x,y} \Big(s'_c[x,y] - s_c[x,y] \Big)^2$$

$s[x,y]$: original samples

$s'[x,y]$: reconstructed samples

N : total number of samples in image

Quality Measure: Peak Signal-To-Noise Ratio (PSNR)

$$\textit{image}: \quad \mathrm{PSNR}\ [\mathrm{dB}] = 10 \cdot \log_{10} \left(\frac{s_{\mathrm{max}}^2}{\mathrm{MSE}} \right) \qquad \Big(s_{\mathrm{max}} = 2^B - 1 \Big)$$

$$\textit{video}: \quad \mathrm{PSNR}\ [\mathrm{dB}] = \frac{1}{K} \sum_{k=0}^{K-1} \mathrm{PSNR}(k) \qquad \Big(\text{average over all pictures} \Big)$$

Trade-Off between Quality and Compression Ratio

Coding Efficiency

- Ability to trade-off bit rate and reconstruction quality
- Want **best reconstruction quality for a given bitrate** (or vice versa)

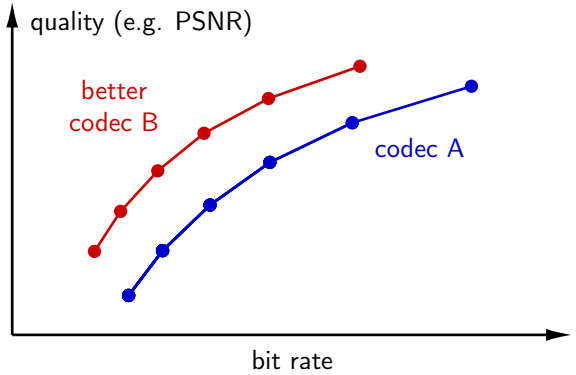
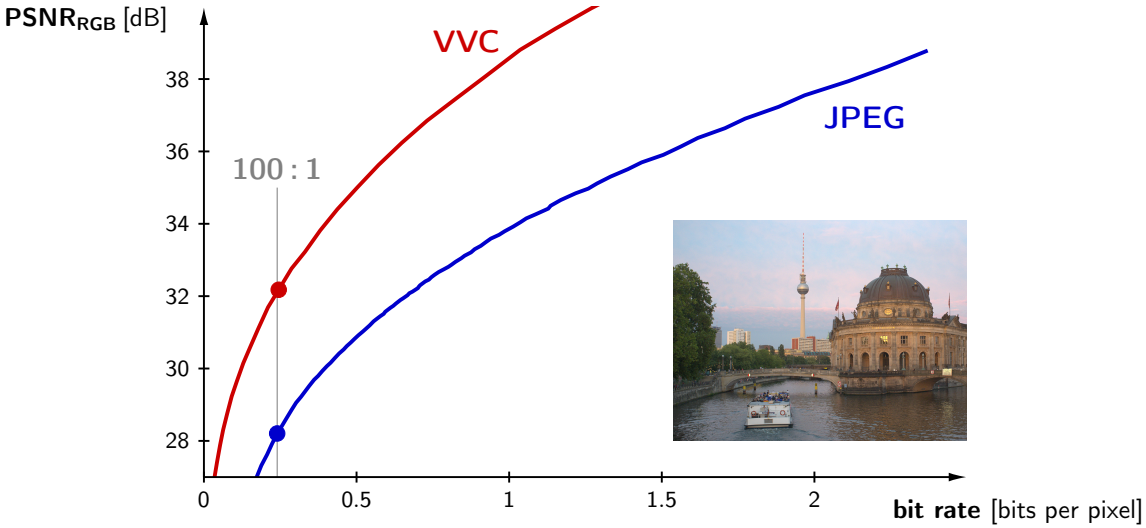


Image Coding Comparion: JPEG versus VVC



Visual Comparison: JPEG versus VVC

$\approx$ 100:1 compression

JPEG (RGB-PSNR = 28.2 dB)

VVC (RGB-PSNR = 32.2 dB)

Summary of Lecture

Image and Video Applications

- Very large raw data rates / file sizes

➜ Require compression for transmission and storage

Goal of Image and Video Coding

- Minimize bit rate while preserving certain reconstruction quality, or
- Maximize reconstruction quality while not exceeding given bit rate budget

Basic Techniques in Image and Video Coding

- Transform coding of sample blocks (transform, quantization, entropy coding)
- Motion-compensated prediction

Coding Efficiency

- Trade-off between bit rate and reconstruction quality
- Simple quality measure: Peak Signal-to-Noise Ratio (PSNR)

Outline of Course

Raw Data Formats

- Human vision, image capture, image display, representation formats

Image Coding

- The JPEG Standard
- Improvements after JPEG

Hybrid Video Coding

- Motion-compensated coding
- Improved inter-picture coding concepts
- State-of-the art video coding standards

Exercises

➡ Collaborative implementation of our own image codec (possible extension to video codec)

➡ Step-by-step improvement using concepts discussed in lectures