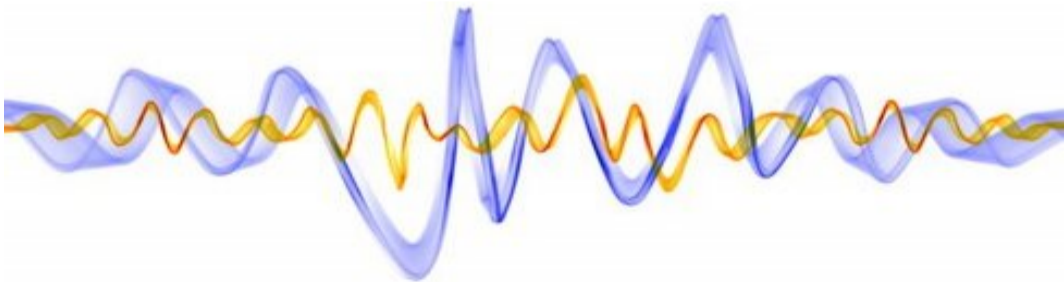


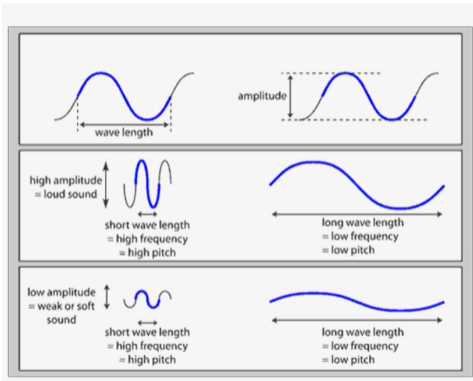
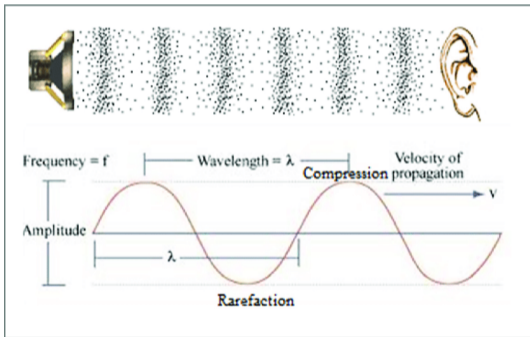
# PV188 Principy zpracování a přenosu multimédií

Miloš Liška  
xliska@fi.muni.cz

podzim 2024

- ▶ Podélné mechanické vlnění v látkovém prostředí (vzduch), které je schopno vyvolat v lidském uchu sluchový vjem
- ▶ Aplikují se běžné fyzikální jevy jako pro jiná vlnění:
  - ▶ odraz
  - ▶ ohyb
  - ▶ difrakce





- ▶ Frekvence
  - ▶ Lidé slyší zvuky ve frekvenčním pásmu přibližně 16 Hz až 20 kHz
- ▶ Intenzita vlnění
  - ▶ Intenzita vlnění (zvuku) je definována jako množství energie, které projde jednotkovou plochou kolmou na směr šíření na jednotku času.
  - ▶ Zavádíme prahová intenzitu jako nejnižší ještě slyšitelnou intenzitu čistého tónu o frekvenci 1 kHz
  - ▶ Hladina intenzity zvuku (dB) se zavádí jako logaritmická škála, která prahové intenzitě přiřadí nulovou hodnotu. Logaritmickou škálu používáme protože intenzita běžných zvuků kolísá v rozsahu několika dekád.
- ▶ Hlasitost zvuku
  - ▶ Vztah mezi intenzitou vlnění a intenzitou vjemu závisí mimo jiné i na frekvenci
  - ▶ Proto definujeme ještě hladinu hlasitosti a hlasitost, které berou do úvahy fyziologické vnímání intenzity zvuku v závislosti na jeho frekvenci

- ▶ Složitý proces, závislý na mnoha faktorech, pro který zatím nebyla vytvořena uspokojivá teorie
- ▶ Frekvenční rozsah
  - ▶ Frekvenční rozsah zvuku, který většina lidí vnímá, začíná kolem 20 Hz a dosahuje ke 20 kHz. S rostoucím věkem horní hranice výrazně klesá.
- ▶ Dynamický rozsah
  - ▶ Dynamický rozsah lidského ucha (rozdíl mezi nejhlasiťejším a nejtíšíším vnímatelným zvukem) je uprostřed slyšitelného frekvenčního pásma asi 120 dB. Na okrajích pásma je mnohem menší.
- ▶ Rozlišování frekvence
  - ▶ Schopnost rozlišit frekvence tónů se u každého člověka liší a je frekvenčně závislá. Uprostřed slyšitelného frekvenčního pásma je rozlišovací schopnost vyšší než na okrajích pásma.

- ▶ Frekvenční maskování
  - ▶ Schopnost odlišit dva frekvenčně blízké tóny. Pokud znějí dva tóny současně, může jeden z nich potlačit slyšitelnost toho druhého. Maximální úroveň maskovaného signálu je závislá na frekvenční vzdálenosti a úrovni maskujícího signálu. Maskovací schopnost je též závislá na frekvenci maskujícího tónu. Vnímání tónů s blízkými frekvencemi je ovlivněno šířkou kritického pásma. To má na nejnižších kmitočtech velikost kolem 100 Hz, zatímco na nejvyšších kmitočtech dosahuje až 4 kHz. Využití u některých algoritmů pro kompresi zvukových dat, např. MP3, Vorbis nebo ATRAC.
- ▶ Časové maskování
  - ▶ Pokud po hlasitém tónu následuje stejný tón s menší hlasitostí, je jeho vnímání potlačeno. Potlačen může být i tichý tón předcházející maskovacímu tónu.

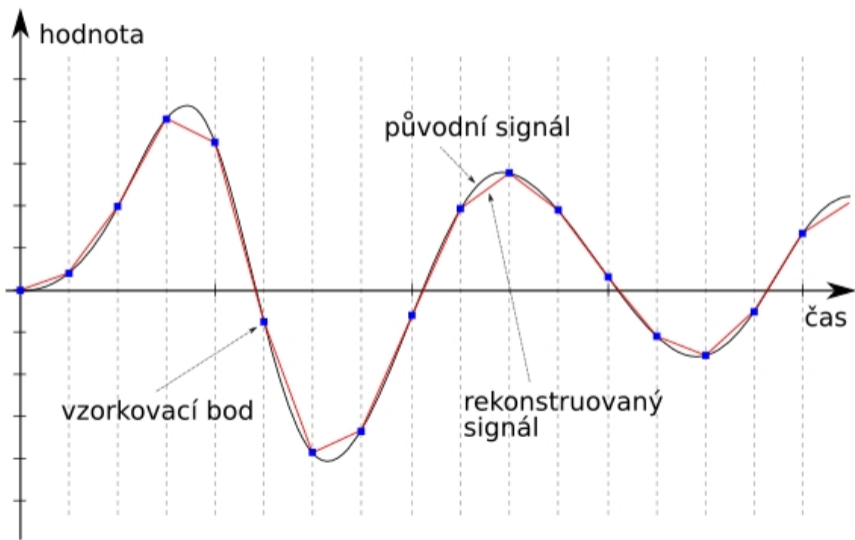
- ▶ Akvizice zvuku
- ▶ Diskrétní reprezentace spojitého zvukového vlnění
- ▶ Zpracování diskrétních zvukových dat
  - ▶ Analýza zvuku za použití psychoakustických modelů
  - ▶ Použití zvukových filtrů: ekvalizace, odstranění šumu a echa apod.
  - ▶ Datová komprese
- ▶ Uložení na datový nosič, pevný disk apod.
- ▶ Přenos po síti

- ▶ A/D převodník, typicky zvuková karta v PC ale i jiná zařízení (mobilní telefon, PDA, MP3 přehrávač)
- ▶ Obvykle schopnost zpracování 2 (ale i více) audio kanálů
- ▶ Mikrofonní a linkový vstup (viz přednášky Ing. Šilera)
- ▶ Kvalita akvizice zvuku závisí nejvíce na parametrech použitého zařízení (vzorkovací frekvence, přesnost vnitřních hodin, odstup signálu od šumu apod.)



- ▶ V pravidelných intervalech změříme intenzitu vstupního analogového signálu
- ▶ Interval mezi jednotlivými vzorky nazýváme vzorkovací interval
- ▶ Informaci o průběhu signálu mezi jednotlivými diskrétními měřeními ztrácíme
- ▶ Abychom byli schopni zaznamenat a rekonstruovat vstupní signál s nejvyšší frekvencí  $f$ , je třeba vzorkovací frekvence alespoň  $2f$  (Nyquistův teorém)
  - ▶ Tj. pro nejvyšší slyšitelnou frekvenci 20 kHz potřebujeme vzorkovací frekvenci větší než 40 kHz
- ▶ Definovaný interval = *vzorkovací frekvence*
- ▶ Typické vzorkovací frekvence:
  - ▶ 8 kHz - telefony
  - ▶ 11 kHz - lidská řeč
  - ▶ 22 kHz - kvalita odpovídající audiokazetě
  - ▶ 44.1 kHz - CD
  - ▶ 48 kHz, 96 kHz, 192 kHz - DVD

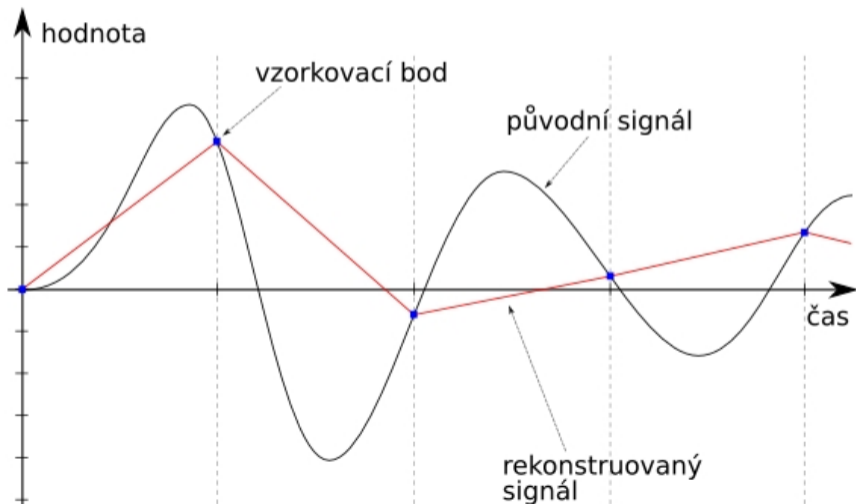
# Vzorkování



- ▶ Nutná (ale ne postačující) podmínka pro to, abychom dokázali dobře rekonstruovat původní zvuk
- ▶ Vzorkovací frekvence musí být více než dvakrát větší, než největší frekvence vzorkovaného zvuku.
- ▶ Lidské ucho slyší frekvence maximálně 20 kHz z toho plyne dohodnutá vzorkovací frekvence 44.1 kHz

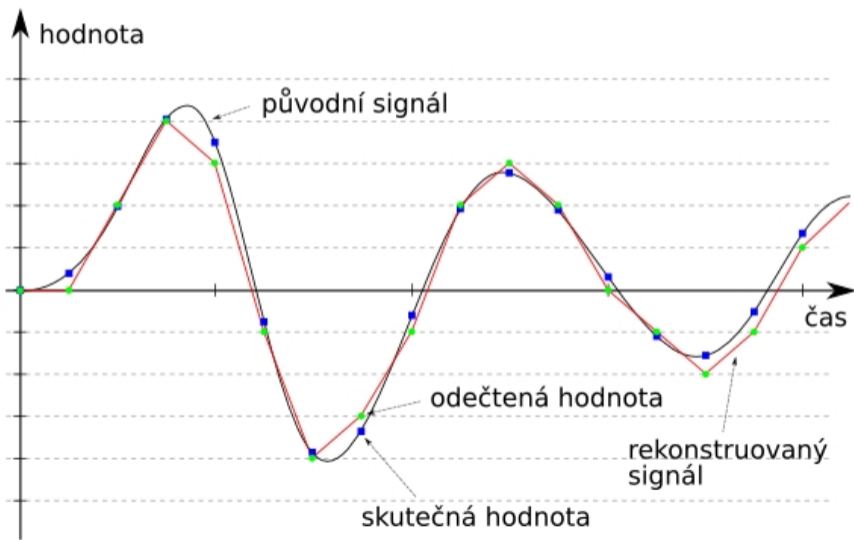
# Podvzorkování

- ▶ Podvzorkované frekvence se „schovají“ za jiné frekvence

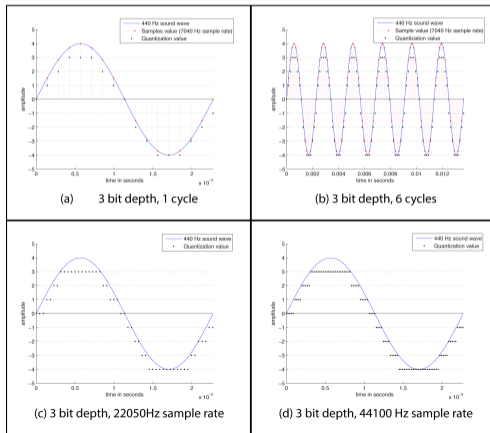


- ▶ Hodnoty, naměřené při vzorkování, reprezentujeme jako čísla omezené délky (reprezentované konečným počtem bitů)
- ▶ Pro lepší představu rozdělení svislé osy zvukové křivky na diskrétní hodnoty
- ▶ Naměřené hodnoty zaokrouhlujeme na nejbližší diskrétní hodnotu
  - ▶ Vzniká kvantizační šum
- ▶ Počet bitů vyhrazených na reprezentaci intenzity vstupního signálu určuje kolik různých intenzit můžeme reprezentovat
- ▶ 1 bit odpovídá přibližně 6 dB
- ▶ Typické reprezentace:
  - ▶ 8 bitů na vzorek (256 úrovní) - telefon
  - ▶ 16 bitů na vzorek (65 536 úrovní) - CD
  - ▶ 24 bitů na vzorek (16 777 216 úrovní) - DVD
  - ▶ 64 bitů na vzorek - použití v sonarech

- ▶ Několik úrovní intenzity zvuku se reprezentuje jednou hodnotou = *kvantovací šum*
- ▶ Lze vylepšovat:
  - ▶ Ditheringem
  - ▶ Nelineárním kvantováním
    - ▶ Ucho vnímá zvuk zhruba logaritmicky
    - ▶ Zhuštění úrovní v oblasti, kde je to třeba. U zvuku je vhodné jemněji rozlišovat v oblasti nízkých kmitočtů, protože ucho je v této oblasti citlivější na šum.



- ▶ Vzorkujeme a kvantujeme analogový signál, jehož časový průběh je na diagramu
- ▶ Vzorkování je v diagramu naznačeno čárkovanými svislými čarami (okamžiky měření na časové ose)
- ▶ Pro reprezentaci intenzity signálu v každém vzorku používáme 3 bity

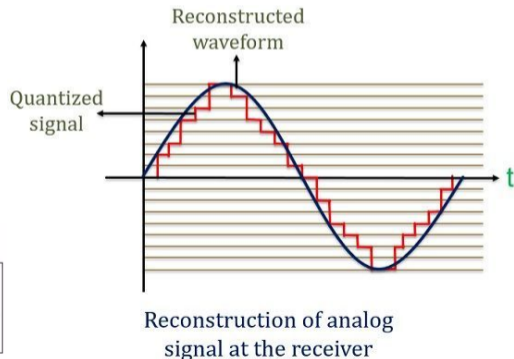
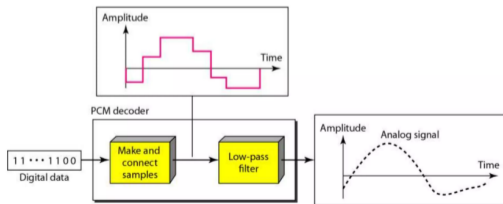




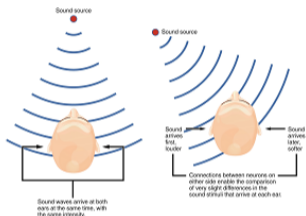
# Rekonstrukce zvukového signálu

## ► Jednoduchý proces

- Kvantované vzorky zpracováváme postupně
- Držíme amplitudu právě zpracovaného vzorku (napětově nebo proudově) dokud nepřijde následující vzorek
- Výstupní signál projde low-pass filtrem, který ořízne všechny vyšší frekvence než je nejvyšší frekvence vzorkovaného signálu



- ▶ Člověk má dvě uši, mozek dostává dvě nezávislé informace o zvuku ze kterých na základě intenzity vnímaného zvuku vytváří zvukový prostorový vjem.



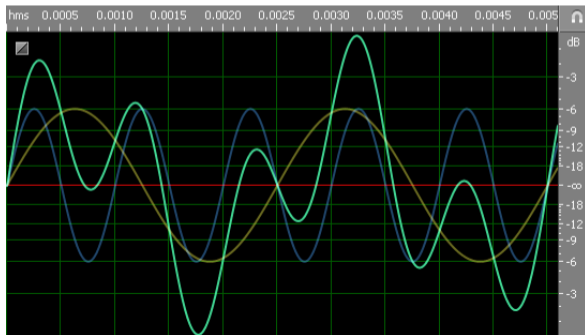
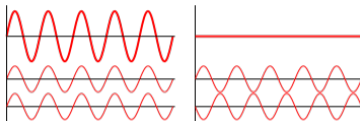
- ▶ Monofonní zvuk = 1 kanál
- ▶ Prostorový zvuk
  - ▶ Pro každý směrový zdroj zvuku nezávislý kanál
  - ▶ Klasický stereofonní zvuk přicházející „zleva a zprava“ jsou dva nezávislé audio kanály
  - ▶ 5.1 prostorový zvuk je 6 nezávislých audio kanálů
- ▶ Každý nezávislý zvukový kanál je vzorkovaný a kvantovaný tak, jak jsme si ukazovali na předchozích slidech

- ▶ Pulse Code Modulation
- ▶ Analogový signál → vzorkování → kvantování → PCM
- ▶ Datové toky:
  - ▶  $8 \text{ kHz} \times 8 \text{ b} \times 1 \text{ kanál} = 64 \text{ kbps}$
  - ▶  $44,1 \text{ kHz} \times 16 \text{ b} \times 2 \text{ kanály} = 1411 \text{ kbps}$  (cca. 1,4 Mbps)
  - ▶  $44,1 \text{ kHz} \times 16 \text{ b} \times 6 \text{ kanálů} = 4,233 \text{ kbps}$  (cca. 4,2 Mbps)
  - ▶  $192 \text{ kHz} \times 24 \text{ b} \times 2 \text{ kanály} = 9000 \text{ kbps}$  (cca 9 Mbps)
- ▶ Použití:
  - ▶ digitální telefonní linky, CD, DVD audio, DVD
- ▶ Bežně implementované v HW
  - ▶ Zvukové karty, audio přehrávače, digitální záznamníky
  - ▶ AD / DA převodník
  - ▶ Závislost na externím hodinovém signálu

- ▶ DPCM (Differential Pulse Code Modulation)
  - ▶ Ztrátová komprese, používá 4 bity na záznam rozdílů mezi dvěma vzorky nezávisle na původním kvantování
- ▶ ADPCM (Adaptive Pulse Code Modulation)
  - ▶ snaha o prediktivní analýzu potřebného počtu bitů pro záznam rozdílu mezi dvěma vzorky v závislosti na složitosti signálu
- ▶  $\mu$ -law (A-law v Evropě)
  - ▶ podobná komprese jako ADPCM, použití v digitální telekomunikaci (zejména Japonsko a USA), kvantování jednotlivých rozdílů odpovídá 13 bitům
- ▶ LPCM (Linear Predictive Coding)
  - ▶ Až 8 kanálů pro audio; vzorkovací frekvence 48 kHz nebo 96 kHz; 16, 20 nebo 24 bitů na vzorek
  - ▶ Maximální bitrate (datový tok) 6.144 MBps

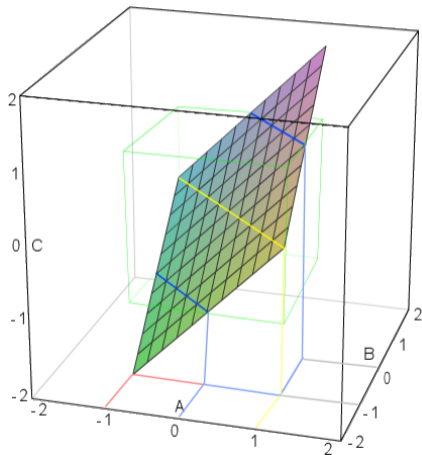
# Mixování audia

- ▶ Směšování dvou zvukových vln



# Mixování audia

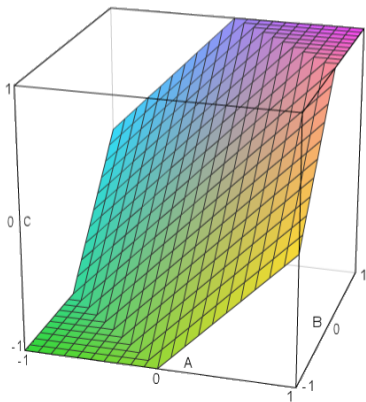
- ▶ Směšování dvou zvukových vstupů
- ▶  $C = A + B$



- ▶ Předpokládáme floating point PCM s hodnotami  $\langle -1, 1 \rangle$

# Mixování audia – normalizace

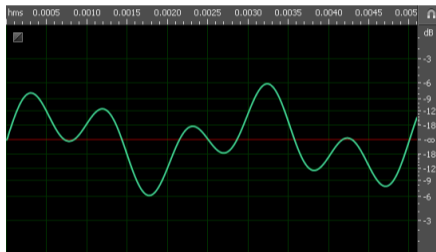
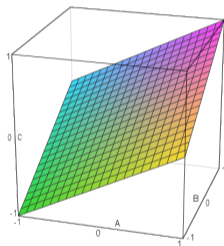
- ▶ Smixované výstupní audio stále reprezentujeme jako floating point PCM s hodnotami  $\langle -1, 1 \rangle$
- ▶ Výstup je třeba normalizovat tak, aby výsledky byly vždy v tomto intervalu
- ▶ Clipping
  - ▶  $C = \min(C_{max}, \max(C_{min}, A + B))$



# Mixování audia – normalizace

▶ Lineární attenuace

▶  $C = \frac{A+B}{2}$



▶ Co když je jeden ze zdrojů potichu?

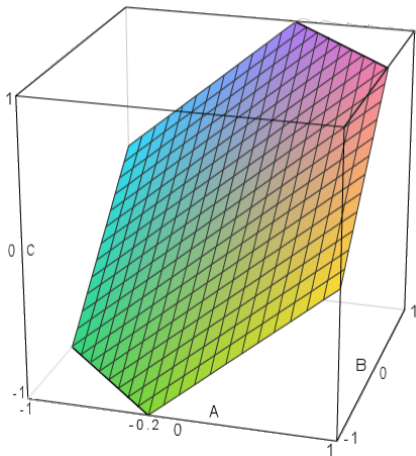
▶ Výstup je celkově utlumený.



# Mixování audia – normalizace

► Pre nebo post-mix normalizace

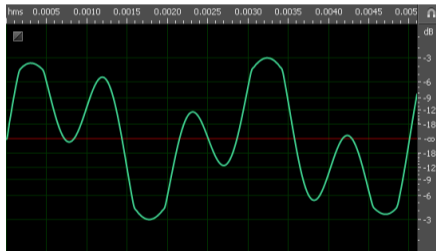
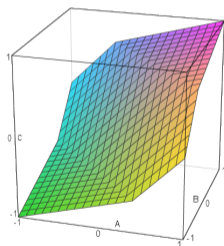
$$\text{► } C = \frac{A+B}{\frac{\max(|A_i+B_i|)}{\frac{\text{Crange}}{2}}} \quad \forall i \in \mathbb{N}, i \leq T$$



# Mixování audia – normalizace

- ▶ Dynamická komprese rozsahu - lineární
- ▶  $x \in \mathbb{R} \wedge -2 \leq x \leq 2; t \in \mathbb{R} \wedge 0 \leq t < 1$

$$f_i(x; t) = \begin{cases} x & -t \leq x \leq t; \\ \frac{x}{|x|} \cdot \left( t + \frac{1-t}{2-t} \cdot (|x| - t) \right) & |x| > t \end{cases}$$



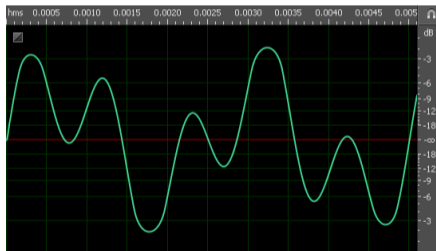
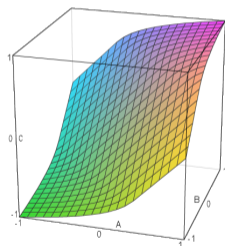
- ▶  $x = A + B, t = 0.6$
- ▶ Všechny amplitudy jsou v rozsahu, nízké a střední amplitudy jsou OK, vysoké amplitudy jsou ztlumené

# Mixování audia – normalizace

- ▶ Dynamická komprese rozsahu - logaritmická

- ▶  $x \in \mathbb{R} \wedge -2 \leq x \leq 2; t \in \mathbb{R} \wedge 0 \leq t < 1$

$$f_l(x; t) = \begin{cases} x & -t \leq x \leq t; \\ \frac{x}{|x|} \cdot \left( t + (1-t) \cdot \frac{\ln(1+f_\alpha(t) \cdot \frac{|x|-t}{2-t})}{\ln(1+f_\alpha(t))} \right) & |x| > t \end{cases}$$



- ▶  $x = A + B, t = 0.6$

- ▶ Všechny amplitudy jsou v rozsahu, nízké a střední amplitudy jsou OK, vysoké amplitudy nejsou tolik utlumené jako v případě lineární komprese

- ▶ Red book audio standard (Philips a Sony v roce 1980)
- ▶ 2 audio kanály
- ▶ Vzorkovací frekvence 44.1 kHz, 16 bitů na vzorek, PCM
- ▶ Bit rate =  $44100 \text{ vzorků/s} \times 16 \text{ bitů/vzorek} \times 2 \text{ kanály} = 172.2 \text{ kBps}$  (více než 10 MB za minutu)
- ▶ Hodnoty jednotlivých vzorků jsou v rozmezí -32768 do +32767.
- ▶ Data nejsou ukládána sekvenčně, kombinovaná detekce a oprava chyb
- ▶ Pozn. pod čarou:
  - ▶ Jakákoliv ochrana proti kopírování porušuje red book standard
  - ▶ S nástupem MP3 se z red book CD stává poněkud obsoletní technologie

- ▶ Ukládání vysoce věrného (hi-fi) zvukového obsahu na DVD média
- ▶ Srovnání s CD
  - ▶ Větší kapacita nosiče, větší stopáž
  - ▶ Mnohem větší kvalita záznamu, větší vzorkovací frekvence, více bitů na vzorek
  - ▶ Mnoho možných konfigurací počtu audio kanálů (1.0 až 7.1)

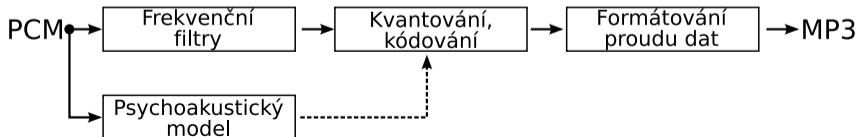
	16-, 20- or 24-bit					
	44.1 kHz	48 kHz	88.2 kHz	96 kHz	176.4 kHz	192 kHz
<b>Mono (1.0)</b>	Yes	Yes	Yes	Yes	Yes	Yes
<b>Stereo (2.0)</b>	Yes	Yes	Yes	Yes	Yes	Yes
<b>Quad (4.0)</b>	Yes	Yes	Yes	Yes	No	No
<b>Surround (5.1)</b>	Yes	Yes	Yes	Yes	No	No

- ▶ Záznam audia ve formátu LPCM
  - ▶ Nekomprimované audio
  - ▶ Většina DVD přehrávačů ale podporuje maximálně 16 bitů na vzorek při 48 kHz
- ▶ Maximální možný bitrate pro všechny kanály je 9.6 Mbps

- ▶ Obecný přístup ztrátové komprese je jednoduchý. Po úvodním předzpracování se přeskupí nebo transformují data tak, aby bylo možno lehce oddělit důležité informace od nedůležitých. Nedůležité informace se pak potlačí mnohem více (zahodí) než důležité a nakonec se výsledek zkomprimuje některým z bezztrátových kompresních algoritmů.
- ▶ Transformace dat
  - ▶ MDCT
  - ▶ převede původní data do jiné domén, typicky z časové do frekvenční
- ▶ Potlačení části dat
  - ▶ psychoakustický model, který určuje, jaká data mohou být potlačena nebo dokonce úplně odstraněna (nahrazena nulami)
  - ▶ při kompresi zvuku se hledají frekvence, které člověk stejně nemůže vnímat a ty se zahodí

- ▶ Modifikovaná diskrétní kosínová transformace
- ▶ MDCT v každém z kanálů transformuje analogový signál na frekvenční koeficienty
- ▶ Výstupem MDCT je řada frekvenčních koeficientů → *bezztrátová transformace*
- ▶ Koeficienty pro méně důležitá frekvenční pásma (na základě poznatků o vnímání zvuku) zahazujeme → princip ztrátového kódování zvuku

- ▶ Ztrátový kompresní mechanismus, ISO/IEC standard v roce 1991, první rozšířená implementace Fraunhofer society v roce 1994.
- ▶ Komprimuje PCM audio v poměru přibližně 1:11
- ▶ Použití psychoakustického modelu
  - ▶ na základě znalostí o lidském sluchu zahazujeme ty části signálu, které lidské ucho hůře slyší, nebo neslyší vůbec
  - ▶ na základě znalostí o zpracovávaném signálu (vážná hudba, metalový koncert apod.) vybíráme ta frekvenční pásma, která jsou nejpodstatnější
- ▶ Komprese signálu:
  - ▶ Komprese probíhá ve frekvenční doméně
  - ▶ MDCT → nelineární kvantování → Huffmanovo kódování





- ▶ Dostupné bitraty:
  - ▶ 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 a 320 kbps
  - ▶ VBR - audio v MP3 je dělené do rámců, každý rámeček může mít jiný bitrate
- ▶ Dostupné vzorkovací frekvence:
  - ▶ 32, 44.1 a 48 kHz
- ▶ Mono, Stereo, Joint-stereo
  - ▶ Člověk lépe pozičně lokalizuje zvuky s vyššími frekvencemi. Celá spodní část spektra se sloučí do jediného kanálu a navíc se uloží informace, jak prostorově posouvat zvuky o určitých frekvencích.
- ▶ Typické parametry kódování:
  - ▶ 44.1 kHz při 128 kbps se považuje za dobrý standard
  - ▶ Slyšitelně lepších výsledků lze dosáhnout při 192 kbps (zhruba ideální bitrate pro MP3 kompresi)

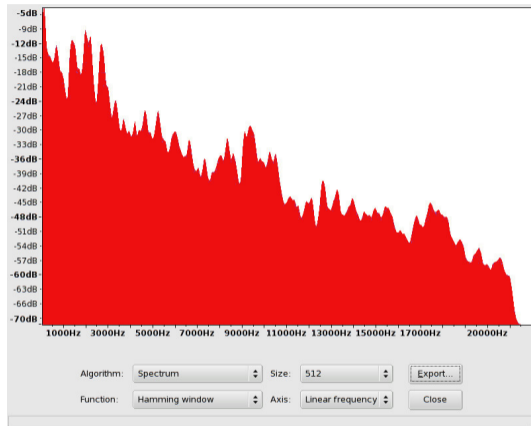
- ▶ CBR
  - ▶ Pro každý komprimovaný časový segment původního signálu máme k dispozici stejný objem dat
  - ▶ Objem dat dostupný pro kompresi signálu nemusí být efektivně využitý
- ▶ VBR
  - ▶ Komprimovaný signál dělíme na časové segmenty a těm variabilně přiřazujeme bitrate na základě složitosti signálu
  - ▶ Dva průchody
    - ▶ Analýza složitosti signálu v jednotlivých časových segmentech
    - ▶ Přiřazení bitrate každému segmentu na základě složitosti signálu a např. průměrného bitrate nebo cílového objemu dat
  - ▶ Vyšší kvalita komprimovaného signálu při zachování stejného objemu dat
  - ▶ VBR lze teoreticky provést i jednorůchodově na základě koeficientu kvality  $q$ . Pak ale nelze kontrolovat objem dat nutný pro kompresi signálu.

- ▶ Podpora metadat v podobě ID3 tagu
- ▶ Omezení a problémy formátu MP3
  - ▶ Slyšitelné artefakty při nižších bitratech (ozvěny apod.)
  - ▶ Maximální bitrate 320 kbps
  - ▶ Licenční poplatky

- ▶ Ztrátový kompresní mechanismus navržený v roce 1991
- ▶ DVD, DVB-T, interně v rádiích
- ▶ Sub-band kódování audia:
  - ▶ Komprese probíhá v časové doméně
  - ▶ Signál se rozdělí na 32 frekvenčních pásem, na základě psychoakustického modelu se některá zahodí a ostatní se komprimují samostatně.
- ▶ Dostupné bitraty:
  - ▶ 32, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320 a 384 kbps
  - ▶ 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 144 a 160 kbps (navíc v MPEG-2 layer II)
- ▶ Dostupné vzorkovací frekvence:
  - ▶ 32, 44.1 a 48 kHz
  - ▶ 16, 22.05 a 24 kHz (navíc v MPEG-2 layer II)
- ▶ MP2 nedokáže využít podobnost signálu ne jednotlivých kanálech. Při méně než 256 kbps je méně účinná než MP3.

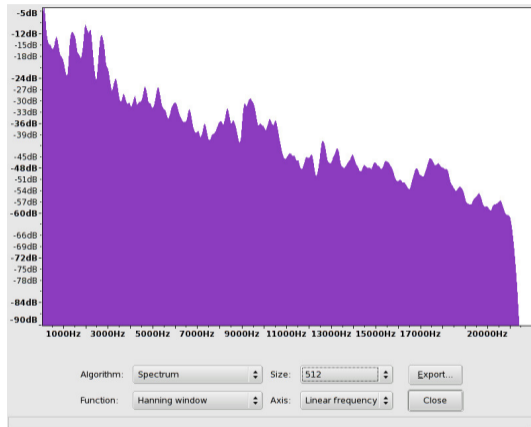
- ▶ Opensource ztrátová komprese obvykle spojovaná s obálkovým formátem OGG.
- ▶ Kódování audia:
  - ▶ Obdobné jako u MP3
  - ▶ Pokročilejší kvantování
  - ▶ Redukce šumu na základě statistických měření
- ▶ Podle poslechových testů dosahuje lepších výsledků než MP3 při kódování na 128 kbps
- ▶ Typické parametry kódování:
  - ▶ 44.1 kHz při 160 kbps se považuje za ideální nastavení
- ▶ Metadata „comments“ obdobně jako u MP3

# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



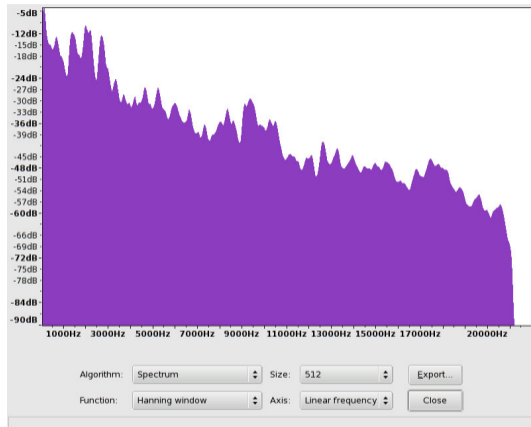
Referenční wav – 44 kHz, 16 b, stereo

# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



Ogg Vorbis – q10 cca. 427 kbps (Oggenc 1.0.2)

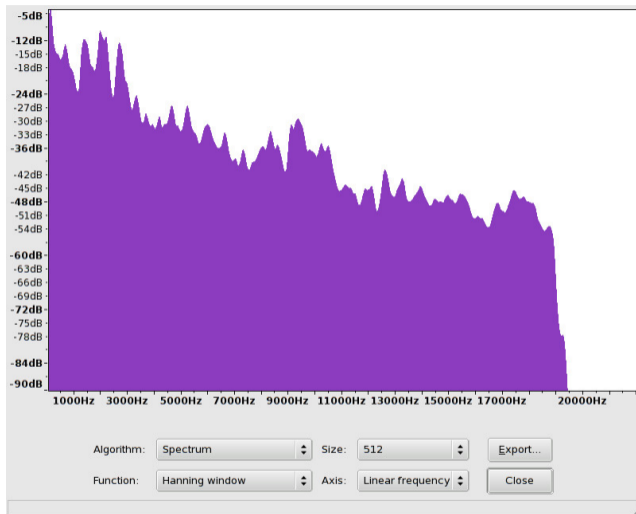
# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



Ogg Vorbis – q8 cca. 236 kbps (Oggenc 1.0.2)

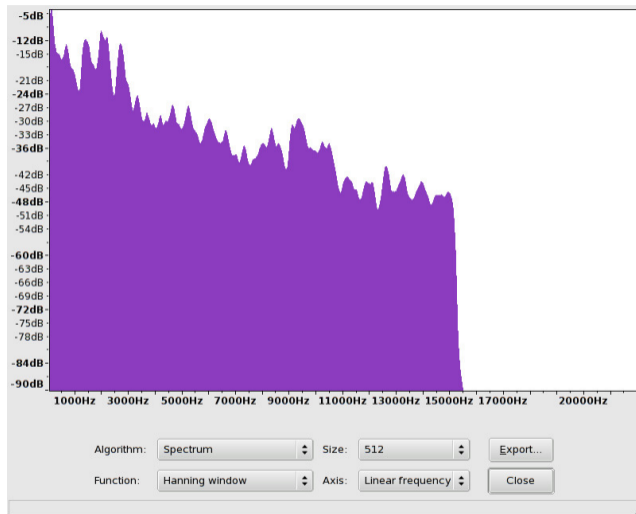


# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



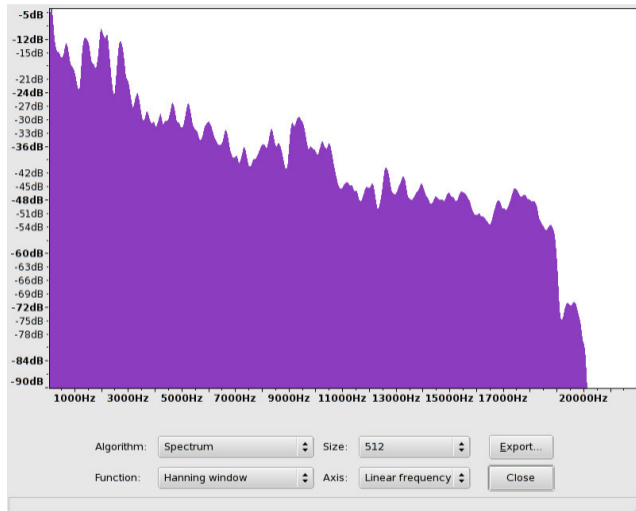
Ogg Vorbis – q4 cca. 122 kbps (Oggenc 1.0.2)

# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



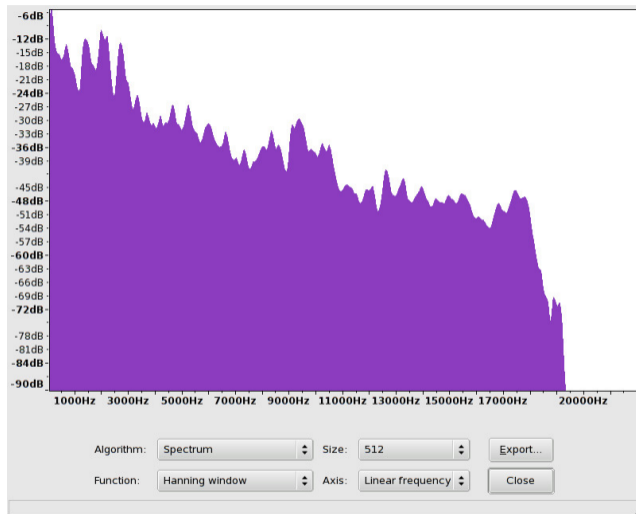
Ogg Vorbis – q0 cca. 56 kbps (Oggenc 1.0.2)

# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



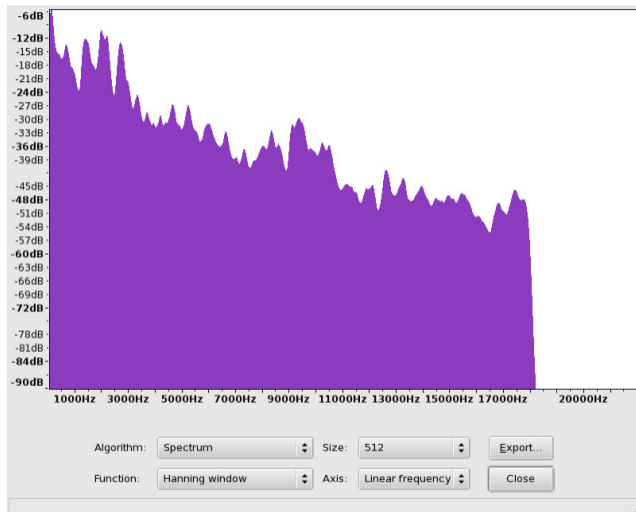
MPEG-I Layer 3 – studio quality = 256 kbps (Lame 3.95)

# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



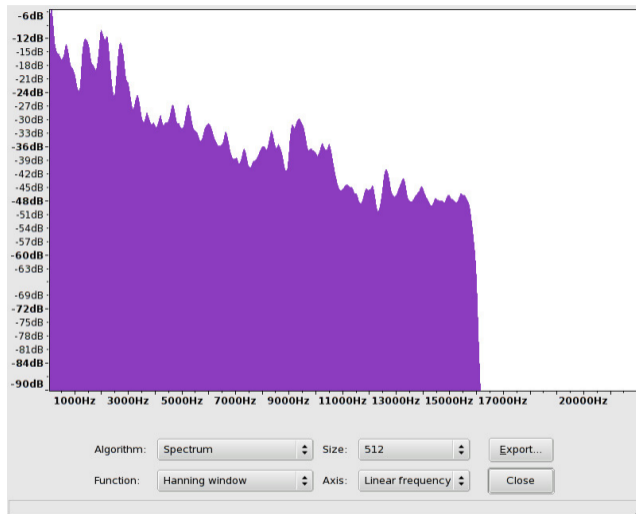
MPEG-I Layer 3 – CD quality = 192 kbps (Lame 3.95)

# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



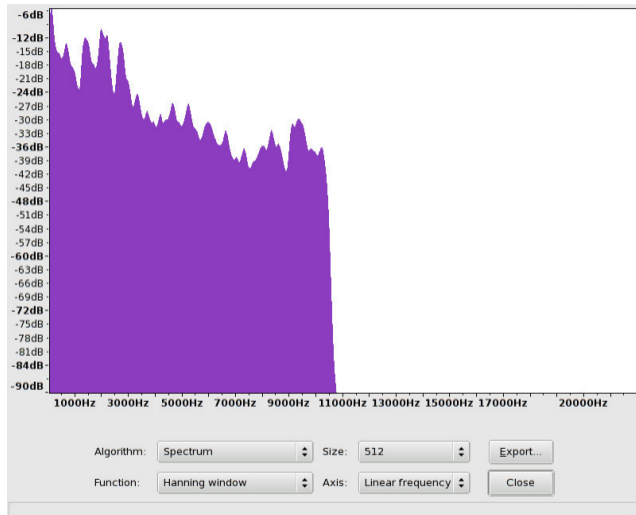
MPEG-I Layer 3 – hifi quality = 160 kbps (Lame 3.95)

# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



MPEG-I Layer 3 – tape quality = 112 kbps (Lame 3.95)

# Srovnání kvality komprese MPEG-I Layer 3 vs. Ogg Vorbis



MPEG-I Layer 3 – voice quality = 56 kbps (Lame 3.95)

- ▶ Standard pro ztrátovou kompresi skupiny MPEG (MPEG-4 part 3, MP4)
- ▶ Technologický nástupce formátu MP3
- ▶ Spotify apod.
- ▶ Kódování audia:
  - ▶ Silně využívá psychoakustické modely
  - ▶ Eliminuje redundance v již zakódovaném signálu
  - ▶ Implementuje samoopravné kódy
  - ▶ Jinak funguje obdobně jako MPEG-1 Layer III
- ▶ Modulární přístup ke kódování audia"
  - ▶ Existuje několik kódovacích profilů včetně Low latency
  - ▶ V závislosti na volbě profilu lze dosáhnout při 96 kbps stejných výsledků jako u MP3 při 128 kbps
- ▶ Téměř libovolné vzorkovací frekvence 8 kHz až 96 kHz
- ▶ Podpora pro až 48 audio kanálů (MP3 podporuje 2)



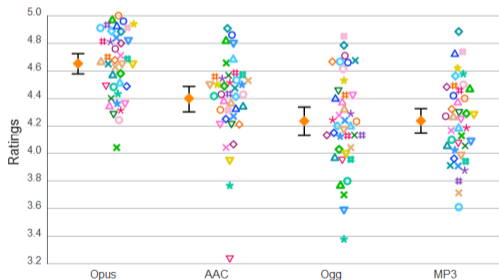
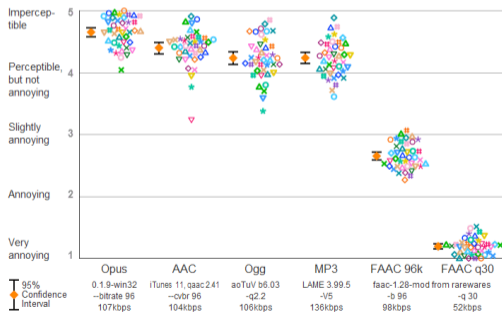
- ▶ Opensource bezztrátový kompresní formát
- ▶ Vzorky jsou kódovány pouze s celočíselnou přesností. To ale na druhou stranu eliminuje chyby vzniklé zaokrouhlováním a umožňuje přesnou reprodukci.
- ▶ Libovolná vzorkovací frekvence
- ▶ Každý vzorek může být kvantován 4 až 32 bity
- ▶ Dosahuje kompresního poměru přibližně 20 až 30% ve srovnání s PCM

- ▶ Opensource ztrátový formát určený pro kompresi řeči
  - ▶ Samohlásky vyžadují vyšší bitrate
  - ▶ Naopak sykavky, f apod. lze kódovat s nízkým bitratem
- ▶ Vzorkovací frekvence: 8 kHz (lze použít až s 48 kHz)
- ▶ Bitrate: CBR i VBR při 2 kbps až 44 kbps
- ▶ Lze použít buď s obálkovým formátem ogg a nebo přímo přenášet po síti pomocí protokolu UDP nebo RTP
  - ▶ VBR kódování vs. přenos po síti
- ▶ Robustní proti výpadkům způsobeným přenosem

- ▶ Opensource ztrátový formát určený pro přenos digitalizovaného hlasu, videokonference, streamování hudby
- ▶ Nízkolatenční komprese zvuku, fixní latence 26.5ms
- ▶ Detekuje typ zvuku a míchá principy lineární predikce (nízký bitrate, SILK režim), modifikované diskrétní kosinové transformace (vysoký bitrate, CELT režim) a „hybridní režim“, kdy řeč do frekvence 8 kHz je kódována LP, zatímco řeč nad 8 kHz je kódována pomocí MDCT.

# Srovnání kvality audio kodeků

- ▶ <https://listening-test.coresv.net/results.htm>
- ▶ Opus 1.1 (opus-tools-0.1.9), AAC iTunes 11.2.2 via qaac 2.41, Ogg Vorbis aoTuV b6.03 Beta6.03, MP3 LAME 3.99.5, FAAC (libfaac)
- ▶ 40 různých stop/skladeb



# Jaké kompresní mechanismy/kodeky používat pro audio

- ▶ Není univerzální odpověď
- ▶ Nekomprimované audio jen pokud potřebujeme co nejmenší zpoždění (závisí i na počtu vzorků zpracovávaných naráz)
- ▶ Kompresní mechanismus  $\neq$  kodek, na kodeku záleží!
- ▶ Typické kompresní mechanismy dle použití
  - ▶ MPEG 1 Layer 3, FLAC pro offline archiv hudby
  - ▶ Ale např. Spotify používá Ogg Vorbis a Opus v desktopových/mobilních aplikacích a AAC na webu
  - ▶ Téměř určitě AAC pro běžnou distribuci dohromady s videem
- ▶ Bitrate typicky cca. 96 až 320 kbps pro dva kanály

*All we see or seem is but a dream within a dream.* E. A. Poe

- ▶ Pohyblivé obrázky
- ▶ Video je jen sekvence statických snímků zachycených a přehrávaných s konkrétní, stabilní frekvencí
- ▶ Temporální (časová) dimenze videa
  - ▶ Video zachycuje scénu a v ní se pohybující objekty (ev. pohybující se celou scénou)
  - ▶ Pohyb ve scéně je spojitý a je třeba jej diskretizovat
  - ▶ Vzorkování dělením spojitého pohybu na sekvenci statických snímků
- ▶ Spaciální (prostorová) dimenze videa
  - ▶ Jednotlivé statické snímky videa zachycují dvourozměrný, v čase statický obraz konkrétní scény a objektů v ní
  - ▶ Statický obraz opět nese spojitou informaci
  - ▶ Statický obraz dále diskretizujeme opět vzorkováním a kvantováním

- ▶ Soustava oko a mozek není dokonalá úplně – naštěstí
- ▶ Obrazová paměť
  - ▶ Člověk nevnímá obraz kontinuálně ale v „kvantech“
  - ▶ Obrazový vjem jsme schopni zpracovat přibližně každých 60 ms
- ▶ Oko nevnímá všechny složky obrazu stejně
  - ▶ Nejcitlivěji vnímáme jas obrazu
- ▶ Oko nevidí přirozeně zcela ostře
- ▶ Mozek má tendence si domýšlet části vjemu na základě zkušeností

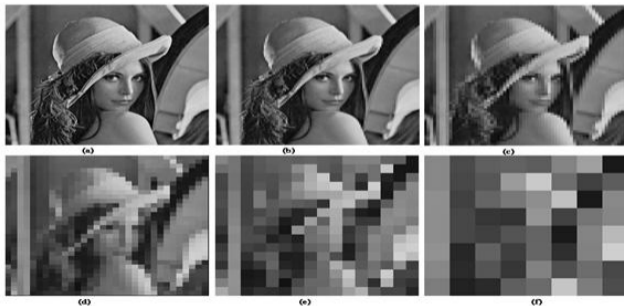
- ▶ Video (film) je sekvence statických snímků zachycených a zpět přehraných s danou snímkovou frekvencí, která má za úkol vyvolat dojem plynule pohybujících se objektů
- ▶ Vzorkování videa v temporální časové (dimenzi)
- ▶ Psychologická hranice pro udržení dojmu přirozeně se pohybujících objektů je 12 snímků za sekundu (frames per second, fps)
- ▶ Časování snímání u nejstarší techniky (a němých filmů) bylo zcela manuální (kameraman točil klikou a ručně posouval filmový pás, na kterém se exponovala jednotlivá políčka), průměrně 16 fps
- ▶ Tradičních 24 fps u filmu vychází jako kompromis mezi cenou filmového materiálu a nemožností synchronizovat zvuk s obrazem při nízkých snímkových frekvencích
- ▶ S nástupem televize se snímková frekvence začala řídit frekvencí střídavého proudu (50 Hz resp 59.94 Hz)
  - ▶ Odtud jsou odvozeny snímkové frekvence 25 fps resp. 29,97 (30) fps
- ▶ Dnes se běžně používá i 60 fps a více – věrnější vzorkování rychle se pohybujících objektů ve scéně

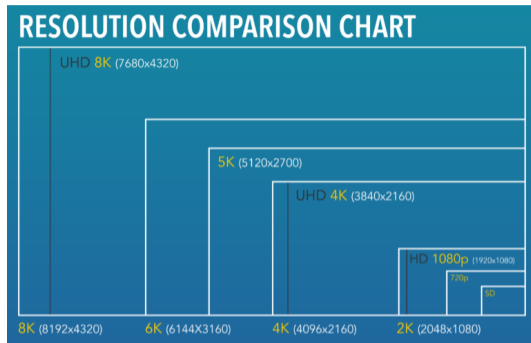


- ▶ Elektromagnetické vlnění s velmi úzkou šířkou spektra (viditelné světlo) odražené od objektů v okolí a dopadající na světlocitlivě buňky sítnice oka
- ▶ V porovnání s jakoukoliv digitální technikou je v jistých oblastech oko spolu s mozkiem velice dokonalá soustava (dynamický rozsah vidění, schopnost vyvážení bílé, rychlost ostření, noční vidění atd.)

# Vzorkování obrazu

- ▶ Obraz vzorkujeme ve dvou rozměrech konečně velkou maticí  $M \times N$  bodů (pixelů) a obrazovou informaci (barvu) v každém bodě kvantujeme do  $K$  úrovní
- ▶ Vzorkování
  - ▶ Vzorkování obrazu se řídí Shanonovou větou
  - ▶ Nejmenší detail, který chceme zachytit, v digitálním obrazu musí být minimálně dvakrát větší než velikost vzorku
  - ▶ Velikost vzorku je daná fyzickými rozměry zachycené scény a počtem pixelů v obou dimenzích



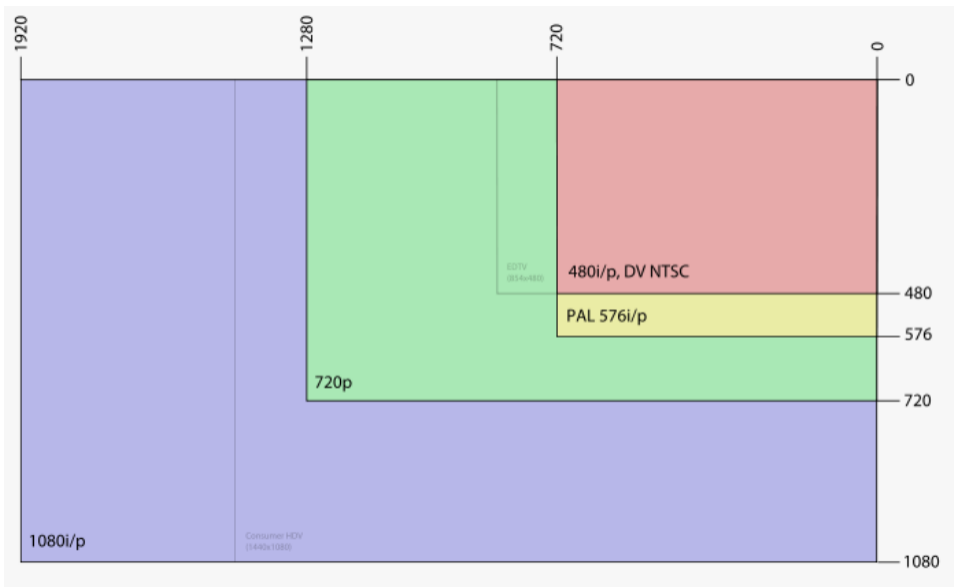


- ▶ Při daných rozměrech scény a zmenšujícím se rozlišení přicházíme o detaily v zachyceném obrazu
- ▶ Příklad: snímáme scénu o rozměrech  $8 \times 4,5$  m
  - ▶ 8K rozlišení → 1 pixel zabírá plochu  $1,04 \times 1,04$  mm
  - ▶ HD rozlišení → 1 pixel zabírá plochu  $4,17 \text{mm} \times 4,17 \text{mm}$
  - ▶ SD rozlišení (analogová TV) → 1 pixel zabírá plochu  $1,04 \times 0,78$  cm

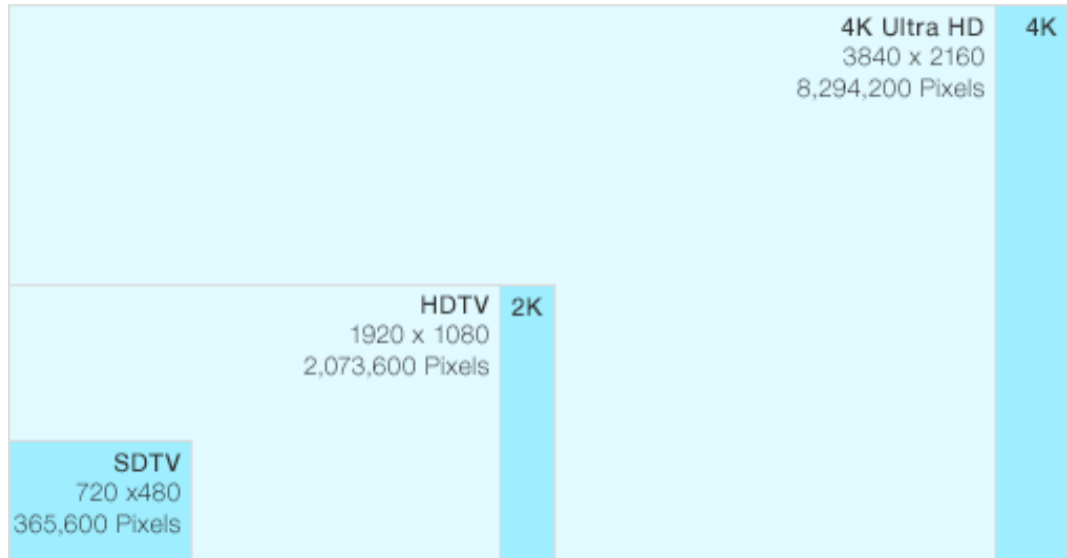
- ▶ Rodina standardů popisující rozlišení a framerate videa
- ▶ V digitálním světě:
  - ▶ PAL: 720×576 obrazových bodů, 25 fps
  - ▶ NTSC: 720×480 obrazových bodů, 29.97 fps
- ▶ VCD
  - ▶ 352×288 obrazových bodů (PAL)
  - ▶ 352×240 obrazových bodů (NTSC)
- ▶ SVCD
  - ▶ 2.7× rozlišení VCD
  - ▶ Nekonzistentní implementace rozlišení v přehrávačích

- ▶ Notace
  - ▶ Počet řádků v rozlišení
  - ▶ Progressive frames (p) nebo interlaced fields (i)
  - ▶ Počet snímků (frames) nebo polí (fields) za sekundu
- ▶ 1080p:  $1920 \times 1080$  (30p, 29.97p, 24p, 23.976p)
- ▶ 1080i:  $1920 \times 1080$  (30i, 29,97i)
- ▶ 720p:  $1280 \times 720$  (60p, 59.94p, 30p, 29.97p, 24p, 23.976p)
- ▶ Větší rozlišení 2K, 4K, 8K

# SDTV vs. HDTV

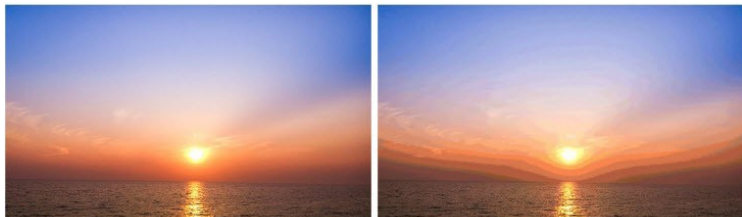


# SDTV vs. HDTV



# Kvantování obrazu

- ▶ Pro každý diskretní vzorek obrazu (pixel) potřebujeme reprezentovat jeho barvu
- ▶ Barev ve viditelném spektru je opět spojitě mnoho
- ▶ Kvantujeme je do  $K$  úrovní. Otázka je kolik úrovní potřebujeme pro věrné zachycení barev.
- ▶ Abychom dokázali reprezentaci barev uchopit, používáme barevné modely
  - ▶ Model RGB - barva se skládá ze tří složek Červená, Zelená, Modrá
  - ▶ Model YUV - barva se skládá z jasové složky a dvou barevných složek
- ▶ Každou složku reprezentujeme typicky 8, 10 (HDR) nebo 12bity (HDR)
- ▶ Při 8 bitech na barevnou složku máme k dispozici  $(2^8)^3 = 16.8mil.$  barev (může se to zdát hodně, ale nemusí to stačit)
- ▶ Při 10 bitech na barevnou složku máme k dispozici  $(2^{10})^3 = 1mld.$  barev





- ▶ RGB

- ▶ YUV

- ▶ Y – luminance (luma), jas

- ▶ U, V – chrominance

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = 0.147R + 0.289G + 0.436B = 0.492(B - Y)$$

$$V = 0.615R + 0.515G + 0.100B = 0.877(R - Y)$$

- ▶ YCrCb

- ▶ YUV + scale + offset

$$Y = 0.299R + 0.587G + 0.114B$$

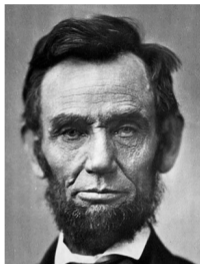
$$Cb = -0.168736R - 0.331264G + 0.5B$$

$$C = 0.5R - 0.418688G - 0.081312B$$

- ▶ Prakticky veškeré moderní kompresní mechanismy pro kompresi videa pracují s reprezentací YUV (kromě MJPEG)

# Vzorkování a kvantování obrazu

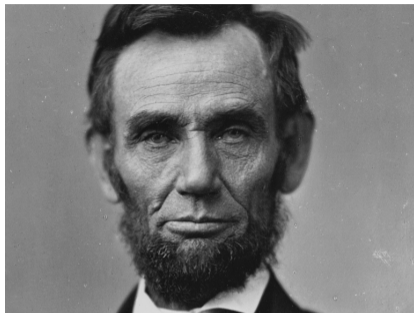
- ▶ Příklad speciálního vzorkování obrazu  $12 \times 16$  vzorky (pixely) a kvantování 8 bity (256 úrovněmi) v jednom barevném kanálu
- ▶  $12 \times 16 \times 8b = 1536b = 192B$



187	183	174	168	160	152	129	151	172	161	155	154
185	182	163	74	76	62	88	37	118	210	180	154
180	180	60	34	54	6	10	33	48	106	109	181
206	108	6	124	191	111	120	204	166	16	66	180
194	68	157	251	237	239	239	238	227	87	71	201
172	106	207	233	233	214	220	239	238	99	74	206
188	88	179	209	185	215	211	158	139	75	20	169
188	97	145	64	90	168	134	31	91	62	22	148
199	148	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	238	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	198	227	210	127	122	36	101	255	224
190	214	173	66	133	143	96	90	2	109	249	215
187	196	236	75	1	81	47	0	4	217	255	211
183	202	237	145	0	0	12	106	200	138	243	236
195	206	123	207	177	121	123	209	176	13	96	218

187	183	174	168	160	152	129	151	172	161	155	154
185	182	163	74	76	62	88	37	118	210	180	154
180	180	60	34	54	6	10	33	48	106	109	181
206	108	6	124	191	111	120	204	166	16	66	180
194	68	157	251	237	239	239	238	227	87	71	201
172	106	207	233	233	214	220	239	238	99	74	206
188	88	179	209	185	215	211	158	139	75	20	169
188	97	145	64	90	168	134	31	91	62	22	148
199	148	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	238	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	198	227	210	127	122	36	101	255	224
190	214	173	66	133	143	96	90	2	109	249	215
187	196	236	75	1	81	47	0	4	217	255	211
183	202	237	145	0	0	12	106	200	138	243	236
195	206	123	207	177	121	123	209	176	13	96	218

- ▶ 192 B je velmi málo dat na reprezentaci obrazu
- ▶ Při vzorkování  $12 \times 16$  má nejmenší detail, který jsme schopni v takovém obraze zachytit jedním pixelem rozměry cca  $4 \times 4$  cm (viz Shannonova věta)
- ▶ Kvantování v jednom barevném kanálu 8 bity umožňuje reprezentovat pouze 256 úrovní jasu/odstínů šedé

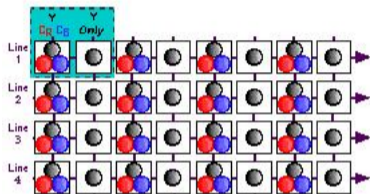


- ▶  $768 \times 576 \times 8[b/\text{pixel}(\text{greyscale})] = 3,456 \text{ kb} = 3,375 \text{ Mb} = 0,421 \text{ MB}$
- ▶ A pokud chceme pohyblivé obrázky, pak potřebujeme takových snímků  $n$
- ▶ Např. tedy 25 fps, pak bude bitrate  $86,400 \text{ kbps} = 84,375 \text{ Mbps}$
- ▶ To je už tak velmi mnoho dat a navíc framerate nemusí stačit. Můžeme s tím něco udělat?

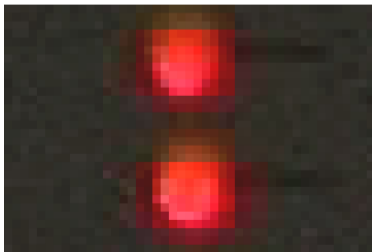
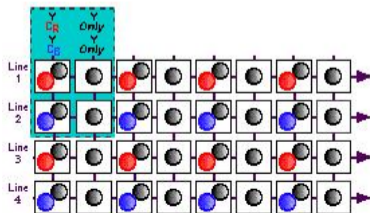
# Subsampling

- ▶ Rozlišení obrazu při akvizici jej dělí na vzorky
  - ▶ Typicky  $768 \times 576$  bodů,  $1920 \times 1080$  bodů apod.

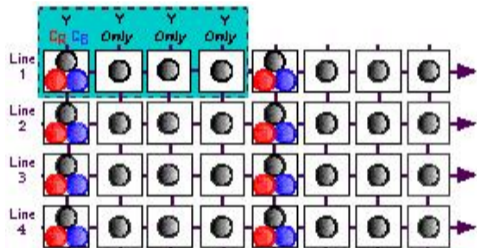
- ▶ 4:2:2



- ▶ 4:2:0



► 4:1:1



► 4:0:0 - Monochromatický obraz

► Viz <http://www.adamwilt.com/pix-sampling.html>

# Interlaced vs. progressive scan

- ▶ Progressive scan
  - ▶ Celý snímek je zaznamenaný a zobrazený najednou
- ▶ Interlaced scan
  - ▶ Snímek je temporálně rozdělen na dvě pole, zaznamenáváme a zobrazujeme jen polovinu snímku s dvojnásobnou frekvencí
  - ▶ Sudé a liché půlsnímky
  - ▶ Lepší zachycení rychle se pohybujících objektů
  - ▶ Ztráta vertikálního rozlišení



# Kolik dat tedy potřebujeme pro reprezentaci nekomprimovaného videa?

- ▶ Analogová TV (PAL):

$$768 \times 576 \times 24[b/\text{pixel}(RGB)] \times 25[\text{fps}] = 265,420 \text{ kbps} = 259 \text{ Mbps}$$

- ▶ Full HD video:

$$1920 \times 1080 \times 24[b/\text{pixel}(YUV)] \times 2/3[4:2:2\text{ sampling}] \times 30[\text{fps}] = 995,328 \text{ kbps} = 972 \text{ Mbps}$$

- ▶ 4K video:

$$3840 \times 2160 \times 30[b/\text{pixel}(YUV)][4:4:4\text{ sampling}] \times 60[\text{fps}] = 14,929,920 \text{ kbps} = 14 \text{ Gbps}$$

- ▶ 8K video:

$$7680 \times 4320 \times 24[b/\text{pixel}(YUV)] \times 2/3[4:2:2\text{ sampling}] \times 60[\text{fps}] = 31,850,496 \text{ kbps} = 31 \text{ Gbps}$$

# Nekomprimované video vs. zpracování na PC

- ▶ USB2 – 480Mbps, USB3 – 5Gbps, USB-C – 10Gbps
- ▶ Thunderbolt3 – 40Gbps
- ▶ PCIe v3 ×1 – 8Gbps, ×8 – 64Gbps, ×16 – 128Gbps
- ▶ Intel DC S3710 1.2TB SSD read – 4.4Gbps, write – 4.1Gbps
  - ▶ 4K: 643s, 8K:301s
- ▶ Síť běžně 1Gbps, 10Gbps, na PC max 100Gbps
- ▶ S PCIe v4 a aktuálními AMD procesory s velkým počtem PCIe linek se situace lepší



- ▶ Nekomprimované video typicky znamená obrovské datové objemy
  - ▶ Nejen vzhledem k propustnosti počítačových sítí ale i např. sběrnic v PC
- ▶ Bezztrátová komprese videa přináší kompresní poměry 1:5 až 1:12 – stále málo
- ▶ Ztrátová komprese (typicky H.264 nebo H.265/HEVC) přináší kompresní poměry 1:20 až 1:200 – to je již zajímavé
- ▶ Doporučený bitrate při kompresi videa pomocí kompresního mechanismu H.264 na Youtube (bitrate může být reálně i nižší pokud se pohybujeme stále ve stejných řádech).

Type	Video Bitrate, Standard Frame Rate (24, 25, 30)	Video Bitrate, High Frame Rate (48, 50, 60)
2160p (4k)	35-45 Mbps	53-68 Mbps
1440p (2k)	16 Mbps	24 Mbps
1080p	8 Mbps	12 Mbps
720p	5 Mbps	7.5 Mbps
480p	2.5 Mbps	4 Mbps
360p	1 Mbps	1.5 Mbps

#### Recommended video bitrates for HDR uploads

Type	Video Bitrate, Standard Frame Rate (24, 25, 30)	Video Bitrate, High Frame Rate (48, 50, 60)
2160p (4k)	44-56 Mbps	66-85 Mbps
1440p (2k)	20 Mbps	30 Mbps
1080p	10 Mbps	15 Mbps
720p	6.5 Mbps	9.5 Mbps

- ▶ Diskrétní kosínová transformace
  - ▶ Vychází z Fourierovy transformace, ale na rozdíl od ní jsou jejím oborem hodnot reálná čísla
  - ▶ Komplexní hodnoty Fourierovy transformace nejsou tak vhodné pro reprezentaci v PC
- ▶ Každý blok obrazu (viz makrobloky u MPEGu) převede na nekonečnou řadu frekvenčních koeficientů
- ▶ DCT není kompresní mechanismus ← bloky obrazu jsou pomocí DCT reprezentovány zcela přesně
- ▶ Výstup z DCT je dále kvantován a kódován → základ ztrátové komprese videa

# Jak funguje MPEG - I, P, B frames

- ▶ Obraz se rozloží do bloků ( $16 \times 16$ ,  $8 \times 8$ ,  $4 \times 4$ ) – tzv. Makrobloky
- ▶ V každém bloku se pomocí DCT vypočítají frekvenční koeficienty DC (jas), AC (barva)
- ▶ Frekvenční koeficienty se dále kvantují
- ▶ 3 typy snímků
  - ▶ I frame - Intrapicture, referenční snímek
  - ▶ P frame - Predicted, rozdíl oproti předchozímu snímku
  - ▶ B frame - Bidirectional predicted, interpolace mezi předchozím a následujícím snímkem
- ▶ P frame nemá smysl bez předchozího I frame
- ▶ B frame nemá smysl bez předchozího I frame a následujícího P frame
- ▶ MPEG nijak nedefinuje poměry I,P a B snímků

# Jak funguje MPEG - interframe vs. intraframe komprese

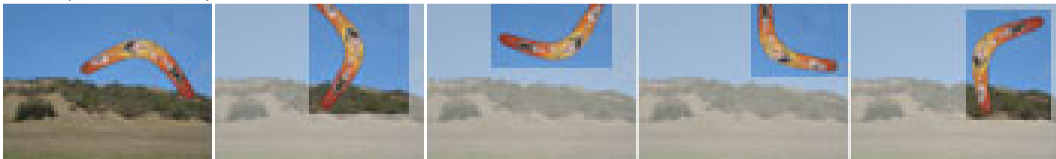
## ▶ Intraframe komprese

- ▶ Každý snímek komprimujeme zvlášť, odděleně a kompletně



## ▶ Interframe komprese

- ▶ Při kompresi můžeme s úspěchem využít vztahy mezi snímky
- ▶ V rámci stejné scény se ve snímku pohybují jen konkrétní objekty, zbytek scény je statický
- ▶ Ukládáme jen referenční snímek a dále změny mezi referenčním a dalšími (předchozími) snímky



- ▶ Motion vector
  - ▶ P a B frames
  - ▶ Motion estimation (motion compensation)
    - ▶ Mezi dvěma snímky hledáme odpovídající makrobloky
    - ▶ Vyhledávání na základě jasové složky makrobloku
  - ▶ Motion vector je čtveřice (dvoudimenzionální vektor) popisující offset souřadnic makrobloku v aktuálním snímku proti souřadnicím téhož makrobloku v referenčním snímku
    - ▶ P frame – Motion vector vzhledem k předchozímu referenčnímu snímku
    - ▶ B frame – Motion vector vzhledem k následujícímu referenčnímu snímku
- ▶ Plus navíc ukládáme tzv. difference frame kódovaný obvyklým způsobem (DCT na makroblocích plus kvantování)

- ▶ Série obrázků komprimovaná pomocí JPEGu
- ▶ Intraframe komprese → pouze I snímky
  - ▶ Vhodné pro střih
  - ▶ Malé zpoždění při kompresi videa - vhodné pro přenos po síti
- ▶ Často HW podpora (starší digitální fotoaparáty, levné USB videokamery)
  - ▶ Obvykle nízká rozlišení (320×240, 640×480)
  - ▶ 10, 12 nebo 15 fps
  - ▶ Kvalita obrazu odpovídá přibližně kvalitě „50“ nastavené u JPEGu, bitrate maximálně 1 Mbps
  - ▶ Obvykle ve spojení s obálkovým formátem AVI nebo MOV

- ▶ Celá rodina standardů, part-2 definuje kompresi videa
- ▶ Podpora pouze progressive scan videa
- ▶ Návrh počítal s kompresí jakéhokoliv obrazu až po rozlišení  $4095 \times 4095$  obrazových bodů
- ▶ Většina implementací dovolovala pouze  $352 \times 240$
- ▶ Maximální bitrate 1.5 Mbps
- ▶ VCD
  - ▶ Kvalita má být srovnatelná s VHS. MPEG-1 ale vytváří řadu viditelných obrazových artefaktů.
  - ▶ Rozlišení  $352 \times 240$  bodů (PAL) při 1150 kbps

- ▶ Part-2 rodiny standardů MPEG-2, zpětně kompatibilní s MPEG-1
- ▶ Podporuje kódování interlaced videa
- ▶ MPEG-2 kódování není optimalizované pro bitrate nižší než 1 Mbps
- ▶ I, P a B snímky, GOP - Group Of Pictures je sekvence I, P a B snímků libovolné délky. Obvykle sekvence 15 snímků I\_BB\_P\_BB\_P\_BB\_P\_BB\_P\_BB\_.
- ▶ CBR i VBR pro jednotlivé GOP (např. změnou kvantování)
- ▶ Několik profilů rozdělených do úrovní

Abbr.	Name	Frames	YUV	Streams	Comment
<b>SP</b>	Simple Profile	P, I	4:2:0	1	no interlacing
<b>MP</b>	Main Profile	P, I, B	4:2:0	1	
<b>422P</b>	4:2:2 Profile	P, I, B	4:2:2	1	
<b>SNR</b>	SNR Profile	P, I, B	4:2:0	1-2	SNR: Signal to Noise Ratio
<b>SP</b>	Spatial Profile	P, I, B	4:2:0	1-3	low, normal and high quality decoding
<b>HP</b>	High Profile	P, I, B	4:2:2	1-3	

Abbr.	Name	Pixel/line	Lines	Framerate (Hz)	Bitrate (Mbit/s)
<b>LL</b>	Low Level	352	288	30	4
<b>ML</b>	Main Level	720	576	30	15
<b>H-14</b>	High 1440	1440	1152	30	60
<b>HL</b>	High Level	1920	1152	30	80

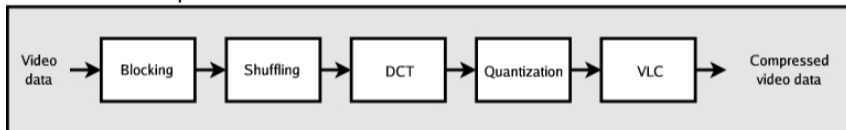


Profile @ Level	Resolution (px)	Framerate max. (Hz)	Sampling	Bitrate (Mbit/s)	Example Application
<b>SP@LL</b>	176 × 144	15	4:2:0	0.096	Wireless handsets
<b>SP@ML</b>	352 × 288	15	4:2:0	0.384	PDAs
	320 × 240	24			
<b>MP@LL</b>	352 × 288	30	4:2:0	4	Set-top boxes (STB)
<b>MP@ML</b>	720 × 480	30	4:2:0	15 (DVD: 9.8)	DVD, SD-DVB
	720 × 576	25			
<b>MP@H-14</b>	1440 × 1080	30	4:2:0	60 (HDV: 25)	HDV
	1280 × 720	30			
<b>MP@HL</b>	1920 × 1080	30	4:2:0	80	ATSC 1080i, 720p60, HD-DVB (HDTV)
	1280 × 720	60			
<b>422P@LL</b>			4:2:2		
<b>422P@ML</b>	720 × 480	30	4:2:2	50	Sony IMX using I-frame only, Broadcast "contribution" video (I&P only)
	720 × 576	25			
<b>422P@H-14</b>	1440 × 1080	30	4:2:2	80	Potential future MPEG-2-based HD products from Sony and Panasonic
	1280 × 720	60			
<b>422P@HL</b>	1920 × 1080	30	4:2:2	300	Potential future MPEG-2-based HD products from Panasonic
	1280 × 720	60			

- ▶ Typické použití:
  - ▶ DVD, DVB, HDV, HDTV

- ▶ Komprese obdobná MPEG-2 kompresi
- ▶ Kompresní mechanismus není zatížený licenčními poplatky
- ▶ Použití ve spotřební elektronice (digitální kamery, některé DVD rekordéry apod.)
- ▶ Komprese:

- ▶ Intraframe komprese



- ▶ Makrobloky  $8 \times 8$  obrazových bodů
- ▶ Shuffling zprůměruje množství obrazové informace v celém snímku promícháním makrobloků → celý snímek je komprimován stejně
- ▶ DCT - DC koeficienty (jas), AC koeficienty (barva)
- ▶ Kvantování - DC koeficienty se nemění, kvantují se pouze AC koeficienty
- ▶ Huffmanovo kódování
- ▶ Pevný datový tok 25 Mbps (pro představu cca. 3.2 MBps)

- ▶ Celá rodina různých standardů
  - ▶ ISO 14496-1 (Systems), Animace/Interaktivita (DVD menu)
  - ▶ ISO 14496-2 (Video), např.: Advanced Simple Profile (ASP), tj. implementace jako XviD, DivX5, 3ivx...
  - ▶ ISO 14496-3 (Audio), Advanced Audio Coding (AAC)
  - ▶ ISO 14496-10 (Video), Advanced Video Coding (AVC), také známé jako H.264
  - ▶ ISO 14496-14 (Container), MP4 obálkový formát (přípona .mp4)
  - ▶ ISO 14496-17 (Subtitles), MPEG-4 formát pro titulky a jejich časování
- ▶ Ne vše ze standardu MPEG-4 musí být nutně implementované

- ▶ Navržený pro rekompresi a zálohování MPEG-2 videa z DVD
- ▶ Rozlišení až do velikosti PAL při 30 fps
- ▶ Generuje I,P i B frames
- ▶ Global Motion Compensation
  - ▶ Pro velké pohyby společné dvěma snímkům se použije pouze jeden Motion vector
- ▶ Custom quantization, Adaptive quantization
  - ▶ h.263 matice - vhodná pro nižší bitraty, rozmazaný obaz
  - ▶ MPEG matice - vhodná pro vyšší bitraty
  - ▶ VBR lze nastavovat pro každý snímek
- ▶ Kodeky:
  - ▶ XviD, DivX5, DivX4/OpenDivX, ffmpeg MPEG4, Nero Digital a mnoho dalších
  - ▶ DivX3.11 (MS MPEG-4), RV9, VP6 and WMV9 nejsou zcela MPEG-4 kompatibilní
  - ▶ HW implementace - certifikace „DivX“

- ▶ Vychází z MPEG-4, návrh z roku 2003
- ▶ Kouzla H.264
  - ▶ CAVLC (Context-Adaptive Variable Length Coding)
  - ▶ CABAC (Context-Adaptive Binary Arithmetic Coding)
    - ▶ Nástroje pro kódování proudu (makrobloky + motion vektory + reference...)
    - ▶ CABAC dokáže uspořit 10-15% dat v porovnání s CAVLC, ale je to pomalá metoda
    - ▶ Obě metody jsou bezztrátové
  - ▶ Deblocking filter - po zakódování každého snímku (ale dříve než je snímek použit jako referenční) eliminuje blokování obrazu
  - ▶ Proměnná velikost makrobloku - cokoliv mezi  $16 \times 16$  a  $4 \times 4$  obrazových bodů
  - ▶ Dovoluje více referenčních snímků pro aktuální snímek při motion estimation

- ▶ Baseline Profile: I/P-Frames, progresivní scan, CAVLC
- ▶ Extended Profile: I/P/B/SP/SI-Frames, progresivní scan, CAVLC
- ▶ Main Profile: I/P/B-Frames, progresivní a interlaced scan, CAVLC nebo CABAC
- ▶ High Profile: navíc k Main Profile: 8x8 makrobloky, uživatelsky nastavitelné kvantování, bezztrátová komprese, více vzorkovacích schémat na vstupu (4:4:4...)

- ▶ MPEG-H Part 2 - technologický nástupce MPEG-4/AVC (H.264), Duben 2013
- ▶ Dvojnásobný kompresní poměr oproti MPEG-4/AVC (H.264) při zachování kvality
- ▶ Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand: Overview of the High Efficiency Video Coding (HEVC) Standard, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, December 2012, [http://iphome.hhi.de/wiegand/assets/pdfs/2012\\_12\\_IEEE-HEVC-Overview.pdf](http://iphome.hhi.de/wiegand/assets/pdfs/2012_12_IEEE-HEVC-Overview.pdf)

# High Efficiency Video Coding (HEVC/H.265)

- ▶ Proč je HEVC o tolik efektivnější?



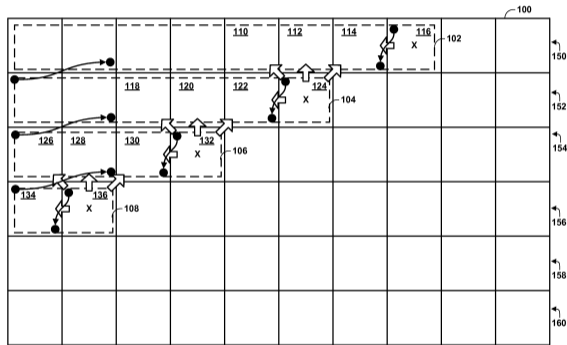
Same quality of video data, the H.265 can save 70~80% bandwidth sources.



H.265 can transfer the high quality video data(720p) at half of the bandwidth, so you can enjoy the 4K\*2K video even at a low-internet-speed environment(1~2Mbs).



- ▶ Podpora paralelního zpracování – frame paralelismus a wavefront tile paralelismus



# High Efficiency Video Coding (HEVC/H.265)

- ▶ Profile
  - ▶ Main, Main 10 a Main Still Picture

Feature	Version 1		Range extensions				
	Main	Main 10	Main 12	Main 4:2:2 10	Main 4:2:2 12	Main 4:4:4 10	Main 4:4:4 12
Bit depth	8	8 to 10	8 to 12	8 to 10	8 to 12	8 to 10	8 to 12
Chroma sampling formats	4:2:0	4:2:0	4:2:0	4:2:0/4:2:2	4:2:0/4:2:2	4:2:0/4:2:2/4:4:4	4:2:0/4:2:2/4:4:4
4:0:0 (Monochrome)	No	No	Yes	Yes	Yes	Yes	Yes
High precision weighted prediction	No	No	Yes	Yes	Yes	Yes	Yes
MinCR reduced to half its base value	No	No	No	Yes	Yes	Yes	Yes
Inter color component prediction	No	No	No	No	No	Yes	Yes
Intra block copy	No	No	No	No	No	Yes	Yes
Intra smoothing disabling	No	No	No	No	No	Yes	Yes
Residual DPCM inter/intra	No	No	No	No	No	Yes	Yes
Transform skip block sizes larger than 4x4	No	No	No	No	No	Yes	Yes
Transform skip context/rotation	No	No	No	No	No	Yes	Yes
Extended precision processing	No	No	No	No	No	No	No
Separate color plane	No	No	No	No	No	No	No

# High Efficiency Video Coding (HEVC/H.265)

## ► Široká škála úrovní

Level	Max luma sample rate (samples/s)	Max luma picture size (samples)	Max bit rate for Main, Main 10, and Main 12 profiles (kbit/s) <sup>[A]</sup>		Example picture resolution @ highest frame rate <sup>[B]</sup> (MaxDpbSize <sup>[C]</sup> ) <a href="#">Click here to hide/show additional examples</a>
			Main tier	High tier	
1	552,960	36,864	128	–	128×96@33.7 (6) 176×144@15.0 (6)
2	3,686,400	122,880	1,500	–	176×144@100.0 (16) 352×288@30.0 (6)
2.1	7,372,800	245,760	3,000	–	352×288@60.0 (12) 640×360@30.0 (6)
3	16,588,800	552,960	6,000	–	640×360@67.5 (12) 720×576@37.5 (8) 960×540@30.0 (6)
3.1	33,177,600	983,040	10,000	–	720×576@75.0 (12) 960×540@60.0 (8) 1280×720@33.7 (6)
4	66,846,720	2,228,224	12,000	30,000	1,280×720@68.0 (12) 1,920×1,080@32.0 (6) 2,048×1,080@30.0 (6)
4.1	133,693,440		20,000	50,000	1,280×720@136.0 (12) 1,920×1,080@64.0 (6) 2,048×1,080@60.0 (6)
5	267,386,880	8,912,896	25,000	100,000	1,920×1,080@128.0 (16) 3,840×2,160@32.0 (6) 4,096×2,160@30.0 (6)
5.1	534,773,760		40,000	160,000	1,920×1,080@256.0 (16) 3,840×2,160@64.0 (6) 4,096×2,160@60.0 (6)
5.2	1,069,547,520		60,000	240,000	1,920×1,080@300.0 (16) 3,840×2,160@128.0 (6) 4,096×2,160@120.0 (6)
6	1,069,547,520	35,651,584	60,000	240,000	3,840×2,160@128.0 (16) 7,680×4,320@32.0 (6) 8,192×4,320@30.0 (6)
6.1	2,139,095,040		120,000	480,000	3,840×2,160@256.0 (16) 7,680×4,320@64.0 (6) 8,192×4,320@60.0 (6)
6.2	4,278,190,080		240,000	800,000	3,840×2,160@300.0 (16) 7,680×4,320@128.0 (6) 8,192×4,320@120.0 (6)

- ▶ Jeden z nejmladších kompresních mechanismů z léta 2004
- ▶ Opensource návrh i implementace
- ▶ Přímá konkurence pro MPEG-4 kompresní mechanismy
- ▶ Vlastnosti:
  - ▶ Ztrátová komprese
  - ▶ Makrobloky  $8 \times 8$  obrazových bodů
  - ▶ Podpora pro vzorkování 4:2:0, 4:2:2, a 4:4:4 (MPEG-4 ASP podporuje pouze 4:2:0)
  - ▶ Podpora motion compensation jen na úrovni jednotlivých bloků
  - ▶ Podpora VBR kódování
  - ▶ Obsahuje pouze I a P frames
  - ▶ Dovoluje více referenčních snímků pro aktuální snímek při motion estimation

- ▶ Standard SMPTE implementovaný Microsoftem jako Windows Media Video 9, alternativa ke klasickým DCT based MPEG kodekům
- ▶ Simple, Main a Advanced profile
- ▶ Podpora kódování interlaced videa bez nutnosti převodu do progressive režimu

	Simple	Main	Advanced
Baseline intra frame compression	Yes	Yes	Yes
Variable-sized transform	Yes	Yes	Yes
16-bit transform	Yes	Yes	Yes
Overlapped transform	Yes	Yes	Yes
4 motion vector per macroblock	Yes	Yes	Yes
¼ pixel luminance motion compensation	Yes	Yes	Yes
¼ pixel chrominance motion compensation	No	Yes	Yes
Start codes	No	Yes	Yes
Extended motion vectors	No	Yes	Yes
Loop filter	No	Yes	Yes
Dynamic resolution change	No	Yes	Yes
Adaptive macroblock quantisation	No	Yes	Yes
B frames	No	Yes	Yes
Intensity compensation	No	Yes	Yes
Range adjustment	No	Yes	Yes
Field and frame coding modes	No	No	Yes
GOP Layer	No	No	Yes
Display metadata	No	No	Yes
	Simple	Main	Advanced

Profile	Level	Maximum Bit Rate	Resolutions by Framerate
Simple	Low	96 kbit/s	176 x 144 / 15 (QCIF)
	Medium	384 kbit/s	240 x 176 / 30 352 x 288 / 15 (CIF)
Main	Low	2 Mbit/s	320 x 240 / 24 (QVGA)
	Medium	10 Mbit/s	720 x 480 / 30 (480p) 720 x 576 / 25 (576p)
	High	20 Mbit/s	1920 x 1080 / 30 (1080p)
Advanced	L0	2 Mbit/s	352 x 288 / 30 (CIF)
	L1	10 Mbit/s	720 x 480 / 30 (NTSC-SD) 720 x 576 / 25 (PAL-SD)
	L2	20 Mbit/s	720 x 480 / 60 (480p) 1280 x 720 / 30 (720p)
	L3	45 Mbit/s	1920 x 1080 / 24 (1080p) 1920 x 1080 / 30 (1080i) 1280 x 720 / 60 (720p)
	L4	135 Mbit/s	1920 x 1080 / 60 (1080p) 2048 x 1536 / 24

- ▶ Expedimentální waveletová komprese
- ▶ Výborná kvalita obrazu při nízkých bitratech v porovnání s klasickými kodeky
- ▶ Analogový signál se reprezentuje pomocí oscilující vlny konečné délky
- ▶ Obecně vhodnější pro reprezentaci obrazu s vysokofrekvenčními složkami (extrémní příklad: hvězdy na noční obloze)
- ▶ Vysokofrekvenční složky lze reprezentovat menším množstvím dat než v případě DCT transformace
- ▶ Další wavelet compression based kodeky: Tarkin (Ogg), Dirac (BBC), Pixlet (Apple)

- ▶ Závislé na mnoho faktorech
- ▶ Objem komprimovaných dat (rozlišení  $\times$  reprezentace barevného prostoru  $\times$  framerate) vs. bitrate pro kodek
  - ▶ Lze zmenšit rozlišení
  - ▶ Obvykle nelze zmenšit framerate
  - ▶ Zmenšení rozlišení = větší kvalita obrazu při stejném bitrate. Ale horší kvalita reprodukce  $\leftarrow$  video obvykle zobrazujeme na obrazovce s větším rozlišením. Je nutné škálovat obraz  $\rightarrow$  horší kvalita obrazu.
  - ▶ Bitrate je obvykle daný kapacitou média  $\leftarrow$  600 - 1000 kbps pro film (1 - 2 hodiny) ukládaný na CD.
- ▶ Nastavení poměru I,P a B frames a GOP
  - ▶ Problémy s přehráváním
  - ▶ Neúčinné VBR u MPEG-2

# Srovnání kompresních mechanismů/kodeků

Features	Standards (Profiles)							Non Std
	MPEG-1	MPEG-2	MPEG-4 ASP	MPEG-4 AVC Main	MPEG-4 AVC High	VC-1	Ogg Theora	SNOW
<b>B-Frames</b>	✓	✓	✓	✓	✓	✓		
<b>Slices</b> Error Resilience	✓	✓	✓	✓	✓	✓		
<b>Interlace</b> (PAFF/MBAFF)		✓	✓	✓	✓	✓		
<b>Entropy Encoding</b>	Huffman	Huffman	Huffman	Exp-Golomb or Adaptive Arithmetic	Exp-Golomb or Adaptive Arithmetic	Huffman and Bitplane	Adaptive Huffman	Adaptive Range
<b>Motion Block Size</b>	16x16	16x16	16x16/8x8	16x16/16x8 8x8/8x4/4x4	16x16/16x8 8x8/8x4/4x4	16x16/8x8 8x4 or 4x4	16x16/8x8	32x32/16x16 or 8x8
<b>Motion Search Precision</b>	One Pixel or Half Pixel	Half Pixel	Half Pixel or Quarter Pixel	Quarter Pixel	Quarter Pixel	Half Pixel or Quarter Pixel	Half Pixel	Half Pixel or Quarter Pixel
<b>OBMC</b>								✓
<b>GMC</b> (Global Motion Compensation)			✓					
<b>Intra Prediction</b>	DC 8x8	DC 8x8	AC 8x8	Spatial 16x16/4x4	Spatial 16x16/8x8 or 4x4	AC 8x8	DC 8x8	DC 32x32/16x16 or 8x8
<b>Spatial Transform</b>	DCT 8x8	DCT 8x8	DCT 8x8	HCT 4x4	HCT 8x8/4x4	8x8/8x4/4x4	DCT 8x8	Wavelet 13/7, 9/7 or 5/3
<b>Bit Exact Decoding</b>				✓	✓	✓	✓	✓
<b>Lossless Mode</b>					✓			✓
<b>Custom Quantizer Matrix</b>	✓	✓	✓		✓		✓	
<b>Look-Ahead Filtering</b>				✓	✓	✓	✓	



# Srovnání kompresních mechanismů/kodeků

- ▶ doom9.org annual codecs shootouts
- ▶ MSU Video Group - <http://www.compression.ru/video/>
- ▶ Bitrate handling
- ▶ Výkon kodeku při kompresi (fps)
- ▶ Kvalita kódování (závisí i na kvalitě implementace kodeku)
  - ▶ Objektivní PSNR metriky
    - ▶ Peak signal to noise raio
    - ▶ Poměr mezi maximální možnou silou signálu a silou šumu vzniklého kompresí
  - ▶ Subjektivní vizuální hodnocení
  - ▶ Výsledky obou metod nemusí korelovat
- ▶ [http://www.compression.ru/video/codec\\_comparison/pdf/MSU\\_MPEG4\\_Comparison\\_eng.pdf](http://www.compression.ru/video/codec_comparison/pdf/MSU_MPEG4_Comparison_eng.pdf)
- ▶ <http://www.doom9.org/index.html?/codecs-quali-105-1.htm>
- ▶ <http://www.doom9.org/index.html?/codecs-main-105-1.htm>

## ▶ AVI

- ▶ Audio Video Interleave
- ▶ Obálkový formát MS z roku 1992
- ▶ FourCC identifikace kodeku
- ▶ AVI vs. AVI 2.0
  - ▶ Více audio a video stop
  - ▶ Podpora pro stopy větší než 2 GB
- ▶ Dnes poněkud obsoletní, ale stále velmi používaný
  - ▶ Nepodporuje titulky a moderní kodeky (Vorbis apod.)
  - ▶ Existují různé hacky nekompatibilní s většinou přehrávačů

## ▶ MOV

- ▶ Obálkový formát pro QuickTime
- ▶ Jedna nebo více stop pro audio, video, efekty nebo text (titulky)
- ▶ Quicktime reference – odkaz např. na video, které je uložené kdekoli jinde na disku nebo na síti
- ▶ Vhodné pro střih

## ▶ OggMedia

- ▶ Nezaměňovat s Ogg - formát proudu dat pro uložení audia nebo videa komprimovaného kodeky Vorbis, Theora apod.
- ▶ OggMedia rozšiřuje možnosti formátu Ogg
  - ▶ Podpora kapitol
  - ▶ Více stop s titulky
  - ▶ Více audio stop různých formátů (MP3, AC3, Vorbis, WAV)

## ▶ Matroska

- ▶ Opensource návrh obálkového formátu, vystavěný na binárním ekvivalentu XML (EBML)
- ▶ Rychlé procházení audiem nebo videem (seek)
- ▶ Implementuje menu jako u DVD
- ▶ Podpora pro streaming

- ▶ MPEG-4 (MP4)
  - ▶ Vychází z obálkového formátu Apple Quicktime
  - ▶ Audio, video plus MPEG-4 titulky
  - ▶ Pomocí tzv. private stream lze do obálky MP4 vložit téměř jakákoliv data (např. vobsub titulky apod)
- ▶ DivX6
  - ▶ Interaktivní menu
  - ▶ Více stop s titulky
  - ▶ Více audio stop
  - ▶ Kapitoly
  - ▶ Další textová metadata
  - ▶ Podpora různých audio a video formátů
- ▶ a další...

- ▶ Proč multimédia a datové sítě?
  - ▶ Dobrý zdroj dat, relativně velké objemy, specifické nároky na samotný přenos
- ▶ Aplikace multimediálních přenosů
  - ▶ Streaming
  - ▶ Videokonference
    - ▶ aplikace požadující zcela konkrétní vlastnosti přenosu multimediálních dat (např. end-to-end zpoždění)
    - ▶ požadavky na přenos zásadně ovlivňují možnosti zpracování multimediálních dat

- ▶ Objem komprimovaných dat (rozlišení  $\times$  reprezentace barevného prostoru  $\times$  framerate) vs. kapacita sítě vs. rychlost kódování multimedialního streamu vs. nároky na zpoždění přenosu
  - ▶ Typicky malý bitrate (řádově max. jednotky Mbps), ačkoliv pro kvalitní přenosy se používá bitrate v řádu desítek i stovek Mbps
  - ▶ Lze snížit rozlišení
  - ▶ Lze snížit framerate – u videokonferencí není framerate tolik podstatný
- ▶ Problematické použití VBR
- ▶ Nemá smysl používat B frames, opatrně např. i s délkou GOP

- ▶ Stavový protokol na transportní vrstvě ISO/OSI modelu
- ▶ Vlastnosti významné pro multimediální přenosy
  - ▶ Bezchybný přenos
    - ▶ Retransmise ztracených paketů
    - ▶ Pakety vždy dorazí ve správném pořadí
  - ▶ Kontrola zahlcení linky
  - ▶ Férový protokol
- ▶ Nevýhody TCP pro multimediální přenosy
  - ▶ Bezchybnost přenosu je na úkor nízké latence
  - ▶ Férovost nedovoluje dostatečnou šířku pásma na vytížených linkách

- ▶ Bezstavový protokol na transportní vrstvě ISO/OSI modelu
- ▶ Nespolehlivý protokol
  - ▶ Pakety mohou přicházet mimo původní pořadí
  - ▶ Pakety se mohou ztratit bez jakéhokoliv upozornění
- ▶ Ale odpadá režie s ověřováním, ze každý paket dorazil v pořádku a hlavně s retransmisemi
- ▶ V porovnání s TCP minimalistický, efektivnější a rychlejší
- ▶ UDP prakticky nezvyšuje latenci při přenosu multimediálních dat
- ▶ Multimediální aplikace využívají v drtivé většině případů protokol UDP pro přenos dat (až na speciální případy)



## ▶ RTP

- ▶ Real-Time Transport Protocol
- ▶ Postavený nad protokolem UDP
- ▶ Klíčové vlastnosti
  - ▶ Identifikace obsahu
  - ▶ Sekvenční číslování paketů
  - ▶ Časové značky pro jednotlivé pakety
- ▶ Protokol sám od sebe nezaručuje kvalitu přenosu, pouze poskytuje prostředky pro zaručení kvality aplikacím

## ▶ RTCP

- ▶ RTP Control Protocol (RTCP)
- ▶ Real time control protocol doplňuje protokol RTP
- ▶ Poskytuje out-of-band informace pro řízení proudu dat přenášeného pomocí RTP
- ▶ RTCP poskytuje aplikaci zpětnou vazbu na kvalitu přenosu pomocí protokolu RTP

## ▶ RTSP

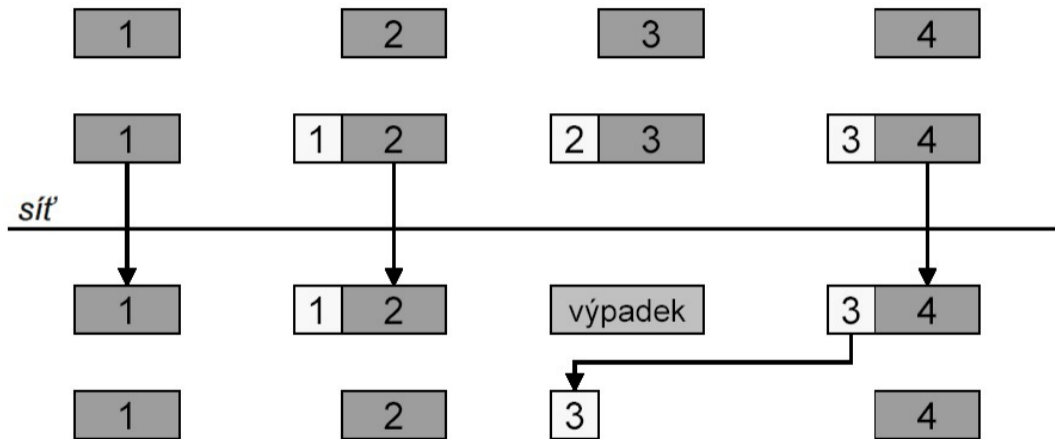
- ▶ Real-time Streaming Protocol
- ▶ Stavový protokol založený na HTTP požadavcích (GET apod.)
- ▶ Ovládání streaming serveru (VCR příkazy jako Play, Pause a Stop) a přístup k souborům podle času
- ▶ Pro přenos dat se používá protokol RTP + RTCP případně jeho proprietární obdoba RDT
- ▶

## ▶ MMS

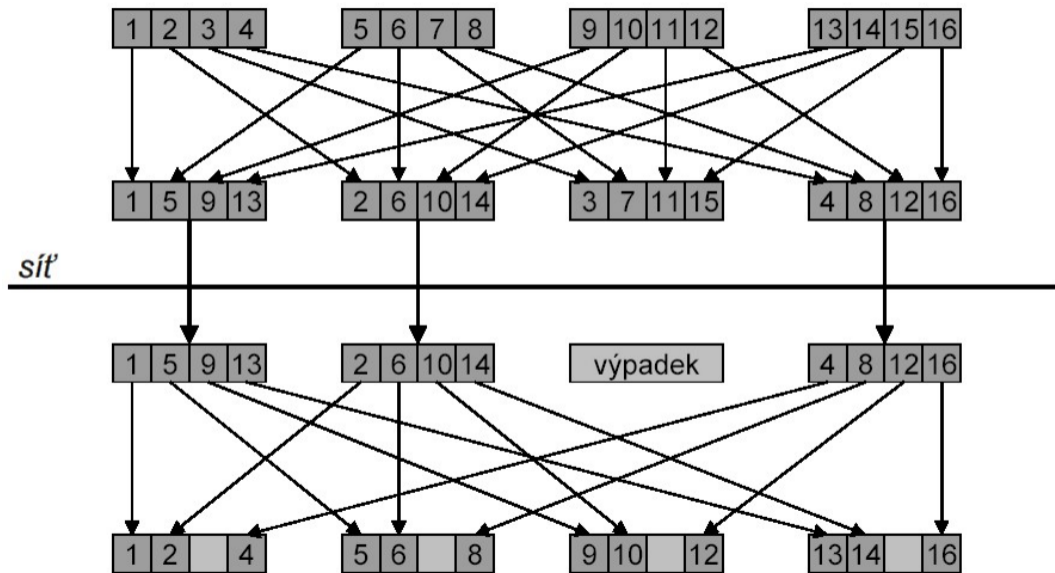
- ▶ Microsoft Media Services nebo také Netshow services
- ▶ Proprietární protokol pro streaming
- ▶ Pro přenos dat se používají protokoly UDP nebo i TCP pokud se nezdaří vyjednat spojení na protokolu UDP
- ▶ Jako poslední z možností je "streaming" pomocí upraveného protokolu HTTP (tedy opět nad protokolem TCP)

- ▶ Nutnější hlavně u zvuku, používá se samozřejmě i u přenosu obrazu
- ▶ Buffery
- ▶ Forward Error Correction (FEC)
  - ▶ XORování
  - ▶ posílání druhého proudu (v nižší kvalitě)
  - ▶ prokládání (interleaving)
  - ▶ oprava chyb na straně klienta
    - ▶ nahrazení daty z předchozího paketu
    - ▶ interpolace

# Posílání druhého proudu



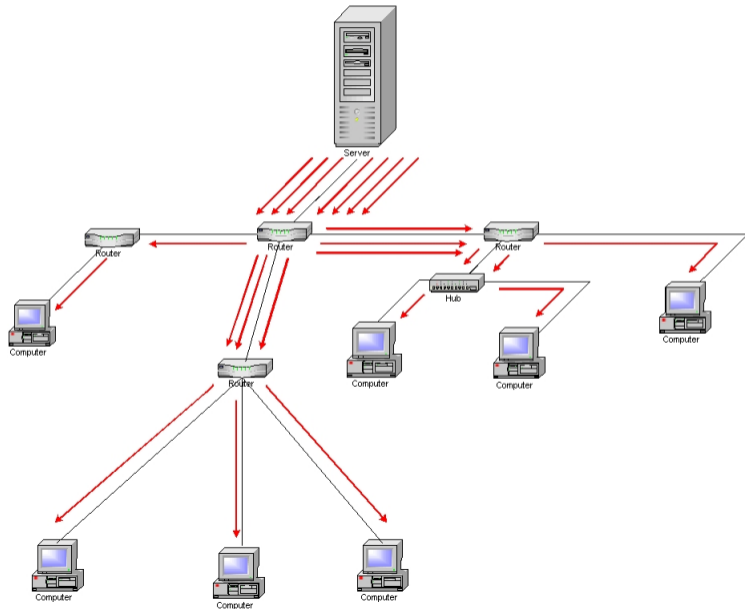
# Interleaving



# Point-to-point vs. multipoint

- ▶ Point-to-point
- ▶ Multipoint
  - ▶ 1:N
    - ▶ Rozšíření point-to-point schématu
    - ▶ Streaming – VOD
    - ▶ Streaming – push schéma
    - ▶ Multimediální stream se může šířit sítí v mnoha kopiích a zahlcovat ji
  - ▶ M:N
    - ▶ Typicky videokonference
- ▶ Problémy šíření multimediálních streamů
  - ▶ Firewally
  - ▶ Nat

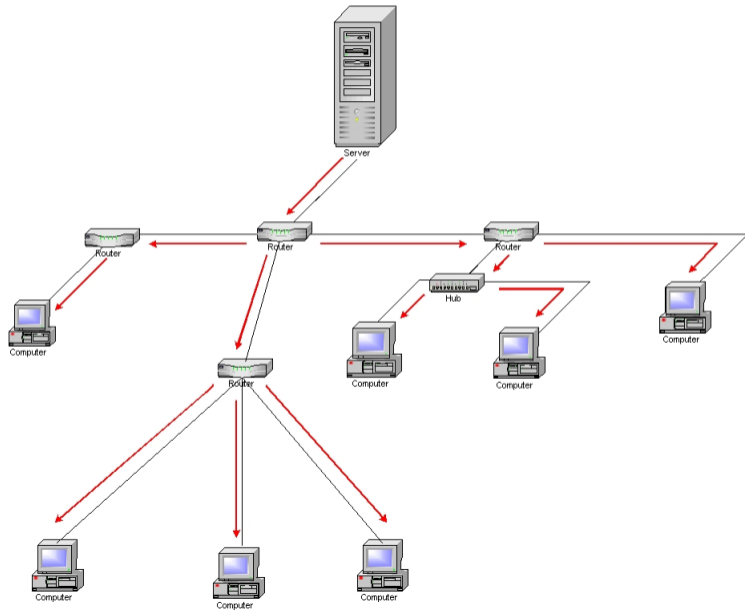
# Unicast



- ▶ Efektivní schéma pro posílání multimediálních dat
- ▶ Routery vytvářejí optimální strom cest po kterých se šíří multimediální data
- ▶ Postavený na protokolu UDP (nad TCP nemá smysl, TCP vytváří spojení mezi dvěma konkrétními uzly)
- ▶ Relativně nespolehlivé schéma
  - ▶ Multicast se v nešíří napříč všemi sítěmi
  - ▶ Bezpečnostní rizika



# Multicast

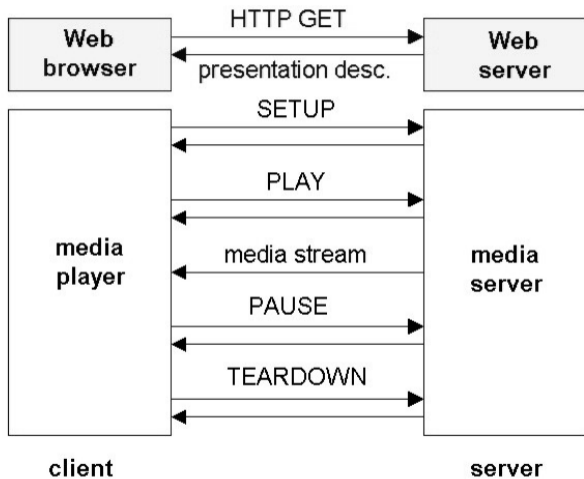


- ▶ SW který přijímá multimediální streamy od jednotlivých klientů a přeposílá je ostatním připojeným klientům
- ▶ Vytváří překryvovou síť, která emuluje multicast v síti, kde se multicast nešíří
  - ▶ Neřeší problém redundance multimediálních streamů na jednotlivých linkách
- ▶ Možná schémata použití – 1:N, M:N

- ▶ Streaming
  - ▶ Způsob doručení multimediálního obsahu klientům prostřednictvím sítě
  - ▶ Přidaná hodnota porovnání s prostým stažením multimediálního obsahu
  - ▶ Live streaming
    - ▶ Doručování multimediálního obsahu, který vzniká živě během streamování
  - ▶ Video on Demand vs. pasivní příjem
    - ▶ Pasivní příjem se obvykle používá pro příjem živých streamů
    - ▶ Je samozřejmě možné streamovat i multimediální archivy
  - ▶ Video a audio nelze kódovat libovolně.
- ▶ Videokonference (přenosy s nízkým zpožděním)
  - ▶ Jednoznačný požadavek na interaktivitu
  - ▶ V porovnání se streamingem přináší další omezující požadavky na zpracování videa a audia.

- ▶ Streaming
  - ▶ Díky jednosměrnosti provozu můžeme data bufferovat
  - ▶ Latence při přenosu vznikající při kompresi videa není problém
  - ▶ Latence vznikající přenosem v síti a její rozptyl také není podstatná - lze řešit bufferem
- ▶ Videokonference (přenosy s nízkým zpožděním)
  - ▶ Při přenosu nelze používat buffery ani na straně odesílajícího ani na straně příjemce - vyžadujeme interaktivitu a tedy nízké latence
  - ▶ Potřeba využívat kodeky s nízkou latencí
  - ▶ Latence a její rozptyl při přenosu sítí je také velmi problematická

# Streaming



- ▶ Kompresní mechanismy
  - ▶ Nejsme limitováni nutností udržet nízkou end-to-end latenci → z tohoto hlediska lze použít prakticky libovolný kodek
  - ▶ Kompresi musí být realtime což diskvalifikuje zejména vaveletovou kompresi ale i některé pokročilé MPEG profily
  - ▶ Obvykle pouze CBR kódování - u VBR nejsme dobře schopni předvídat, zda nepřekročíme bitrate daný dostupným pásmem
  - ▶ Audio – MP3, Vorbis, AAC nebo Opus
  - ▶ Video – H.264, HEVC, VP8 nebo VP9
- ▶ Obálkové formáty
  - ▶ Zapouzdření více proudů videa a audia
  - ▶ Metadata
  - ▶ Podpora pro zotavení z chyb způsobených přenosem
  - ▶ Adaptace na změny parametrů přenosových linek
  - ▶ MP4, FLV, WebM nebo ASF

- ▶ Obvykle varianta H.263, případně MJPEG, MPEG4 AVC
- ▶ Audio ve formátu PCM, ADPCM nebo MP3, AAC
- ▶ Široká podpora v přehrávačích napříč platformami (ne jen Macromedia Flash player)
- ▶ Streaming pomocí proprietárního Real Time Messaging Protocol (RTMP) protokolu od Adobe a Flash Media serveru
- ▶ Progressive download
  - ▶ Přenos protokolem HTTP → neblokované firewally
  - ▶ Libovolný přístup k videu ← není nutné přehrávat sekvenčně
  - ▶ Buffer na straně klienta
  - ▶ Neporadí si s kolísající šířkou pásma a s nižší šířkou pásma než je bitrate videa

- ▶ Advanced Systems Format, dříve Advanced Streaming Format
- ▶ Proprietární obálkový formát Microsoftu, podpora streamování
- ▶ Podpora pro přehrávání obsahu ze streaming serveru, HTTP serveru nebo z lokálního disku
- ▶ Specifikuje strukturu pro ukládání audia a videa a přístup k jednotlivým multimediálním proudům
- ▶ Nespecifikuje konkrétní formáty pro kódování audia nebo videa, ale obvykle se používá spolu s Windows Media Video resp. Windows Media Audio
- ▶ Implementuje techniky pro korekci chyb vzniklých během přenosu
- ▶ Podpora DRM (pouze ve spojení s WMW nebo WMA)



- ▶ Základní přenos RTP + RTCP a RTSP
- ▶ Real Time Messaging Protocol – RTMP
- ▶ Apple HTTP Live Streaming – HLS
- ▶ Dynamic Adaptive Streaming over HTTP – MPEG-DASH
- ▶ Posun klientů od samostatných aplikací k webovému prohlížeči
- ▶ Podpora v HTML5

- ▶ Proprietární protokol Macromedia/Adobe
- ▶ TCP přenos dat (včetně šifrování a enkapsulace do HTTP požadavků)
- ▶ Fragmentace dat – 64B pro audio, 128B pro video
- ▶ V rámci RTMP sezení několik nezávislých kanálů pro přenos RPC, audia, videa apod.)
- ▶ Není podpora v HTML5, pouze Adobe Flash

- ▶ Všechny dotazy a přenos multimediálních dat probíhají nad protokolem HTTP
- ▶ Možnost přizpůsobení datového toku aktuální propustnosti sítě
- ▶ Videa (H.264 video a MP3, AAC nebo AC-3 audio) se zakódují do MPEG-TS transportu, který je určen pro přenos multimediálních dat
- ▶ Každé takové video se následně rozdělí na malé segmenty, které umožňují například posouvání ve videu bez nutnosti čekání na stažení celého videa
- ▶ Adaptivní bitrate
- ▶ Klient si dle vlastního uvážení stahuje segmenty a skládá z nich původní média
- ▶ Přímá podpora v HTML5

- ▶ Adaptivní technologie
- ▶ Média jsou rozdělena na rámce
- ▶ Přenos pomocí HTTP
- ▶ Technologie agnostická vůči zpracování audia/video
- ▶ MPEG4 nebo MPEG-TS kontejner
- ▶ Není přímo podporovaný v HTML5, ale existují knihovny v JS

# Zpoždění při zpracování zvuku

- ▶ Zvuk zpracováváme v řetězci operací (pipeline)
- ▶ Digitalizace zvukového signálu zvukovou kartou → komprese zvuku (→ síťový přenos) → dekomprese zvuku → převod zvukového signálu zpět do analogové podoby na zvukové kartě
- ▶ Zpoždění zpracování audia v celé pipeline je tzv. end-to-end zpoždění
- ▶ Zvukový signál nemůžeme zpracovávat po jednotlivých vzorcích
  - ▶ Při vzorkovací frekvenci 48 kHz je vzorkovací interval dlouhý 20  $\mu$ s
  - ▶ Časování zpracování vzorků zvuku musí být velmi přesné, každá nepřesnost je slyšitelná
  - ▶ Vzorky, které nestihneme zpracovat v daném čase, můžeme už pouze zahodit, protože už potřebujeme zpracovávat další vzorky zvuku
  - ▶ Tak přesného časování nejsme schopni na procesoru s běžným operačním systémem dosáhnout
- ▶ Proto každá operace v pipeline bufferuje a zpracovává více vzorků naráz
- ▶ Obvykle 64, 128 nebo 265 vzorků audia naráz → zpoždění každé operace je 1,3 ms až 5,3 ms
- ▶ Zpoždění pipeline pro zpracování zvuku tedy může tedy být i cca. 20 ms (bez zpoždění síťového přenosu)

# Zpoždění při zpracování obrazu

- ▶ Zpoždění nabíráme v celém řetězci počínaje kamerou (zdrojem videa), přes akvizici obrazu a zvuku až po přehrávání obrazu a jeho zobrazování na displeji
  - ▶ Kamera čas daný vzorkovacím intervalem mezi jednotlivými snímky na expozici statického obrazu
  - ▶ Při 60 fps je vzorkovací interval  $1/60\text{ s} = 16,6\text{ ms}$ . Při 25 fps je vzorkovací interval  $1/25\text{ s} = 40\text{ ms}$ 
    - ▶ Pokud není scéna dostatečně nasvícena, jsou to vlastně relativně krátké expoziční časy
  - ▶ Video dále zpracováváme typicky po celých snímcích v rámci posloupnosti operací (pipeline)
  - ▶ Typicky akvizice snímku → komprese snímku (→ síťový přenos) → dekomprese snímku → zobrazení snímku na grafické kartě (výstupní kartě pro zpracování videa → zobrazení snímku na displeji
  - ▶ Doba maximálního trvání každé operace je daná délkou vzorkovacího intervalu mezi jednotlivými statickými snímky

# Zpoždění při zpracování obrazu

- ▶ Akvizice jednoho statického, digitálního snímku z kamery trvá celý vzorkovací interval
- ▶ Komprese a dekomprese je typicky výrazně rychlejší, k síťovému přenosu se ještě dostaneme
- ▶ Zobrazení snímku na grafické kartě nebo kartě pro výstup videa trvá opět celý vzorkovací interval
- ▶ Zpoždění displeje může být řádově jednotky milisekund u herních displejů až po desítky milisekund
- ▶ Tj. při zpracování videa po celých snímcích bude zpoždění větší než  $2 \times 1/fps$
- ▶ Pokud je vzorkovací interval 40 ms (25 fps), pak bude zpoždění při zpracování obrazu minimálně 80 ms, neobvyklé není ani zpoždění 250 ms (1/4 s)
  - ▶ I komprese a dekomprese každého snímku může trvat až 40 ms, zobrazování bývá také pomalejší

# Vadí nám takové zpoždění?

- ▶ Typicky očekáváme synchronní obraz a zvuk
- ▶ Pokud obraz a zvuk explicitně nesynchronizujeme, bude se jejich zpoždění rozcházet
- ▶ Rozdíl zpoždění  $> 100$  ms je již viditelný in na desynchronizaci řeči a pohybu rtů mluvčího ve videu (lip-sync)
- ▶ V případě hudby a například záběrů prstů na klaviatuře klavíru vadí i rozdíl zpoždění v řádu malých desítek ms
- ▶ Mezinárodní telefonní unie ITU-T stanovuje, že maximální end-to-end zpoždění v telefonii je 150 ms
- ▶ Při zpoždění nad 300 ms už může být nepříjemné vést delší dobu normální rozhovor



# Zpoždění přenosu v počítačových sítích

- ▶ Zpoždění je způsobeno jednak fyzikálně při přenosu signálu (bezdrátově, v metalických kabelech i v optice) a zpožděním zpracování paketů v síťových prvcích
- ▶ Rychlost šíření signálu je shora omezená rychlostí (pomalostí světla)
  - ▶ Cca 300000 km/s
  - ▶ Ale např. nejkratší vzdálenost mezi Brnem a San Diegem je 9825 km tj. zpoždění přenosu signálu (i po optice) bude cca  $0,033\text{ s} = 33\text{ ms}$
  - ▶ Reálná délka všech optických kabelů na trase je mnohem větší (délky transatlantických optických kabelů se pohybují mezi 6000 a 13000 km)
- ▶ Zpoždění jednotlivých síťových prvků je v řádu desítek mikrosekund
- ▶ Reálné zpoždění vysokorychlostních sítí při přenosech v rámci ČR je okolo 2 ms
- ▶ Zpoždění kabelového připojení v síti Vodafone (UPC) je při přenosech v rámci ČR jen okolo 3 ms
- ▶ Ale při použití telefonních linek (ADSL v síti O2) a domácí Wi-fi už cca 20 ms
- ▶ A reálné zpoždění vysokorychlostí sítě při přenosu z Brna do San Diega bude okolo 85 ms
- ▶ Zpoždění síťového přenosu je potřeba přičíst ke zpoždění zpracování obrazu a zvuku!

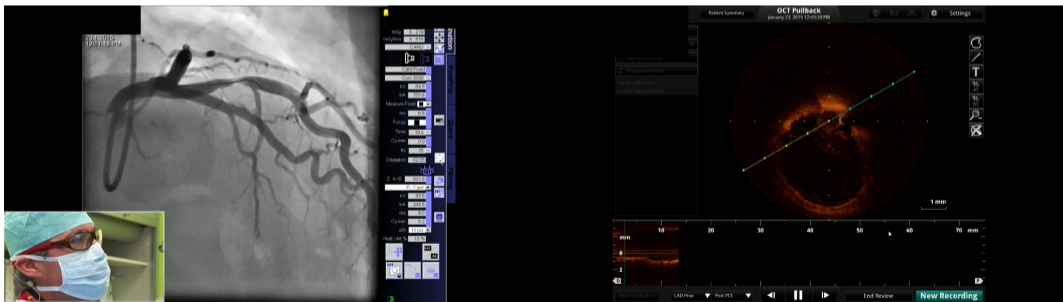
- ▶ Streaming videa z Youtube a videokonference přes Skype, Whatsapp, Facebook, Google Hangouts Meet, Zoom apod. všichni známe.
- ▶ Jaké nové aplikace založené na audiovizuálních přenosech současné sítě umožňují?
  - ▶ Hraní her prostřednictvím audiovizuálních přenosů včetně VR/AR.
  - ▶ Je možné operovat prostřednictvím videa na dálku?
  - ▶ Může produkční tým pracovat na nejnovějším hollywoodském filmu z obýváku?
  - ▶ Mohou spolu hrát hudebníci po síti?

# Hraní her prostřednictvím audiovizuálních přenosů

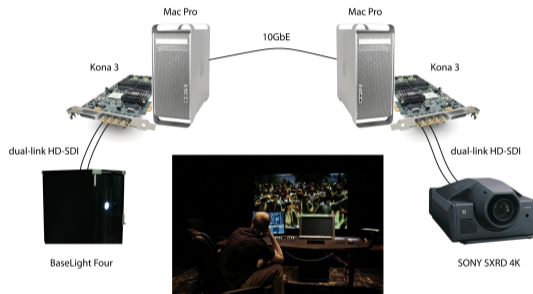
- ▶ Dnes relativně běžné technologie (Google Stadia, NVidia GeForceNOW)
- ▶ Obraz a zvuk her je generovaný v cloudu a přenášený v podstatě obdobně jako streaming z Youtube, zpět se přenáší události z klávesnice/herního ovladače
- ▶ Větší nároky na kvalitu a šířku pásma (cca 35 Mbps pro 4K video)
- ▶ Obvykle implementováno pomocí webových technologií, které nejsou nutně stavěné na co nejnižší zpoždění
- ▶ Jednosměrné zpoždění herní cloud → obrazovka uživatele je běžně cca 100 ms
  - ▶ K tomu je potřeba přičíst zpoždění ovládání a zpoždění sítě mezi herním zařízením a cloudovou platformou
- ▶ Při hraní na lokálním PC je zpoždění obvykle kolem 40 ms (60 fps video)
- ▶ I tak je to docela použitelné a dá se na hraní prostřednictvím přenosů obrazu a zvuku zvyknout

# Je možné operovat prostřednictvím videa?

- ▶ Operace, kdy operátor provádí zákrok na základě obrazů z kamer a dalších medicínských zobrazovacích metod jsou dnes zcela běžné
- ▶ Šetrnější pro pacienta, malé operační rány
- ▶ Endoskopické/laparoskopické zákroky, kardiologie, dentální chirurgie apod.



- ▶ Vzdálená postprodukce filmového materiálu (např. color grading) se přímo nabízí
  - ▶ HDR video s reprezentací barev až 16 bity na barevný kanál je nutná
  - ▶ Dnes velká rozlišení (4K–8K)
  - ▶ Datové toky v řádech stovek Mbps až jednotek Gbps



- ▶ Filmová studia by ráda tento proces nasadila tak, aby například koloristi mohli pracovat z domu (zejména v souvislosti s COVID-19)
  - ▶ Studio má typicky desítky koloristů
  - ▶ I při sofistikované kompresi je potřeba 100 Mbps na stream
  - ▶ Problém s domácími sítěmi

# Mohou spolu hrát hudebníci po síti?

- ▶ Rychlost zvuku ve vzduchu je  $343 \text{ ms}^{-1}$ , při šířce orchestřiště 10 m je zpoždění mezi hudebníky do cca 35 ms, na takové zpoždění jsou hudebníci cvičeni
- ▶ End-to-end zpoždění 35ms je pro tuto aplikaci svatý grál i při síťových přenosech
  - ▶ Ideálně pro zvuk i obraz
- ▶ Takové zpoždění je extrémně obtížně dosažitelné
  - ▶ End-to-end zpoždění při běžném zpracování zvuku na PC je v řádu malých desítek ms (bez síťového přenosu)
  - ▶ Pokud zpracováváme obraz po jednotlivých snímcích, pak je zpoždění přímo závislé na snímkové frekvenci a dosáhnout end-to-end zpoždění  $\leq 35 \text{ ms}$  je možné jen při velmi vysokých snímkových frekvencích
  - ▶ Síťové zpoždění při přenosech v rámci ČR může být 10 ms v jednom směru, mezi Evropou a Amerikou/Asií ale 100 a více ms