I.3 Media Type Audio

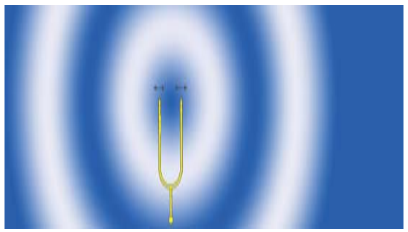
- ❑ Grundlagen der Akustik
- ❑ Digitalisierung
- ❑ Formate
- ❑ Operationen
- ❑ MIDI (Musical Instruments Digital Interface)

Grundlagen der Akustik

- ❑ Was ist Schall?
- ❑ Wie wird er mathematisch beschrieben?
- ❑ Eigenschaften (Amp., Frequenz) Auswirkung aufs Ohr?
- ❑ Ton, Klang, Geräusch
- ❑ Frequenzspektrum (diskrete und kontinuierliche Fourierzerlegung)
- ❑ Schalldruckpegel
- ❑ Schallpegeladdition
- ❑ Kurven gleicher Lautstärke
- ❑ Lautstärkepegel

Was ist Schall?

- ❑ Physikalische Definition: Schall ist die wellenförmige Ausbreitung von Druckschwankungen in elastischen Medien (d.h. Bindungskräfte zwischen Molekülen oder Atomen sind elastisch, wie z.B. in der Luft, Wasser,Knochen, Holz, Metall...). Dabei findet periodisches Komprimieren und Dekomprimieren des Mediums statt.



Stimmgabel überträgt mechanische Schwingung auf angrenzende Luftmoleküle:

Kompression↔Dekompression

Druckschwankungen werden an benachbarte Moleküle weitergegeben ⇒ Schallwelle

Wie wird er mathematisch beschrieben?1

- ❑ Da Schall auf Schwingungen beruht, lassen sich Schallereignisse durch ihren zeitlichen Schwingungsverlauf beschreiben. Einfachste Schwingung: Sinusschwingung (alle anderen Schwingungen lassen sich ableiten):
- ❑ **Frequenz f:** Anzahl der Schwingungen pro Sekunde
  Einheit : Hertz [Hz]
- ❑ **Periodendauer τ:** Dauer einer Schwingung $\tau = 1/f$
  Einheit: Sekunde [s]
- ❑ **Amplitude p_0:** Druckwert bei maximaler Kompression
  Einheit: Pascal... [Pa]

Wie wird er mathematisch beschrieben? 2

- **Wellenlänge λ:** der während einer Periodendauer zurückgelegte Weg. Einheit:..[m]

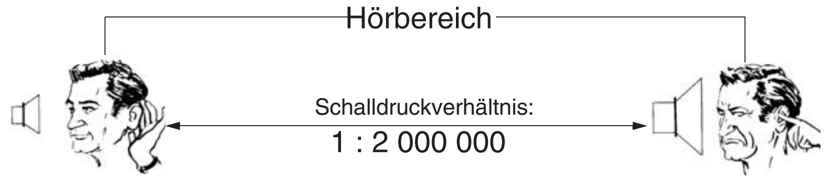
$$\lambda=\nu \times \tau=\nu/f$$

- **Schallgeschwindigkeit** ν: ist abhängig vom Ausbreitungs-medium: Einheit: [m/s]

Medium	ν[m/s]	λ [m] 16Hz	λ [m] 100Hz	λ [m] 4kHz	λ [m] 20kHz
Luft (0°)	331,0	20,68	3,31	0.08	0,0166
Luft (20°)	343,6	21,48	3,44	0,09	0,0172
Wasser (20°)	1484	92,75	14,84	0,37	0,0742
Ziegel	3650	228,13	36,50	0,91	0,1825

Eigenschaften und deren Wirkung 1

- **Frequenz:** ein Maß für die Tonhöhe. Je höher Frequenz, desto höher der Ton. Musikalisch entspricht eine Verdopplung der Frequenz einer Tonerhöhung von einer Oktav.
  hörbare Frequenzen: ca. 16Hz - 20000Hz
- **Amplitude der Druckwelle:** ein Maß für die Lautstärke.



Hörschwelle: $p_0=10^{-5}$ [Pa]
Ton gerade noch hörbar

Schmerzgrenze:p=20[Pa]
physiologischer Schmerz

Eigenschaften und deren Wirkung 2

Wellenlänge: Ist nicht direkt hörbar, aber spielt bei folgenden Hörphänomenen eine wichtige Rolle:

- Beugung: Wellen weichen von ihrem geradlinigen Weg ab und "lugen" um die Ecke. Verstärkt bemerkbar, wenn Wellenlänge größer als Hindernis.("um die Ecke hören")
- Interferenz: Überlagerung der direkten Schallwelle und indirekten Wellen, die durch Beugung und Reflexion entstanden sind. Interferenzeffekte, die im Bereich unseres Kopfes stattfinden, lassen uns unterscheiden, ob Schall von vorne oder hinten, oben oder unten kommt.

Eigenschaften und deren Wirkung 3

- Brechung: Schallwellen ändern ihre Richtung, wenn sie durch Gebiete mit unterschiedlicher Schallgeschwindigkeit laufen.
- Dispersion: Für verschiedene Tonhöhen ist die Stärke der Brechung verschieden. Dieser Effekt heißt Dispersion.

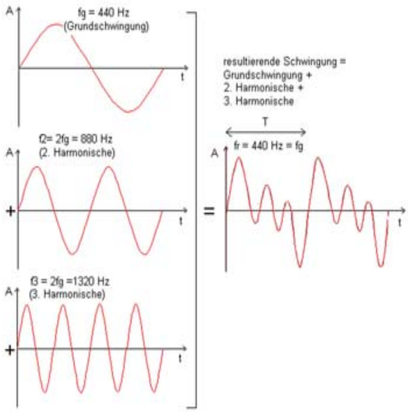
Ton, Klang, Geräusch

- Ton:

Eine einzige Sinusschwingung wird als Ton bezeichnet. p(t) ist sinusfunktion

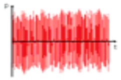
- Klang:

Klang entsteht aus einer Überlagerung von Grundton(hören wir alsTonhöhe) und Obertönen (f=n*fg) (hören wir als Klangfarbe). p(t)-Funktion allgemeine periodische Funktion mit Frequenz fg.



- Geräusch:

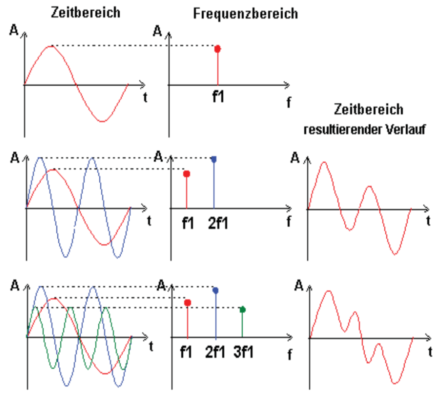
Kein Grundton mehr erkennbar, aperiodische p(t) Funktion



Frequenzspektrum 1/2 [1,6]

- Frequenzspektrum:

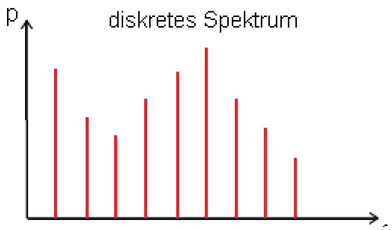
Jede periodische Schwingung beliebiger Wellenform (= Klang) kann als eine Überlagerung von Grundschwingung und Oberschwingungen dargestellt werden.Trägt man in einem Diagramm die Amplituden aller beteiligten Schwingungen in Abhängigkeit der Frequenz auf, so erhält man das sogenannte Frequenzspektrum



Frequenzspektrum 2/2

- Diskretes Spektrum:

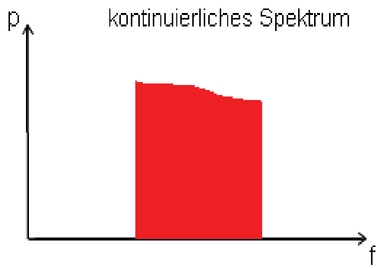
Klänge enthalten nur Schwingungen, deren Frequenzen ein ganzzahliges Verhältnis zur Grundfrequenz aufweisen.



- Kontinuierliches Spektrum:

Allgemeine Schallereignisse (Geräusche) enthalten eine unendliche Anzahl von Einzelschwingungen, deren Frequenzwerte sich kontinuierlich über die x-Achse erstrecken. Es entsteht eine kontinuierliche mathematischeFunktion:

p=p(f)



Schalldruckpegel

- **Definition des Schalldruckpegels:**

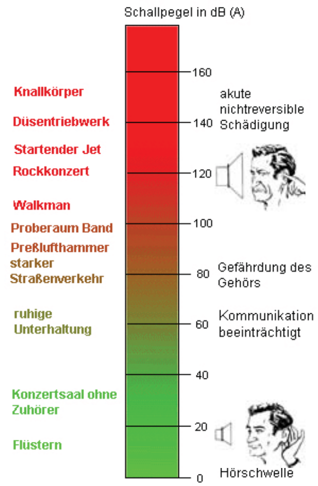
$L=20*\log(p/p_0)$ [dB]

L...Schalldruckpegel

p...Schalldruck der betrachteten Schallwelle

p_0...Bezugsschalldruck: definitionsgemäß der Schalldruck eines 1000Hz Tones, der gerade noch hörbar ist. $p_0=2*10^{-5}$[Pa]

dB...Dezibel, Einheit des Schallpegels



Wozu Schalldruckpegel?

- ❑ Damit man den gigantischen Hörbereich (Hörschwelle / Schmerzgrenze 1 : 2 000 000!!) sinnvoll beschreiben und darstellen kann, Abbildung in einen kleineren Wertebereich (Hörschwelle=0dB,Schmerzgrenze= 120dB)
- ❑ Experimente zeigen, daß unser Empfinden von Lautstärkedifferenzen einem logarithmischen Gesetz folgen
- ❑ der logarithmische Schalldruckpegelwert wird daher unserem Lautstärkeempfinden mehr gerecht als die Angabe eines absoluten Druckamplitudenwertes(*siehe Schallpegeladdition)*.

Schallpegeladdition

Anzahl Schallquellen	 x1	 x2	 x10	 x100
Schalldruck-änderung	x1	x1,4	x3	x10
Schallpegel-differrenz	0dB	+3dB	+10dB	+20dB
Subjektives Hörempfind-en	Grund-lautstärke	etwas lauter	doppelt so laut wie eine einzige Geige	viermal so laut wie eine einzige Geige

Schallpegeladdition

Addition von Schallpegeln gleicher Intensität

- ❑ Überlagern sich die Schallintensitäten von n Schallpegeln mit gleicher Schallintensität, so gilt für die am Immissionsort gemessene Schallintensität

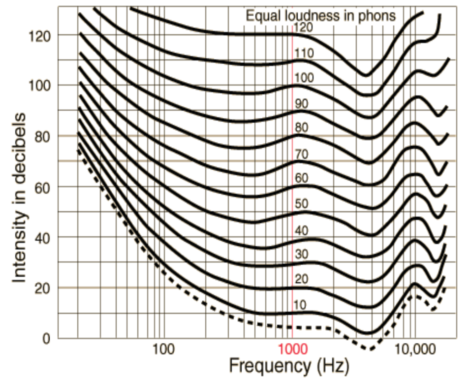
$$I_{ges} = I_1 + I_2 + I_3 + ... + I_n = n*I_1$$

- ❑ Für den Summenpegel gilt:

$$L_{ges} = 10*\log(I_{ges}/I_0) = 10*\log(I_1/I_0) + 10*\log n$$

Kurven gleicher Lautstärken

- ❑ Töne mit verschiedenen Frequenzen, aber selbem **L** beurteilt unser Ohr verschieden laut. Die Kurven gleicher Lautstärke geben an, wie hoch im Vergleich zum Pegel eines 1000Hz-Vergleichtons der Schallpegel eines Tones variabler Frequenz sein muß, damit dieser Ton als gleichlaut empfunden wird.



Lautstärkepegel

- Damit alle Töne, die für unser Ohr gleich laut klingen, auch mit einem gleichen Pegelwert beschrieben werden, wurde der **Lautstärkepegel L_N** definiert: Einheit: [Phon]

  L_N eines Tones mit der Frequenz f ist gegeben durch den Schalldruckpegel L eines Tones mit 1kHz, der als gleich laut gehört wird.
- Den Lautstärkepegel eines beliebigen Tones kann man leicht aus den Kurven gleicher Lautstärke herauslesen:
- Bsp: Ton mit 100Hz und Schalldruckpegel *L*=60dB: 50*Phon*

Einführung Digitales Audio

Mikrofon wandelt Schallwelle in elektrisches Signal:
Spannungsverlauf analog dem Druckschwankungsverlauf
an der Membran des Mikrofons.

- Analoge Audiotechnik:
  analoges Signal gespeichert auf Magnetband, Schallplatte...
  Editieren: mechanisches Schneiden, überspielen
- Digitale Audiotechnik:
  analoges Signal wird mittels A/D Konverter in binäre Darstellung umgewandelt, gespeichert, am Computer editiert; mittels D/A Konverter wieder als Analogsignal am Lautsprecher ausgegeben.

Vorteile / Nachteile DA

Vorteile *(digital)*:

- weniger Rauschen
- höhere Dynamik
- Verlustloses Kopieren
- Höhere Linearität: Frequenzgang, Übertragungskurve (Klirrfaktor)
- Temperatur unempfindlich
- keine Gleichlaufschwankungen

Nachteile *(digital)*:

- anfällig für Datenverlust
- höhere Übertragungsbandbreiten
- aufwendigere Hardware

Digitization in General

- microphones, video cameras produce analog signals (continuous-valued voltages)
- to get audio or video into a computer, it must be converted into a stream of numbers; *discrete sampling* (both time and voltage)
- sampling—divide the horizontal axis (the time dimension) into discrete pieces; uniform sampling is ubiquitous
- quantization—divide the vertical axis (signal amplitude) into pieces; sometimes, a non-linear function is applied

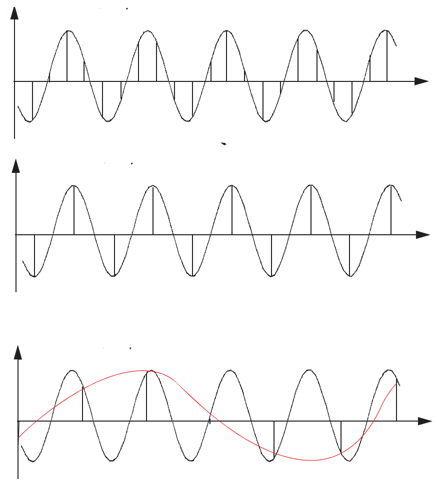
Digitization in General

questions for producing digital audio (Analog-to-Digital Conversion):

- ❑ how often do you need to sample the signal?
- ❑ how good is the signal?
- ❑ how is audio data formatted?

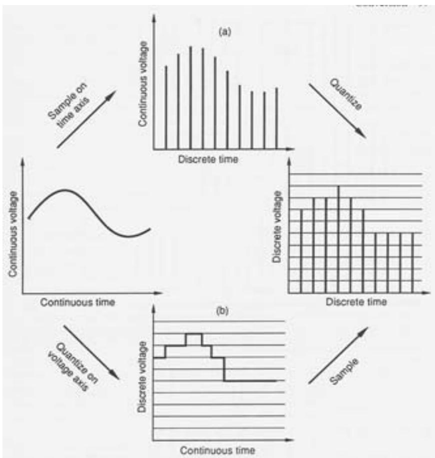
- ❑ suppose we are sampling a sine wave. How often do we need to sample it to figure out its frequency?
- ❑ if we sample at 1 time per cycle, we can think it's a constant
- ❑ if we sample at 1.5 times per cycle, we can think it's a lower frequency sine wave --> alias

Aliaskomponente



Grundlagen Digitalisierung

- ❑ Eingang: zeit- und spannungskontinuierliche Wellenforml
- ❑ Abtasten- Quantisieren: sind voneinander unabhängig $\rightarrow$ zeitliche Reihenfolge egal
- ❑ Ergebnis: zeit- und spannungsdiskretes Format



(a) Eingangssignal wird abgetastet, Samples werden quantisiert (üblich bei Audio)

(b) Eingangssignal wird quantisiert, dann erst abgetastet (üblich bei Video)

Abtasten

- ❑ Prozedur: in periodischen Zeitintervalen wird dem analogen Eingangssignal eine Probe (=sample) entnommen.
- ❑ Wie oft muß ein analoges Signal abgetastet werden?

Nyquist-Shannon Theorem:
Ein abgetastetes Signal läßt sich nur dann ohne Informationsverlust rekonstruieren, wenn

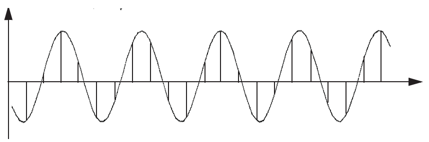
$$f_s \geq 2\ x\ f_{max}$$

f_s .....Abtastfrequenz

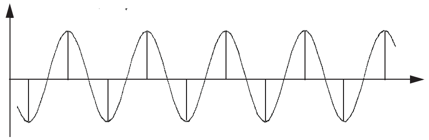
f_{max}...die höchste im Signal vorkommende Frequenz

Nyquist Rate, Nyquist Frequenz

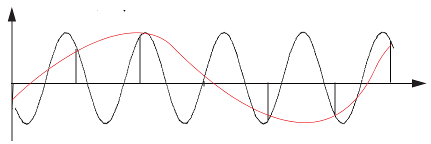
Aliaskomponente 1



- $f_s > 2 \, x \, f_{max}$
  Rekonstruiertes Signal mit ursprünglichem ident



- $f_s = 2 \, x \, f_{max}$
  Rekonstruiertes Signal gleiches f wie ursprünglich

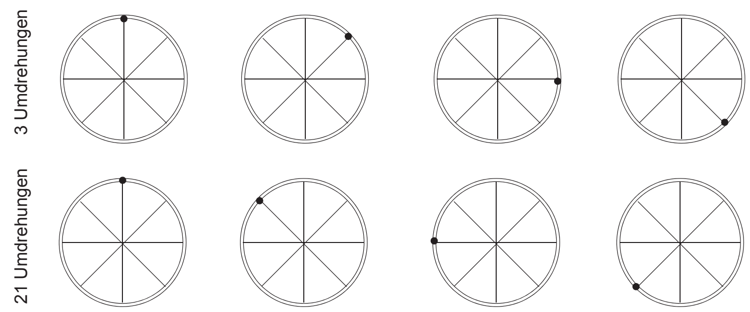


- $f_s < 2 \, x \, f_{max}$
  Störfrequenz "Aliaskomponente"

Bsp.: $f_s = 48kHz$, $f_{ein} = 30kHz \rightarrow f_{alias} = 18kHz$

Aliaskomponente 2

- Bsp: drehendes Rad im Film
  - Sampling Frequenz ist dabei fix (FS = 24)
  - Anzahl der Umdrehungen (u) des Rades variabel



Beispiele

Abtastfrequenz und max. Frequenz

Format	Abtastfrequenz	Frequenzbereich
Telefon	8 kHz	200-3400 Hz
Audio- CD	44,1 kHz	20-20000 Hz
DAT, prof. Audio	48 kHz	20-20000 Hz
Sat- Radio	32 kHz	20-15000 Hz

Quantisierung

- Abgetasteten Spannungswerten werden diskrete Zahlenwerte zugeordnet: Gesamtspannungsbereich in Quantisierungsintervale Q unterteilt $\rightarrow$ kontinuierlicher Wert nächstgelegenem Zahlenwert zugeordnet.
- Qualität: Q soll möglichst klein sein, ergibt sich aus der Länge des Datenworts:

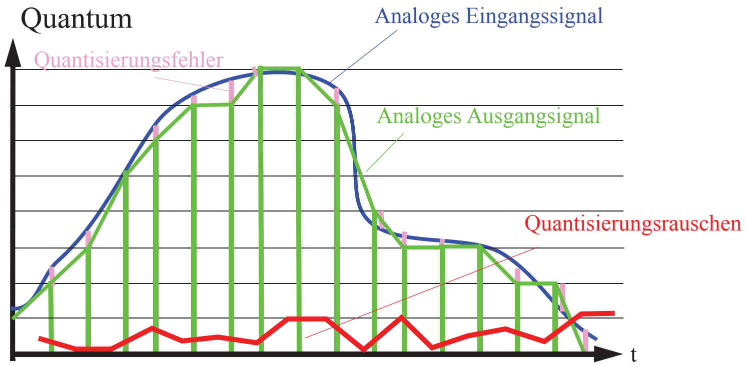
  8 bit *(256 Intervale)*: $Q = V_{pp} / 2^8 = (4x10^{-3}) \, x \, V_{pp}$

  16 bit *(65536 Intervale)*: $Q = V_{pp} / 2^{16} = (15x10^{-6}) \, x \, V_{pp}$

  mit $V_{pp} = 2 \, x \, abs(V_{max})$ V_{pp} *... peak to peak Bereich*

Quantisierungsfehler

- Zahlenwert weicht vom abgetasteten Wert um $\leq 0{,}5\ Q$ ab. Bei Rekonstruktion Fehler als "Quantisierungsrauschen" hörbar.



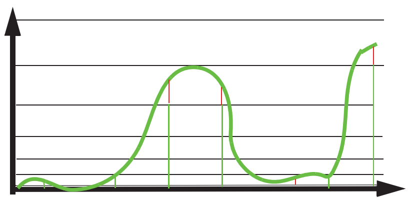
Dynamikbereich

- Schallwiedergabe:
  Quantisierungsrauschen erzeugt Schalldruckpegel $\mathbf{L_{noise}}$.
  Maximales Nutzsignal Schalldruckpegel $\mathbf{L_{max}}$.
- Dynamikbereich:
  SNR = $\mathbf{L_{max}} - \mathbf{L_{noise}}$
- **SNR**..Signal to noise ratio [dB]

Auflösung	8 bit	12 bit	14 bit	16 bit
Dynamik	49,8 dB	73,7 dB	85,7 dB	97,6 dB

Lineare, nicht lineare Quantisierung

- Lineare Quantisierung: Q konstant; übliche Methode in Audiotechnik
- Nichtlineare Quantisierung: *Q von unterschiedlicher Größe.*
  kleine Werte → kleiner Quantisierungsfehler
  große Werte → großer Quantisierungsfehler
- Rauschen von großem Nutzsignal maskiert; ⇒ auch mit kleiner Auflösung passable Qualität
  Verwendung: für Systeme mit geringer Übertragungsbandbreite

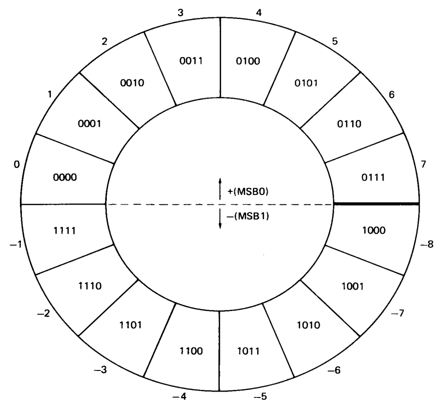


Codierung

Signale von Audio bipolar (pos. und neg.) ⇒

Zweierkomplementcode:

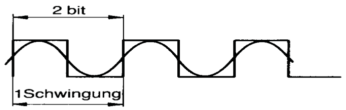
- MSB=1...neg. Wert
  MSB=0...pos. Wert
- bei Addition zweier Signale kein Offset (=Nullpunktverschiebung)



Bsp.: Zweierkomplementcode für 4-bit Daten

Pulscodemodulation

- Prozedur: Binärwerte werden seriell in Form von modulierten Spannungspulsen über eine einzige Leitung übertragen.
- Datenrate:
  Abtastfrequenz[Hz] x Bitauflösung[bit] = Datenrate[bit/s]
  Bsp.: CD-Audio(mono): 44,1kHz;16 bit ⇒ 705600 bit/s
- Bandbreite: nötiger Frequenz-bereich, um PCM verlustlos zu übertragen:



  Bandbreite[Hz] = Datenrate[bit/s] : 2
  Bsp.: CD-Audio(mono) ⇒ 352,8 kHz (vgl.: analog 20 kHz !!)

Characteristics of Digital Audio

- sampling frequency (rate)
- sample size / quantization
- number of channels (tracks)
- interleaving
- sample representation (negative values)
- coding/compression

Example Audio Formats

	DAT	CD audio	CD-I (Level B)	A-law[a] µ-law
sampling rate (kHz)	48[b]	44.1	37.8	8
sample size (bits)	16	16	4	8
quantization	uniform	uniform	uniform	log
no. of channels	2	2	1-8	1
data rate per channel (10^3 bit/sec)	768	705	174	64
encoding	PCM	PCM	ADPCM	PCM
quality	very high	very high	“Mid Fi”, “FM”	“telephone”

a. A-law is used within European telephone systems, while µ-law is North American.
b. DAT has a number of audio formats – three sampling frequencies are possible (32 kHz, 44.1 kHz and 48 kHz) and the 32 kHz rate may use either 16 bit linear quantization or 12 bit nonlinear, in the second case two or four tracks may be recorded. The numbers listed are for the highest quality DAT format.

Operations on Digital Audio

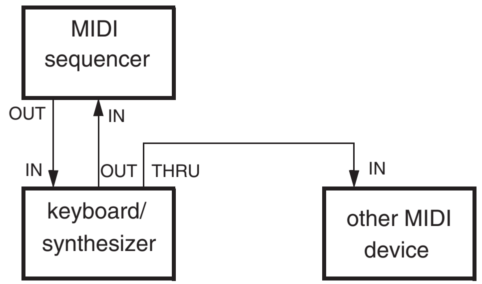
- storage and retrieval (CD, DAT)
- editing (non-linear, non-destructive, play-lists)
- digital audio effects (delay, equalization, noise reduction, time compression and expansion, pitch shifting ...)
- conversion

Audio Processing—Examples

- reshape impulse response to simulate a different room
- move perceived location from which sound comes
- locate speaker in 3D space using microphone arrays
- cover missing samples
- mix multiple signals (i.e. conference)
- echo cancellation

Musical Instruments Digital Interface

example MIDI system configuration



MIDI Terminology

- MIDI—a protocol that enables computer, synthesizers, keyboards, and other musical device to communicate with each other
- synthesizer—a sound generator (various pitch, loudness, tone color); often has a microprocessor, keyboard, control panels, memory, etc.
- sequencer—a stand-alone unit or a software program; used to be a storage for MIDI data; nowadays more a software music editor; has one or more MIDI INs and MIDI OUTs.

MIDI Terminology

- track—used to organize the recordings; can be turned on or off on recording or playing back.
- channel—used to separate information in a MIDI system; there are 16 MIDI channels in one cable; channel numbers are coded into each MIDI message.
- voice—the portion of the synthesizer that produces sound; synthesizers can have many (16, 20, 24, 32, 64, etc.) voices; each voice works independently and simultaneously to produce sounds of different timbre and pitch.

MIDI Terminology

- key number — notes are identified by 128 key numbers
- controller — controller values specify the operational characteristics of a MIDI device
- patch / program — the control settings that define a particular timbre

Important MIDI Concepts

- timing clocks — MIDI sequencer time stamps messages, timebase measured in *parts per quarter note,* tempo in *beats per minute*
- MIDI synchronization — external or internal sync
- MIDI Time Code (MTC) — used to synchronize with film or video

Part II Compression

- Requirements
- Classification
- Examples
- Entropy and Source Coding
- JPEG
- MPEG

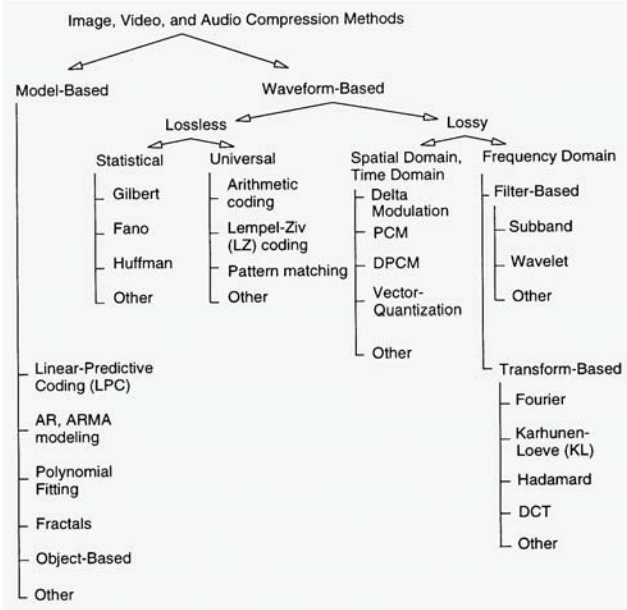
Compression Requirements

- for dialog mode applications
  - end-to-end delay (EED) should not exceed 150 ms
  - face-to-face application needs EED of 50 ms (incl. comp.)
- retrieval mode applications
  - fast forward and backward data retrieval with simultaneous display
  - random access to single images and audio samples, access time < 0.5 s
- decompression without a link to other data units—allows random access and editing

Compression Requirements

- for dialog and retrieval mode applications
  - support scalable video in different systems
  - support of various audio and data rates
  - synchronization of audio and video
  - economical solutions

Compression Taxonomy



Compression Techniques-Classification

entropy coding	run-length coding	
	Huffman coding	
	arithmetic coding	
source coding	prediction	DPCM
		DM
	transformation	FFT
		DCT
	vector quantization	
hybrid coding	JPEG	
	MPEG	
	px64	
	DVI (RTV, PLV)	

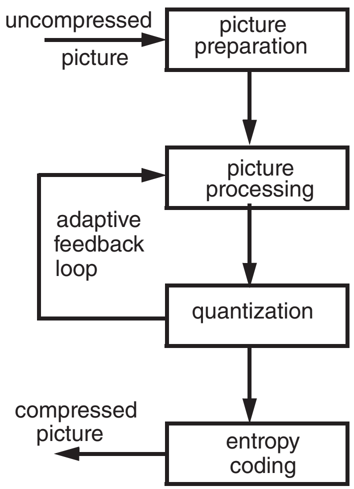
Classification (continued)

- entropy coding
  - lossless encoding
  - used regardless of media’s specific characteristics
  - data taken as simple digital sequence
- source coding
  - lossy encoding
  - takes into account the data semantics
  - degree of compression depends on data content

Decompression Requirements

- dialog mode applications need symmetric compression
- retrieval mode applications need asymmetric compressions
  - compression is performed once and sample time available
  - decompression is performed frequently and needs to be done fast

Compression Steps



- analog-to-digital conversion
- generation of appropriate digital representation
- image division into 8 x 8 blocks
- fix number of bits/pixel

Compression Algorithm

- transformation from time to frequency domain e.g., using DCT
- motion vector application for motion video

- mapping of real numbers to integers (reduction in precision)
  example: μ-law audio—from 12 bits to 8 bit quantization

- compress a sequential digital stream without loss

Run-length Encoding

- entropy coding algorithm—content dependent coding
- replaces the sequence of the same consecutive bytes with the number of occurrences
- the number of occurrences is indicated by a *special flag (!)*
- procedure:
  - if the same byte occurs at least 4 times count number of occurrences
  - write compressed data in the format
    the counted byte '!' number of its occurrences

Run-length Encoding

- example:
  - uncompressed sequence (20 Bytes):
    ABCCCCCCCCCDEFFFFGGG
  - compressed sequence (13 Bytes):
    ABC!9DEF!4GGG
- variations
  - zero suppression technique
  - text compression technique (also for images, video. audio)
  - diatomic encoding

Statistical Encoding

- frequency dependent encoding—belongs to entropy encoding
- given is a sequence of symbols: S_1, S_2, S_3. .... and the probability of occurrence of each symbol $p(S_i)=p_i$
- Ex.: p(A)=0.16, p(B)=0.51, p(C)=0.09, p(D)=0.13, p(E)=0.11
  first choice: encode A,B,C,D,E as 000,001,010,011,100
- Q: What is the minimum number of bits per symbol?
- A: The theoretical minimum average number of bits per code word is known as *entropy* (H), according to Shannon
  $H = - \Sigma\, p_i \log_2 p_i$ bits per code word (Ex: 1.36 bits/symbol)

Huffman Encoding

- statistical encoding, depends on occurrence frequency of single characters or sequences of data bytes
- characters stored with their probabilities
- length (number of bits) of the coded characters differs
- shortest code assigned to the most frequently occurring character
- to determine Code, it is useful to construct a binary tree
  - leaves are characters to be encoded
  - nodes carry occurrence probabilities of the characters belonging to the subtree

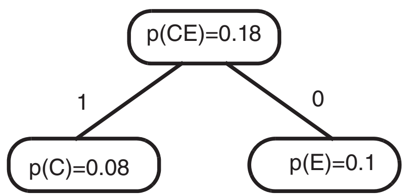
Huffman Encoding—Example

build a Huffman Tree:
p(A)=0.17, p(B)=0.51, p(C)=0.08, p(D)=0.14, p(E)=0.1

```
build ordered list of symbols (increasing probability)
do while list contains at least 2 elements
   construct tree using the first two elements in list
   add parent node for the union of these elements and
      compute probability
   mark edges by '0' and '1'
   delete the first 2 elements in list; insert parent into list
end
```

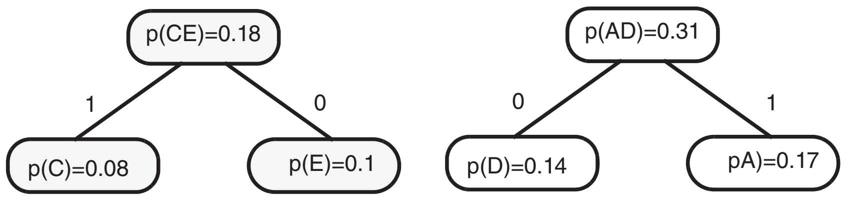

Huffman Encoding—Example

step 1: p(C)=0.08, p(E)=0.1, p(D)=0.14, p(A)=0.17, p(B)=0.51



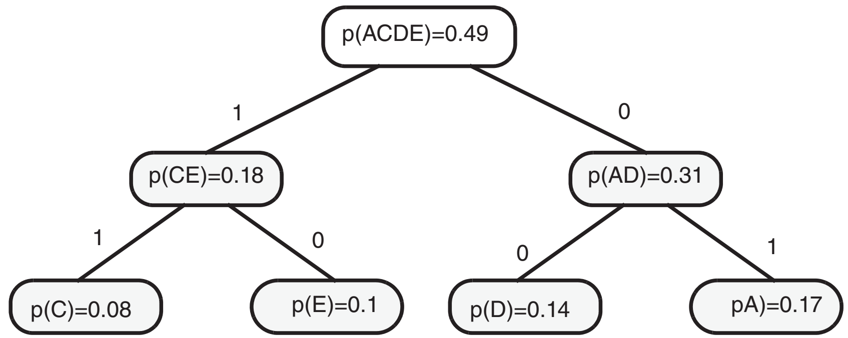
Huffman Encoding—Example

step 2: p(D)=0.14, p(A)=0.17, p(CE)=0.18, p(B)=0.51



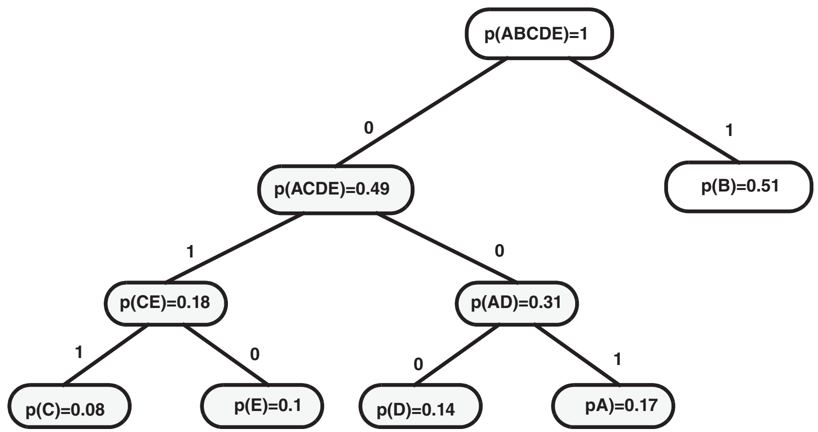
Huffman Encoding—Example

step 3: p(CE)=0.18, p(AD)=0.31, p(B)=0.51



Huffman Encoding—Example

last step: p(ACDE)=0.49, p(B)=0.51



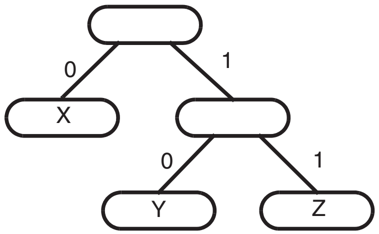
Huffman Encoding—Example

encode symbols according to tree

A	001
B	1
C	011
D	000
E	010

Huffman Decoding—Example

given: 000101011 and Huffman Tree



reconstruct message from bit stream using finite state machine

Application of Huffman Coding for Video

- if the image information can be transformed into a bit stream, Huffman table can be used to compress data without loss
- simple way to generate a bit stream is to code pixels individually and read them line by line
- for video: Huffman table can be used for a single sequence of images
    - for a set of scenes
    - for an entire film clip
- need Huffman table for encoding and decoding

Differential Encoding

- belongs to source coding
- consider sequence of symbols S_1, S_2, S_3, ... where values are not zeros, but do not vary much; calculate difference from previous value
- Ex.: still images—calculation of differences between nearby pixels or pixel groups

0	0	0	0	0
0	255	94	87	100
0	0	0	0	255
0	0	0	0	0

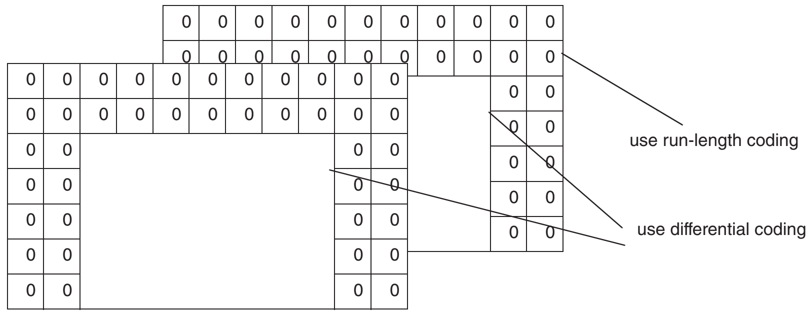
edges result in large values

similar chrominance or luminance values result in small values

- zeros can be suppressed by run-length coding

Differential Coding in Video (Examples)

- static background (videoconferencing, newscast, etc.)



- motion compensation—8x8 blocks are compared; areas similar, only shifted, e.g., to the right (*motion vector*)

Joint Photographic Experts Group (JPEG)

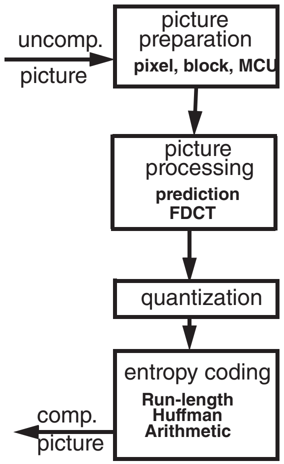
requirements:

- ❑ to be near the state-of-the-art for degree of compression versus image quality,
- ❑ to be parameterizable (user can select parameters),
- ❑ to be applicable to any kind of source image, without regard to dimensions, content, image aspect ratio, pixel aspect ratio, etc.,
- ❑ to have computational requirements that are reasonable for both, hw and sw implementations,
- ❑ to run on as many standard platforms as possible

Joint Photographic Experts Group (JPEG)

- ❑ to support four different modes of operation:
  - ❑ sequential encoding—encoded in same order as scanned
  - ❑ progressive encoding—multiple pass encoding
  - ❑ lossless encoding
  - ❑ hierarchical encoding—encoded at multiple resolutions

JPEG Processing Steps



Sequential Encoding	Progressive Encoding	Lossless Encoding	Hierarchical Encoding
block, MCU 8-bit/pixel	12 bit/pixel	2-12 bit/pixel	
Transformation Source Coding lossy DCT	Layered Coding	Predictive Entropy Coding	switch between lossy DCT and loss-less technique
Run-length Huffman			

JPEG General Image Model

- ❑ not based on
  - ❑ 9-bit YUV coding
  - ❑ fixed number of lines, columns
  - ❑ mapping of encoded chrominance
- ❑ independence from image parameters
- ❑ source image consists of 1 to 255 components (planes)
- ❑ all pixels of all components within the same image are coded with the same number of bits

JPEG Image Preparation

- images divided in data units (blocks), DCT operates on blocks
- lossy mode operates on 8x8 pixel blocks
  lossless mode operates on data units of 1 pixel
- in most cases data units are processed component by component and passed to image processing
- processing order of data units
  - left-to-right, top-to-bottom
  - interleaved data ordering
- interleaved data units of different components are combined to *minimum coded units* (MCU)

JPEG Image Preparation

after image preparation step:

- uncompressed image samples are grouped into data units of 8x8 pixels and passed to JPEG encoder
- order of data units is defined by MCUs
- values in the range [0, 255]

JPEG Image Processing

first step:

- pixel values shifted into the range [-128, 127]
- values in the 8x8 pixel blocks are defined by S_{yx}, y,x ε [0,7]
- Forward Discrete Cosine Transformation (Forward DCT) maps values from time to frequency domain

$$svu = \frac{1}{4} cucv \sum_{x=0}^{7} \sum_{y=0}^{7} syx \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

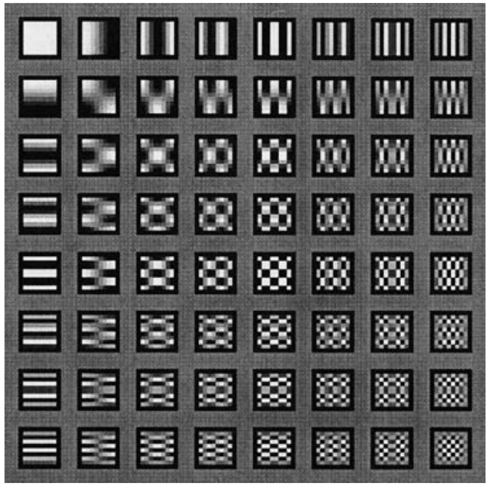
with c_u, $c_v = 1/\sqrt{2}$ for u,v=0; otherwise c_u, $c_v = 1$

JPEG Image Processing (DCT)

S(u,v) coefficients

- S(0,0)—lowest frequency in both directions is called DC coefficient
  - determines the fundamental color of the block
  - frequency = 0 in both directions
- S(0,1), ..., S(7,7) are called AC coefficients
  - frequency in one or both directions non-zero
- computing of DCT—use factoring

DCT basis functions



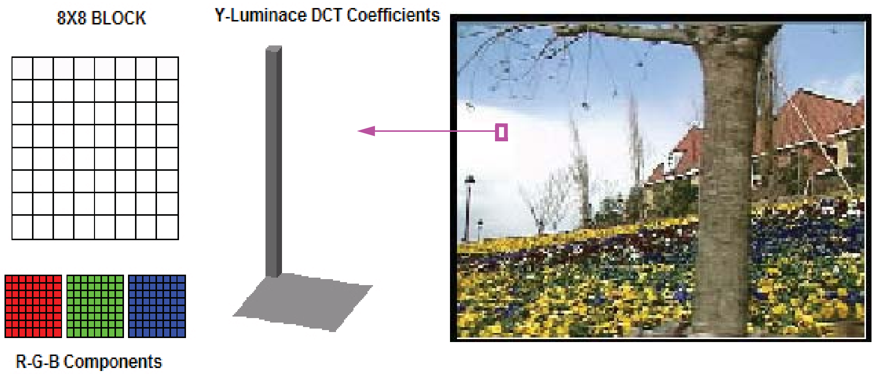
Discrete Cosine Transform

human beings have only a restriced capability to perceive high frequences. Images reduced from their high frequent elements may have a high degree of quality while being compressed quite considerably

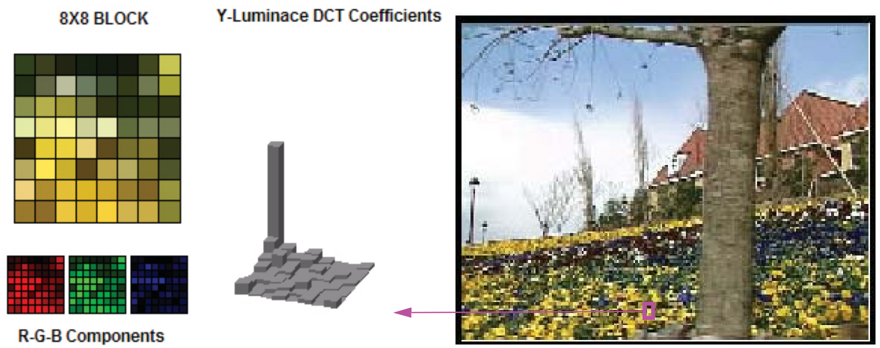
- ❑ Low frequency components of an image correspond to brief outlines of image objects
- ❑ High frequency components of an image correspond to fine structures

To decrease the high frequencies the data must be sorted by frequencies → DCT

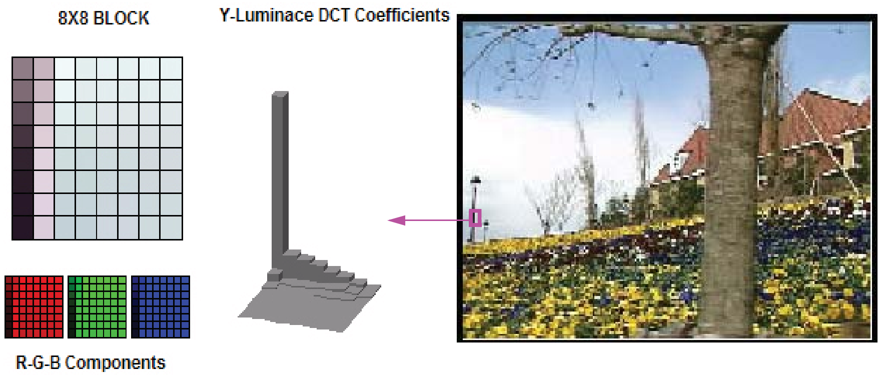
DCT-Coefficients



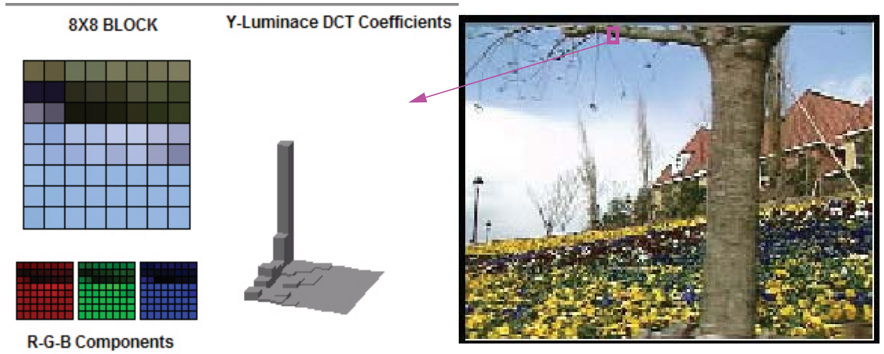
DCT-Coefficients



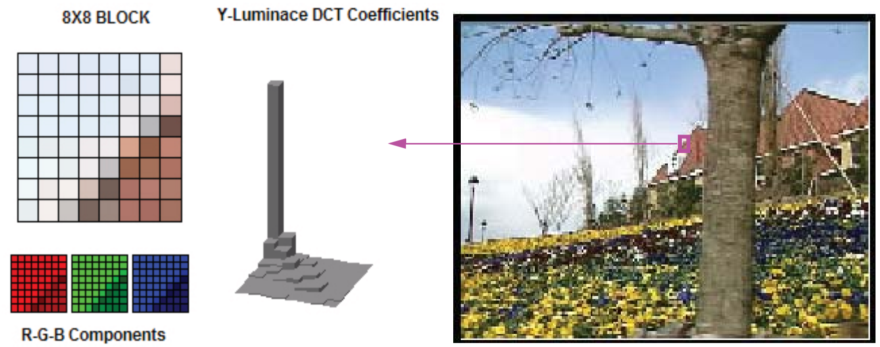
DCT-Coefficients



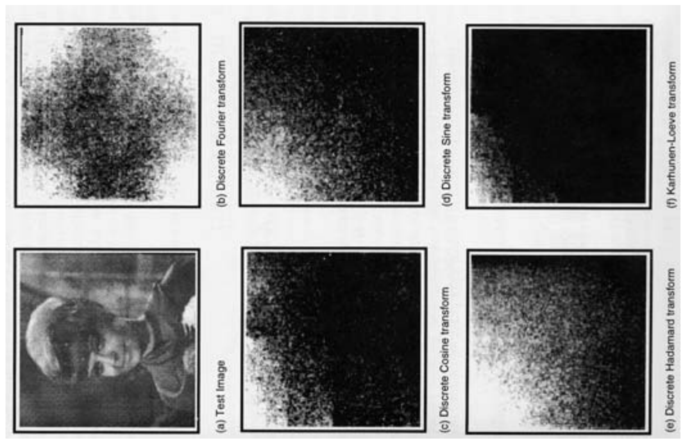
DCT-Coefficients



DCT-Coefficients



Why DCT?



JPEG Image Processing (DCT)

- Inverse Discrete Cosine Transformation (I-DCT) maps DCT coefficients to sampled values

$$s_{xy} = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} c_u c_v s_{vu} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}$$

with c_u, $c_v = 1/\sqrt{2}$ for u,v=0; otherwise c_u, $c_v = 1$

- DCT and I-DCT cannot be calculated in full precision ==> lossy compression
- JPEG does not define precision parameters, therefore various implementations exist
- many AC coefficients with small values (around zero)

JPEG Quantization

- goal: to “throw out” bits (truncation)
- uniform quantization: divide coefficient values S(u,v) by N and round result

Q: In S(u,v) how many bits should be truncated?

A: use quantization tables

  - quantization tables consist of 64 elements, each value uses 8-bit: Q_{uv}
  - new compressed values by using tables
    $Sq_{uv} = S_{uv} / Q_{uv}$
  - standard defines two default tables (luminance, chroma)

JPEG Quantization

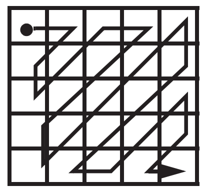
luminance quantization table

16	11	12	14	12	10	16	14
13	14	18	17	16	19	24	40
26	24	22	22	24	49	35	37
29	40	58	51	61	60	57	51
56	55	64	72	92	78	64	68
87	69	55	56	80	109	81	87
95	98	103	104	103	62	77	113
121	112	100	120	92	101	103	99

human eye most sensitive to low frequencies (upper left corner)

JPEG Entropy Encoding

- first step: map 8x8 pixel blocks into 64 element vector using zig-zag-scan



- DC coefficients processing:
  - DC coefficients determine basic color of data unit
  - DC coefficient is large, but often close to previous value ==> encode difference

JPEG Entropy Encoding

DC_0	DC_1	DC_2
DC_3	DC_4	DC_5
DC_6	DC_7	DC_8

Predictor

DC_0	$Diff_1$	$Diff_2$
$Diff_3$	$Diff_4$	$Diff_5$
$Diff_6$	$Diff_7$	$Diff_8$

$Diff_i = DC_i - DC_{i-1}$ for $i>0$

- ❑ AC coefficients processing:
  - ❑ processing order of the AC coefficients using zig-zag scan (coefficients with lower frequencies are encoded first) ==>
  - ❑ sequence of similar data bytes ==> efficient entropy coding
- ❑ JPEG standard specifies Huffman or Arithmetic coding, but sequential encoding mode uses only Huffman coding.

JPEG — Entropy Encoding

Algorithm:

1. apply run-length coding of AC coefficient of zero values
2. apply Huffman coding on DC and AC coefficients

(only DC coefficients explained in detail)

JPEG — Entropy Encoding

DC coefficients—Huffman coding

- ❑ categorize DC values into DC code tables
- ❑ difference magnitude categories for DC coefficients (12 categories)

Diff Values	**SSSS (number of bits needed to encode Diff**
0	0 bit
-1, 1	1 bit
-3, ..., 3	2 bits
-7, ..., 7	3 bits
.	
.	
.	
-2047, ..., 2047	11 bits

JPEG — Entropy Encoding

DC coefficients —Huffman encoding

- ❑ handle SSSS as Huffman symbol, get p(0), p(1), ... p(11) and create Huffman tree ==> Huffman code for SSSS
- ❑ for each category, an additional bits field is appended to the code word to uniquely identify which difference in that category actually occurred.
- ❑ send: (Huffman codeword, actual value)

EX: if SSSS=2 has the Huffman code 001 and Diff=-3 ==>

send $\underbrace{001}_{\text{Huffman code for 2}}\underbrace{00}_{\text{-3 in two's complement is 00 because 3=>11}}$

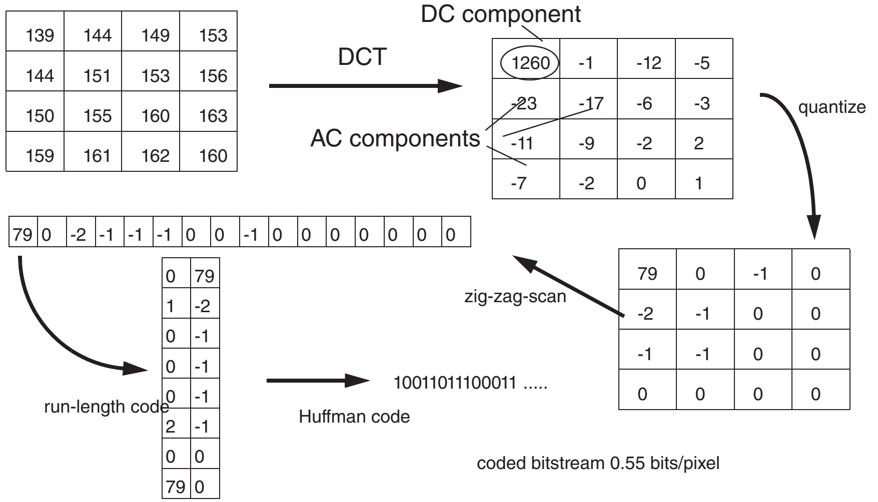
Huffman code for 2

-3 in two's complement is 00 because 3=>11

Comments

- applications do not have to include both an encoder and decoder if the compression process agrees on a common table
- the encoded data stream has a fixed *Interchange Format*
    - encoded image data
    - chosen parameters
    - tables of the coding process
- in regular mode, the interchange format includes all of the information necessary for decoding without any previous knowledge of the coding process

JPEG Block Encoding (Example)



JPEG Coding/Decoding (Example)

original block

139	144	149	153
144	151	153	156
150	155	160	163
159	161	162	160

reconstructed block

144	146	149	152
148	150	152	154
155	156	157	158
160	161	161	162

errors

-5	-2	0	1
-4	1	1	2
-5	-1	3	5
-1	0	1	-2

Motion Picture Experts Group (MPEG)

- MPEG Objectives and Standards
- General Information about MPEG
- MPEG-1
- MPEG-2
- MPEG-4
- (MPEG-7 und MPEG-21)

MPEG Objectives

- to deliver acceptable video quality at compressed data rates between 1.0 and 1.5 Mbps (MPEG-1)
- to support either symmetric or asymmetric compress/ decompress applications
- when compression takes it into account, random-access playback is possible
- when compression takes it into account, fast-forward, fast-reverse or normal reverse playback modes are available
- audio/video synchronization will be maintained

MPEG Objectives

- catastrophic behavior in the presence of data errors should be avoidable
- when required, compression/decompression delay can be controlled
- editability should be available when required by applications
- sufficient format flexibility to support playing of video in windows
- the processing requirements should not preclude the development of low-cost chipsets which are capable of encoding in real-time

MPEG Standards

Standard specifies audio, video and system layers

- several standards defined
  - MPEG-1 targeted at low data rates (VHS quality at 1.5 Mbits/ sec) [1992]
  - MPEG-2 targeted at high quality, hence high data rates (studio quality up to 15Mbits/sec) [1994]
  - MPEG-4 targeted at very low bit rates (<64 kbs) with small images [1998]

MPEG General Information

- MPEG standard defines audio, video coding and system data streams with synchronization
- MPEG considers explicitly functionalities of other standards, e.g. it uses JPEG
- MPEG stream provides information on
  - aspect ratio (e.g., 1:1, 4:3, 16:9)
  - refresh frequencies (8 frequencies encoded: 23.976 Hz. 24 Hz, 25 Hz, 29.97 Hz, 30 Hz, 50 Hz, 59.94 Hz and 60 Hz)

MPEG-1 General Information

- ❑ audio and video compression designed to operate at CD-ROM speeds (video—1.150 Mbits/sec, audio—0.256 Mbits/sec, system 0.094 Mbits/sec)
- ❑ audio compressor uses subband coder with psychoacoustic model to reduce bit rate
- ❑ video compressor uses block transform coder with motion-compensated inter-frame coding
- ❑ must distinguish syntax and semantics in the standard—syntax is much more flexible than current standard (wide variation in usage: e.g., variable/constant bit rates, higher bit rates, larger images)

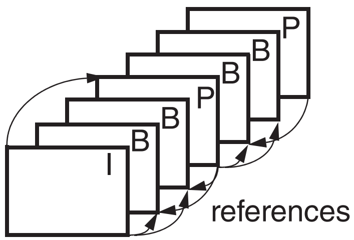
MPEG Video Standard

- ❑ each image consists of three components (1 luminance, 2 chrominance with half resolution)
- ❑ pixel precision—8 bits for each component
- ❑ example of a video format: 352x240 pixels, 30 frames/sec, chrominance components: 176x120 pixels
- ❑ each image is divided into areas called macroblocks (useful for compression based on motion estimation)
- ❑ each macroblock is partitioned into 16x16 pixels for luminance, 8x8 pixels for chrominance components

MPEG Video Standard

Video/Image Processing

- ❑ 4 types of image coding for video processing



why?

- ❑ demand for an efficient coding scheme and fast random access
- ❑ achieve high compression rates; exploit temporal redundancies of subsequent frames (interframe)

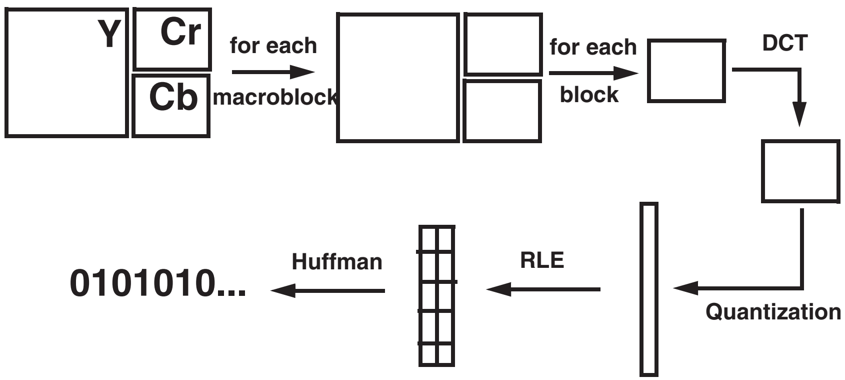
MPEG Video Standard

Intra-coded images (I-frames)

- ❑ self-contained without any references to other images, treated as still images; MPEG uses JPEG for I-frames
- ❑ The compression rate of I-frames is the lowest within MPEG.
- ❑ I-frames are points to random access in MPEG stream
- ❑ i-frames use 8x8 blocks defined within a macroblock, on these blocks DCT is performed. Quantization is by constant value for all DCT coefficients.

MPEG Video Standard

Intra-coded images (I-frames)



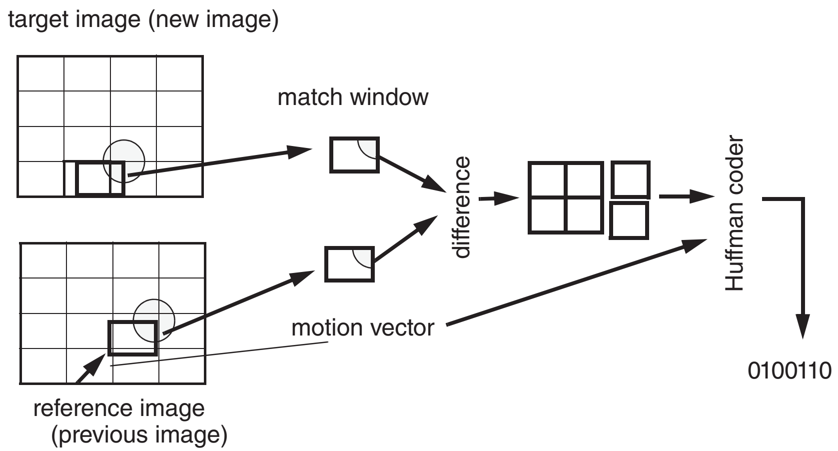
MPEG Video Standard

Predictive-coded Frame (P-frame)

- ❑ require information of the previous I-frame and/or previous P-frames for encoding and decoding
- ❑ coding of P-frames: utilize successive images in which areas do not change or are shifted
- ❑ temporal redundancy: determine the last P-frame or I-frame that is most similar to the block under consideration
- ❑ use motion estimation method at the encoder

MPEG Video Standard

motion estimation



MPEG Video Standard (P-Frame)

- ❑ motion vector—a pair (x-offset, y-offset) that specifies a block elsewhere in the frame
- ❑ block coded by specifying motion vector and error term between source block and motion vector block
- ❑ block can also be skipped - just use previous block at that location
- ❑ the coder must determine if a macroblock should be coded predictively or as a macroblock of an I-frame
- ❑ quantization value per macroblock, we want to vary quantization to fine tune compression

MPEG Video Standard

Motion Computation (P-Frames)

- look for match window within a given search window
  - match window; macroblock
  - search window: how far away are we willing to look
- methods:
  - 1. SSD correlation $\Sigma (x_i - y_i)^2$
  - 2. SAD correlation $\Sigma |x_i - y_i|$

MPEG Video Standard

P-Frame Processing:

- apply 2D DCT to macroblocks not reduced
- motion vector of adjacent macros often differs slightly
- the maximum size of the motion vector is not defined in the standard
- P-frames consist of I-frame macroblocks and predictive macroblocks
- P-frames are quantized and entropy encoded: RLE

MPEG Video Standard

B-Frames (Bi-directionally predictive-coded)

- require information of the previous and following I and/or P-frame
- motion vector from previous and future reference frame
- average of blocks from previous and future frame

MPEG Video Standard

D-Frames (DC-coded Frames)

- D-frames can be used for fast forward, fast rewind mode
- DC-parameters are DCT-coded, AC coefficients are neglected
- D-frames consist of the lowest frequencies of an image
- used only in MPEG-1

MPEG Video Standard

Decoding

- using B-frames, the order of images in an MPEG-encoded data stream differs from the actual decoding order:
- display order
  - type of frame: I B B B P B B B I B B B P
  - frame number: 1 2 3 4 5 6 7 8 9 10 11 12 13
- decoding order
  - type of frame: I P B B B I B B B P B B B
  - frame number: 1 5 2 3 4 9 6 7 8 13 10 11 12

MPEG Video Standard

Quantization

- AC coefficients of B- and P-frames are usually large values, I-frames have smaller values
- ==> MPEG quantization is adjusted
  if data rate increases over threshold => quantization enlarges the step size
  if data rate decreases => quantization is performed with finer granularity