Information Theory and Coding – Image, Video and Audio Compression

Markus Kuhn

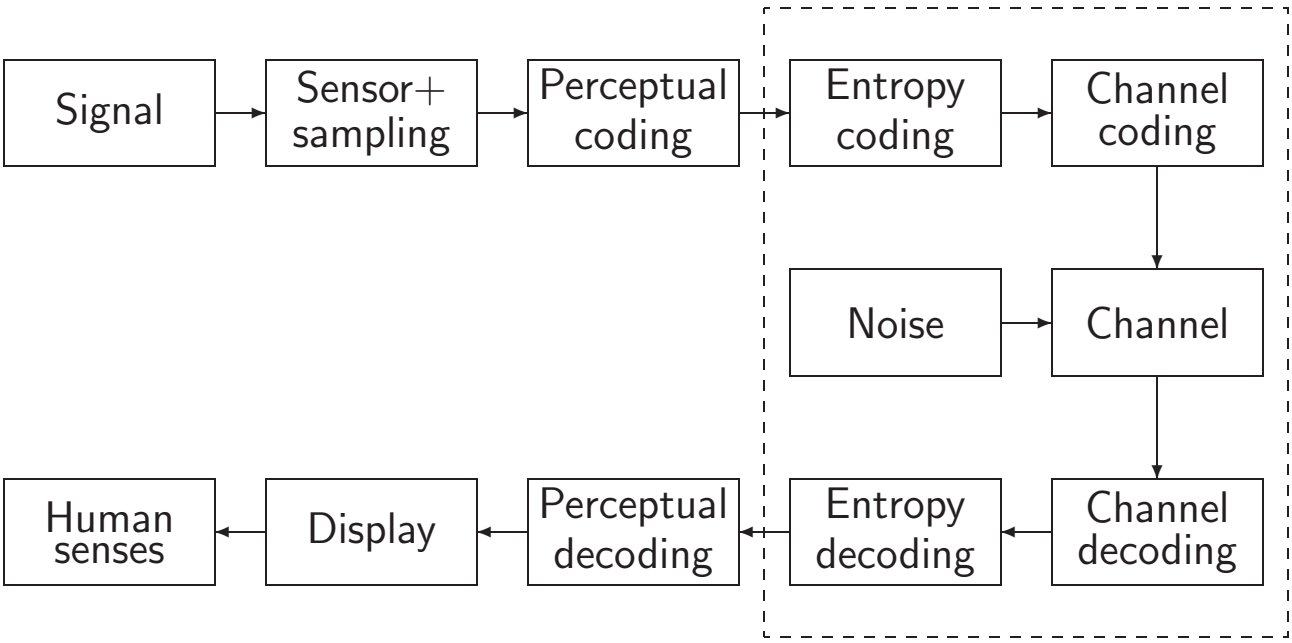
Lent 2003 – Part II

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

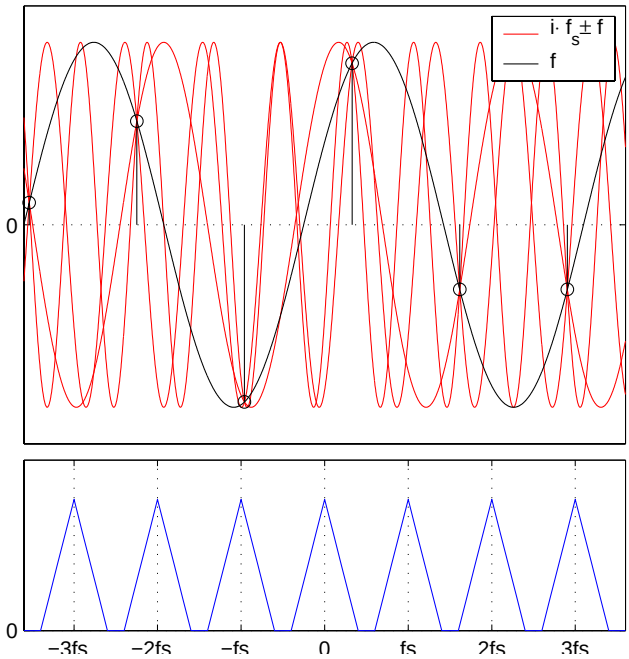
http://www.cl.cam.ac.uk/Teaching/2002/InfoTheory/

Structure of modern audiovisual communication systems



The dashed box marks the focus of the main part of this course as taught by Neil Dodgson.

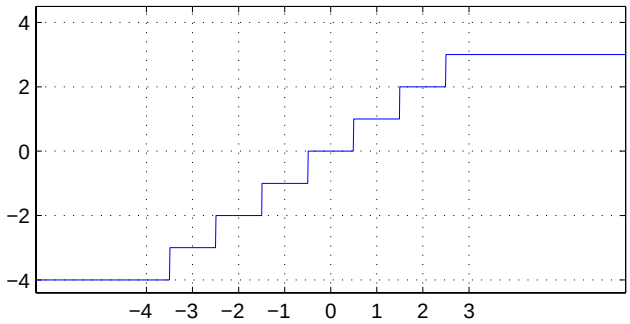
Sampling, aliasing and Nyquist limit



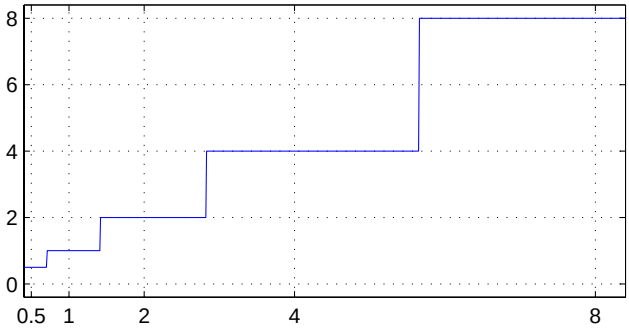
A wave $\cos(2\pi t f)$ sampled with frequency f_s cannot be distinguished from $\cos(2\pi t(if_s \pm f))$ for any $i \in \mathbb{Z}$, therefore ensure $|f| < f_s/2$.

Quantization

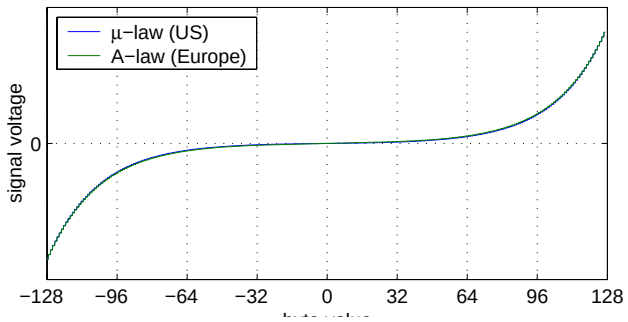
Uniform:



Non-uniform (e.g., logarithmic):



Example for non-uniform quantization: digital telephone network



Simple logarithm fails for values $\leq 0 \rightarrow$ apply μ-law compression

$$y = \frac{V \log(1 + \mu|X|/V)}{\log(1 + \mu)} \text{sgn}(x)$$

before uniform quantization ($\mu = 255$, V maximum value).

Lloyd's algorithm: finds least-square-optimal non-uniform quantization function for a given probability distribution of sample values.

S.P. Lloyd: Least Squares Quantization in PCM. IEEE Trans. on Information Theory. Vol. 28, March 1982, pp 129–137.

Psychophysics of perception

Sensation limit (SL) = lowest intensity stimulus that can still be perceived

Difference limit (DL) = smallest perceivable stimulus difference at given intensity level

Weber's law

Difference limit $\Delta\phi$ is proportional to the intensity ϕ of the stimulus (except for a small correction constant a describe deviation of experimental results near SL):

$$\Delta\phi = c \cdot (\phi + a)$$

Fechner's scale

Define a perception intensity scale ψ using the sensation limit ϕ_0 as the origin and the respective difference limit $\Delta\phi = c \cdot \phi$ as a unit step. The result is a logarithmic relationship between stimulus intensity and scale value:

$$\psi = \log_c \frac{\phi}{\phi_0}$$

Fechner's scale matches older subjective intensity scales that follow differentiability of stimuli, e.g. the astronomical magnitude numbers for star brightness introduced by Hipparchos (≈150 BC).

Stevens' law

A sound that is 20 DL over SL is perceived as more than twice as loud as one that is 10 DL over SL, i.e. Fechner's scale does not describe well perceived intensity. A rational scale attempts to reflect subjective relations perceived between different values of stimulus intensity ϕ. Stevens observed that such rational scales ψ follow a power law:

$$\psi = k \cdot (\phi - \phi_0)^a$$

Example coefficients a: temperature 1.6, weight 1.45, loudness 0.6, brightness 0.33.

Decibel

Communications engineers love logarithmic units:

- ⟶ Quantities often vary over many orders of magnitude ⟶ difficult to agree on a common SI prefix
- ⟶ Quotient of quantities (amplification/attenuation) usually more interesting than difference
- ⟶ Signal strength usefully expressed as field quantity (voltage, current, pressure, etc.) or power, but quadratic relationship between these two ($P = U^2/R = I^2R$) rather inconvenient
- ⟶ Weber/Fechner: perception is logarithmic

Plus: Using magic special-purpose units has its own odd attractions (⟶ typographers, navigators)

Neper (Np) denotes the natural logarithm of the quotient of a field quantity F and a reference value F_0.

Bel (B) denotes the base-10 logarithm of the quotient of a power P and a reference power P_0. Common prefix: 10 decibel (dB) = 1 bel.

Where P is some power and P_0 a 0 dB reference power, or F is a field quantity and F_0 the reference:

$$10\ \text{dB} \cdot \log_{10} \frac{P}{P_0} = 20\ \text{dB} \cdot \log_{20} \frac{F}{F_0}$$

Common reference vales indicated with additional letter afer dB:

$$\begin{aligned}
0\ \text{dBW} &= 1\ \text{W} \\
0\ \text{dBm} &= 1\ \text{mW} = -30\ \text{dBW} \\
0\ \text{dB}\mu\text{V} &= 1\ \mu\text{V} \\
0\ \text{dB}_{\text{SPL}} &= 20\ \mu\text{Pa} \quad \text{(sound pressure level)} \\
0\ \text{dB}_{\text{SL}} &= \text{perception threshold (sensation level)}
\end{aligned}$$

3 dB = double power, 6 dB = double pressure/voltage/etc.
10 dB = 10× power, 20 dB = 10× pressure/voltage/etc.

RGB video colour coordinates

Hardware interface (VGA): red, green, blue signals with 0–0.7 V

Electron-beam current and photon count of cathode-ray display are proportional to $(v - v_0)^\gamma$, where v is the video-interface or screen-grid voltage and γ is usually in the range 1.5–3.0. CRT non-linearity is compensated electronically in TV cameras and approximates Stevens scale.

Software interfaces map RGB voltage linearly to $\{0, 1, \ldots, 255\}$ or 0–1

Mapping of numeric RGB values to colour and luminosity is at present still highly hardware and sometimes even operating-system or device-driver dependent.

New specification "sRGB" aims to fix meaning of RGB with $\gamma = 2.2$ and standard primary colour coordinates.

http://www.w3.org/Graphics/Color/sRGB
http://www.srgb.com/
IEC 61966

YCrCb video colour coordinates

Human eye processes color and luminosity at different resolutions, therefore use colour space with luminance coordinate

$$Y = 0.3R + 0.6G + 0.1B$$

and colour components

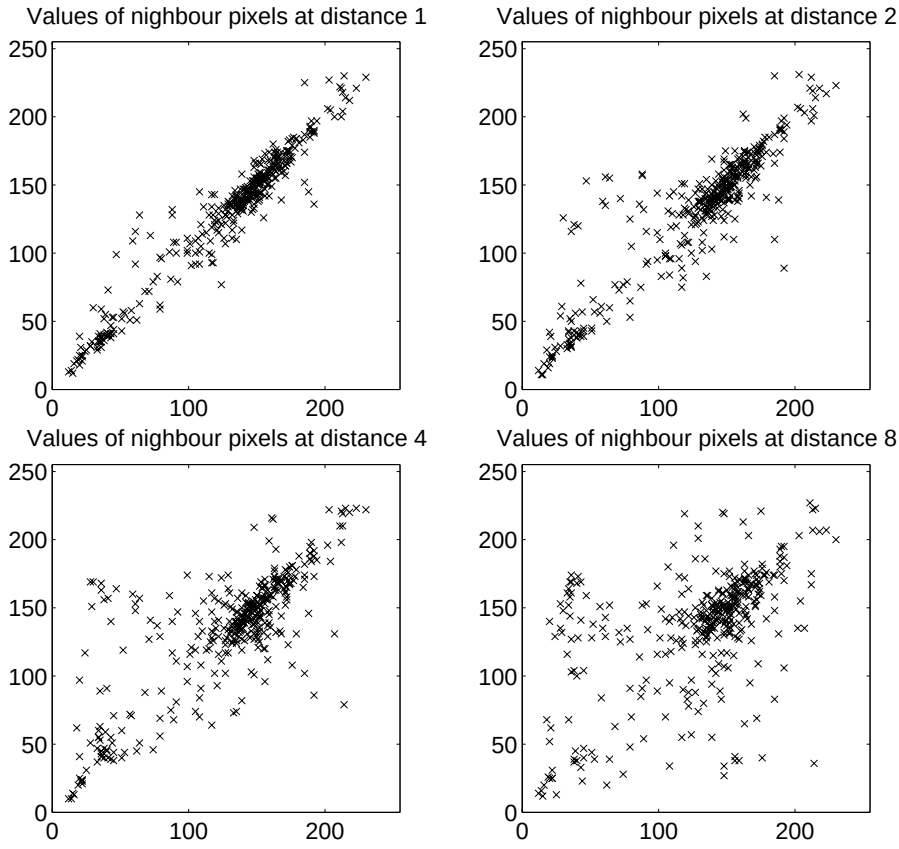
$$\begin{aligned} V &= R - Y = 0.7R - 0.6G - 0.1B \\ U &= B - Y = -0.3R - 0.6G + 0.9B \end{aligned}$$

Since $-0.7 \leq V \leq 0.7$ and $-0.9 \leq U \leq 0.9$, a more convenient normalized encoding of chrominance is:

$$\begin{aligned} Cb &= \frac{U}{2.0} + 0.5 \\ Cr &= \frac{V}{1.6} + 0.5 \end{aligned}$$

Modern image compression techniques operate on Y, Cr, Cb channels separately, using half the resolution of Y for storing Cr, Cb.

Correlation of neighbour pixels



Karhunen-Loève transform (KLT)

Two random variables x, y are not correlated if their covariance

$$\mathrm{cov}(x, y) = E\{(x - E\{x\}) \cdot (y - E\{y\})\} = 0.$$

Take an image (or in practice a small 8×8 pixel block) as a random-variable vector $\mathbf{b}$. The components of a random-variable vector $\mathbf{b} = (b_1, \ldots, b_k)$ are decorrelated if the covariance matrix $\mathrm{cov}(\mathbf{b})$ with

$$(\mathrm{cov}(\mathbf{b}))_{i,j} = E\{(b_i - E\{b_i\}) \cdot (b_j - E\{b_j\})\} = \mathrm{cov}(b_i, b_j)$$

is a diagonal matrix. The Karhunen-Loève transform of $\mathbf{b}$ is the matrix A with which $\mathrm{cov}(A\mathbf{b})$ is diagonal.

Since $\mathrm{cov}(\mathbf{b})$ is symmetric, its eigenvectors are orthogonal. Using these eigenvectors as the rows of A and the corresponding eigenvalues as the diagonal elements of the diagonal matrix D, we obtain the decomposition $\mathrm{cov}(\mathbf{b}) = A^T D A$, and therefore $\mathrm{cov}(A\mathbf{b}) = D$.

The Karhunen-Loève transform is the orthogonal matrix of the singular-value decomposition of the covariance matrix of its input.

Discrete cosine transform (DCT)

The forward and inverse discrete cosine transform

$$\begin{aligned} S(u) &= \frac{C(u)}{\sqrt{N/2}} \sum_{x=0}^{N-1} s(x) \cos \frac{(2x+1)u\pi}{2N} \\ s(x) &= \sum_{u=0}^{N-1} \frac{C(u)}{\sqrt{N/2}} S(u) \cos \frac{(2x+1)u\pi}{2N} \end{aligned}$$

with

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & u > 0 \end{cases}$$

is an orthonormal transform:

$$\sum_{x=0}^{N-1} \frac{C(u)}{\sqrt{N/2}} \cos \frac{(2x+1)u\pi}{2N} \cdot \frac{C(u')}{\sqrt{N/2}} \cos \frac{(2x+1)u'\pi}{2N} = \begin{cases} 1 & u = u' \\ 0 & u \neq u' \end{cases}$$

The 2-dimensional variant of the DCT applies the 1-D transform on both rows and columns of an image:

$$S(u,v) = \frac{C(u)}{\sqrt{N/2}} \frac{C(v)}{\sqrt{N/2}} \cdot \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} s(y,x) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2x+1)v\pi}{2N}$$

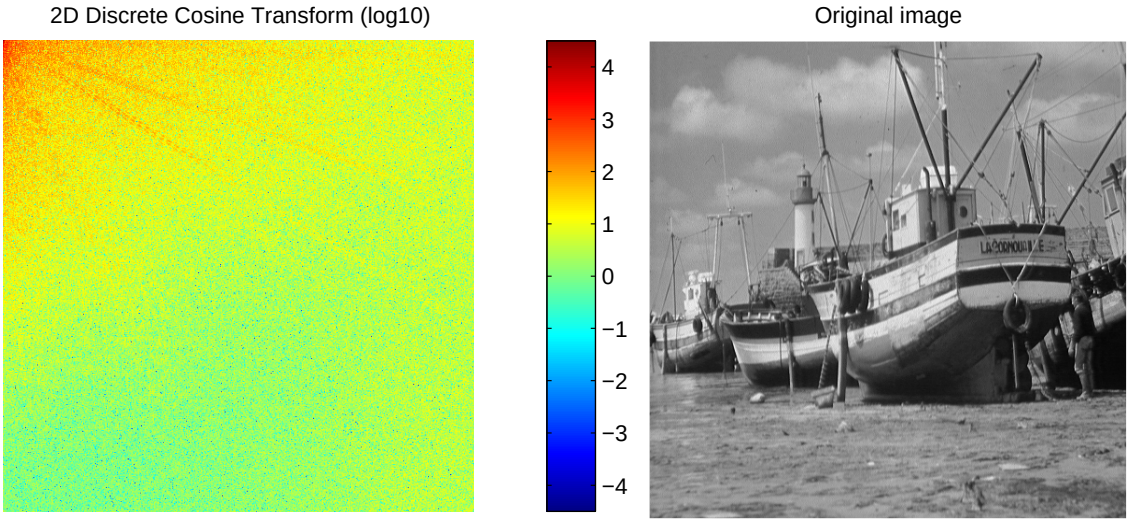
Breakthrough:

Ahmed/Natarajan/Rao discovered the DCT as an excellent approximation of the KLT for typical photographic images, but far more efficient to calculate.

Ahmed, Natarajan, Rao: Discrete Cosine Transform. IEEE Transactions on Computers, Vol. 23, January 1974, pp. 90–93.

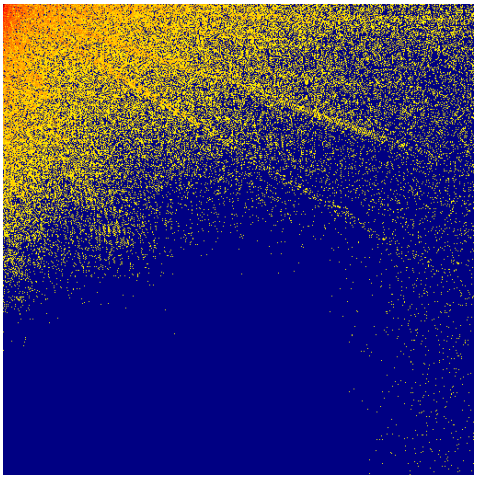
A range of fast algorithms have been found for calculating 1-D and 2-D DCTs (e.g., Ligtenberg/Vetterli).

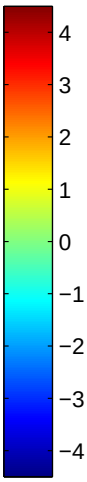
Whole-image DCT



Whole-image DCT, 80% coefficient cutoff

80% truncated 2D DCT (log10)

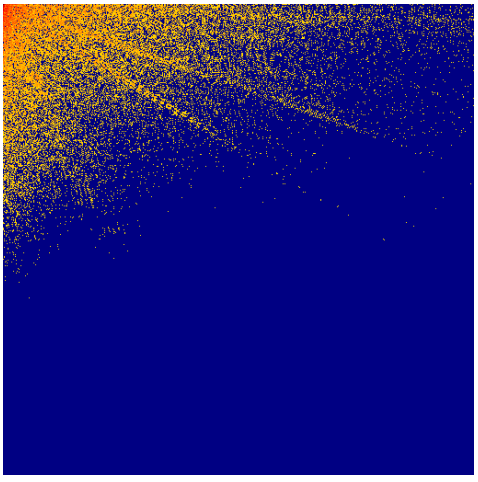


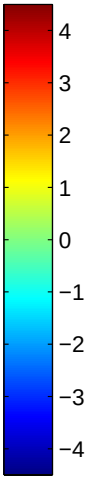


80% truncated DCT: reconstructed image

Whole-image DCT, 90% coefficient cutoff

90% truncated 2D DCT (log10)

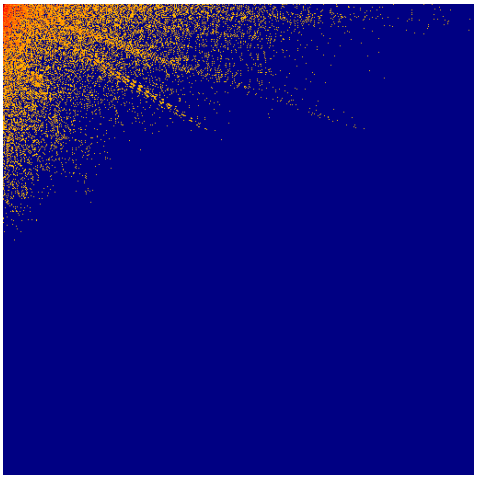


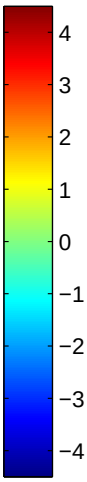


90% truncated DCT: reconstructed image

Whole-image DCT, 95% coefficient cutoff

95% truncated 2D DCT (log10)

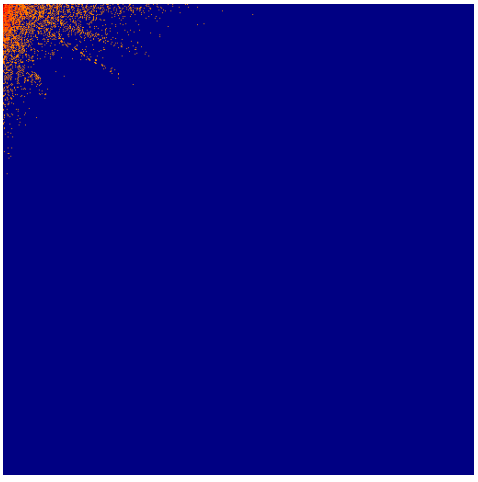


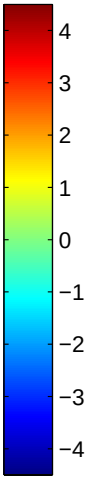


95% truncated DCT: reconstructed image

Whole-image DCT, 99% coefficient cutoff

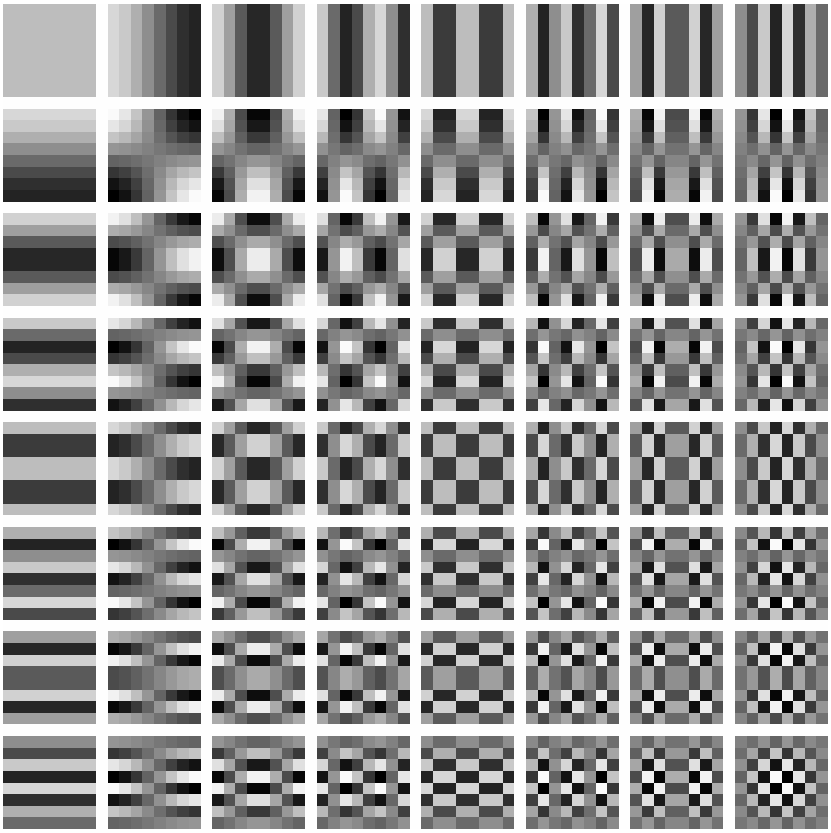
99% truncated 2D DCT (log10)





99% truncated DCT: reconstructed image

Base vectors of 8×8 DCT



Joint Photographic Experts Group – JPEG

Working group "ISO/TC97/SC2/WG8 (Coded representation of picture and audio information)" was set up in 1982 by the International Organization for Standardization.

Goals:

- → continuous tone grayscale and colour images
- → recognizable images at 0.083 bit/pixel
- → useful images at 0.25 bit/pixel
- → excellent images quality at 0.75 bit/pixel
- → indistinguishable images at 2.25 bit/pixel
- → feasibility of 64 kbit/s (ISDN fax) compression with late 1980s hardware at the time (16 MHz Intel 80386).
- → workload equal for compression and decompression

JPEG standard (ISO 10918) was finally published in 1994.

William B. Pennebaker, Joan L. Mitchell: JPEG still image compression standard. Van Nostrad Reinhold, New York, ISBN 0442012721, 1993.

Summary of baseline JPEG algorithm

- ⟶ RGB → YCrCb
- ⟶ reduce CrCb resolution by factor 2
- ⟶ split each of Y, Cr, Cb into 8×8 block
- ⟶ apply 8×8 DCT on each block
- ⟶ apply 8×8 quantisation matrix (divide and round)
- ⟶ apply DPCM coding to DC values
- ⟶ read AC values in zigzag pattern
- ⟶ apply runlength coding
- ⟶ apply Huffmann coding
- ⟶ add standard header with compression parameters

`http://www.jpeg.org/`
Example implementation: `http://www.ijg.org/`

Joint Bilevel Experts Group – JBIG

- ⟶ lossless algorithm for 1–6 bits per pixel
- ⟶ main applications: fax, scanned text documents
- ⟶ context-sensitive arithmetic coding
- ⟶ adaptive context template for better prediction efficiency with rastered photographs (e.g. in newspapers)
- ⟶ support for resolution reduction and progressive coding
- ⟶ "deterministic prediction" avoids redundancy of progr. coding
- ⟶ "typical prediction" codes common cases very efficiently
- ⟶ typical compression factor 20, 1.1–1.5× better than Group 4 fax, about 2× better than "`gzip -9`" and about ≈3–4× better than GIF (all on 300 dpi documents).

Information technology — Coded representation of picture and audio information — progressive bi-level image compression. International Standard ISO 11544:1993.
Example implementation: `http://www.cl.cam.ac.uk/~mgk25/jbigkit/`

Moving Pictures Experts Group – MPEG

- → MPEG-1: Coding of video and audio optimized for 1.5 MBit/s (1× CD-ROM). ISO 11172 (1993).
- → MPEG-2: Adds support for interlaced video scan, optimized for broadcast TV (2–8 Mbit/s) and HDTV, scalability options. Used by DVD and DVB. ISO 13818 (1995).
- → MPEG-4: Enables algorithmic or segmented description of audio-visual objects for very-low bitrate applications. ISO 14496 (2001).
- → System layer multiplexes several audio and video streams, time stamp synchronization, buffer control.
- → Standard defines decoder semantics.
- → Asymmetric workload: Encoder needs significantly more computational power than decoder (for bit-rate adjustment, motion estimation, psychoacousic modeling, etc.)

`http://mpeg.telecomitalialab.com/`

MPEG Video Coding

- → Uses all of YCrCb, 8×8-DCT, quantization, zigzag scan, RLE and Hufmann, just like JPEG (with some improvements such as adaptive quantization).
- → Predictive coding with motion compensation based on 16×16 macro blocks.
- → Three types of frames:
  - I-frames: Encoded independently of other frames
  - P-frame: Encodes difference to previous P- or I-frame
  - B-frame: Interpolates between the two neighboring B- and/or I-frames.

J. Mitchell, W. Pennebaker, Ch. Fogg, D. LeGall: MPEG video compression standard. ISBN 0412087715, 1997.

Audio demo: loudness and masking

loudness.wav

Two sequences of tones with frequencies 40, 63, 100, 160, 250, 400, 630, 1000, 1600, 2500, 4000, 6300, 10000, and 16000 Hz.

- ⟶ Sequence 1: tones have equal amplitude
- ⟶ Sequence 2: tones have roughly equal perceived loudness
  Amplitude adjusted to IEC 60651 "A" weighting curve for soundlevel meters.

masking.wav

Twelve sequences, each with twelve probe-tone pulses and a 1200 Hz masking tone during pulses 5 to 8.

Probing tone frequency and relative masking tone amplitude:

	10 dB	20 dB	30 dB	40 dB
1300 Hz				
1900 Hz				
700 Hz				

MPEG audio coding

Waveforms sampled with 32, 44.1 or 48 kHz are split into segments of 384 samples. Three alternative encoders of different complexity can the be applied.

- ⟶ Layer I: Each segment is passed through an orthogonal filterbank that splits the signal into 32 subbands, each 750 Hz wide (for 48 kHz). Each subband is then sampled with 1.5 kHz (12 samples per window). This is followed by uniform quantization based on a psychoacoustic model.
- ⟶ Layer II: Adds better encoding of scale factors and bit allocation.
- ⟶ Layer III ("MP3"): Adds modified DCT step to decompose subbands further into 18 frequency lines, non-uniform quantisation, Huffman entropy coding, buffer with short-term variable bitrate, dynamic window switching (to enable control or preechos before sharp percussive sounds), joint stereo processing

Psychoacoustic models

MPEG audio encoders use a psychoacoustic model to estimate the spectral and temporal masking that the human ear will apply. The subband quantisation levels are selected such that the quantisation noise remains in each subband below the masking threshold.

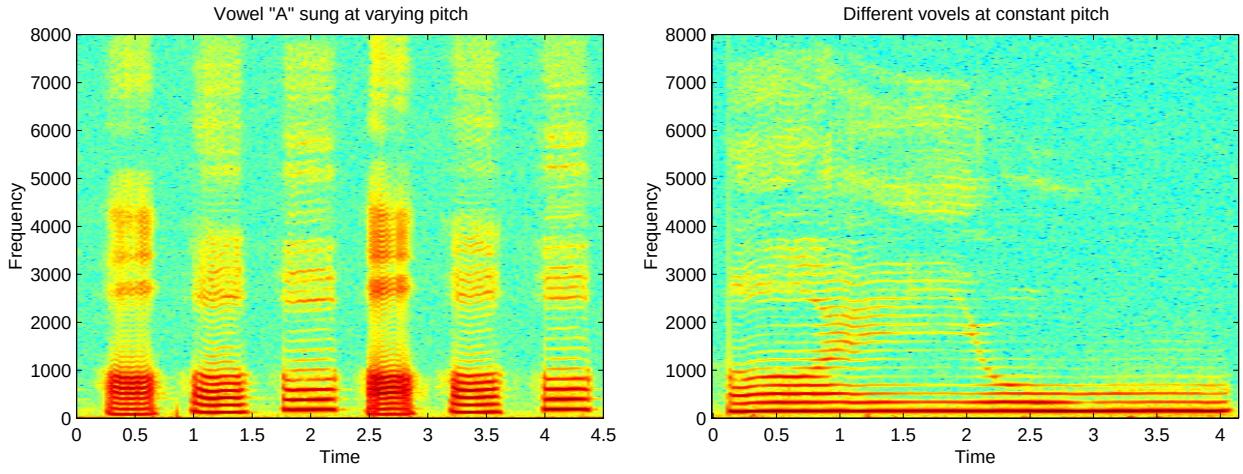
The masking model is not standardised and each encoder developer can chose a different one. The steps typically involved are:

- ⟶ Fourier transform for spectral analysis
- ⟶ Group the resulting frequencies into "critical bands" within which masking effects will not differ significantly
- ⟶ Distinguish tonal and non-tonal (noise-like) components
- ⟶ Apply masking function
- ⟶ Calculate threshold per subband
- ⟶ Calculate signal-to-mask ratio (SMR) for each subband

Masking is not linear and can be estimated accurately only if the actual sound pressure levels reaching the ear are known. Encoder operators usually cannot know the sound pressure level selected by the decoder user. Therefore the model must use worst-case SMRs.

Voice encoding

The human vocal tract can be modeled as a variable-frequency impulse source (used for vowels) and a noise source (used for fricatives and plosives), to which a variable linear filter is applied which is shaped by mouth and tongue.



Vector quantisation

A multi-dimensional signal space can be encoded by splitting it into a finite number of volumes. Each volume is then assigned a single codeword to represent all values in it.

Example: The colour-lookup-table file format GIF requires the compressor to map RGB pixel values using vector quantization to 8-bit code words, which are then entropy coded.

Literature

References used in the preparation of this part of the course in addition to those quoted previously:

- D. Salomon: A guide to data compression standards. ISBN 0387952608, 2002.
- A.M. Kondoz: Digital speech – Coding for low bit rate communications systems. ISBN 047195064.
- L. Gulick, G. Gescheider, R. Frisina: Hearing. ISBN 0195043073, 1989.
- H. Schiffman: Sensation and perception. ISBN 0471082082, 1982.
- British Standard BS EN 60651: Sound level meters. 1994.

Exercise 1 Compare the quantization techniques used in the digital telephone network and in audio compact disks. Which factors to you think led to the choice of different techniques and parameters here?

Exercise 2 Which steps of the JPEG (DCT baseline) algorithm cause a loss of information? Distinguish between accidental loss due to rounding errors and information that is removed for a purpose.

Exercise 3 How can you rotate/mirror an already compressed JPEG image without loosing any further information. Why might the resulting JPEG file not have the exact same filelength?

Exercise 4 Decompress this G3-fax encoded pixel sequence, which starts with a white-pixel count: 11010010111101111011000011011100110100

Exercise 5 You adjust the volume of your 16-bit linearly quantising soundcard, such that you can just about hear a 1 kHz sine wave with a peak amplitude of 200. What peak amplitude do you expect will a 90 Hz sine wave need to have, to appear equally loud (assuming ideal headphones)?