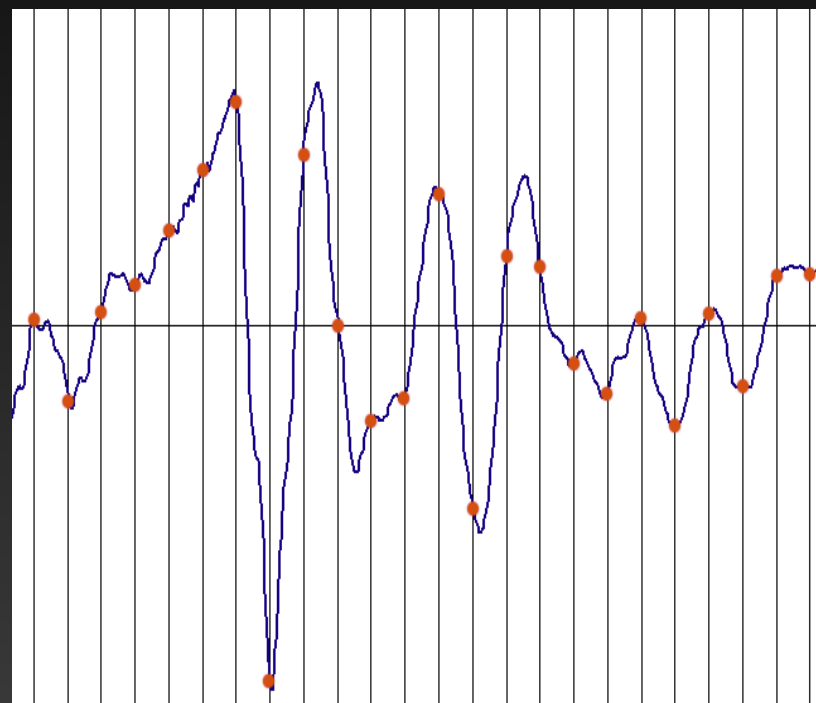


# Resampling - převzorkování

- Vzorkovat lze jen spojitý signál...
  - interpolace: Diracův impuls jako digitalizační filtr
  - filtrace: digitalizační filtr s nenulovou plochou (např. čtverec, konus)
- ... ale data jsou už navzorkovaná (rastr).
  - hledáme spojitou aproximaci diskrétního signálu
  - aproximace může být znovu navzorkována s jinou frekvencí

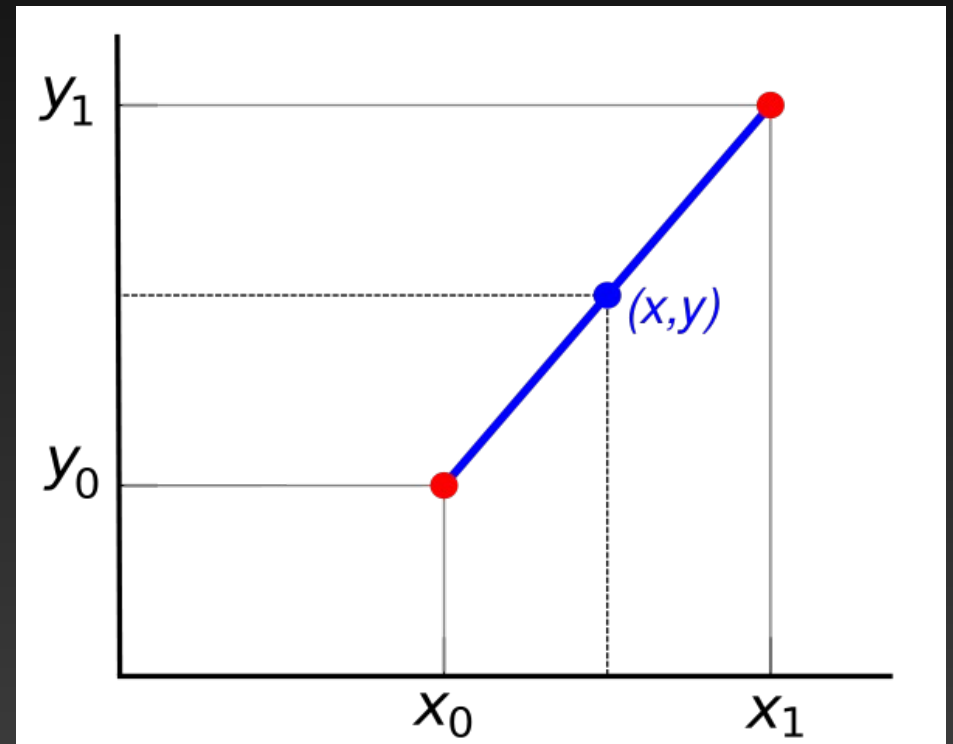
# Rekonstrukce

- Získání spojitého signálu z už navzorkovaného
  - Nearest neighbour
  - Linear (bilinear)
  - Cubic (bicubic)
  - Lanczos
  - Sinc (optimum, teor.)



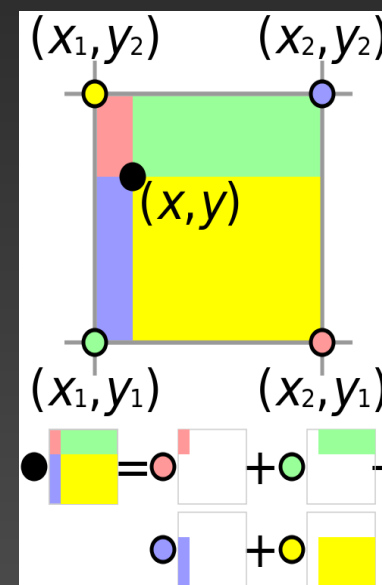
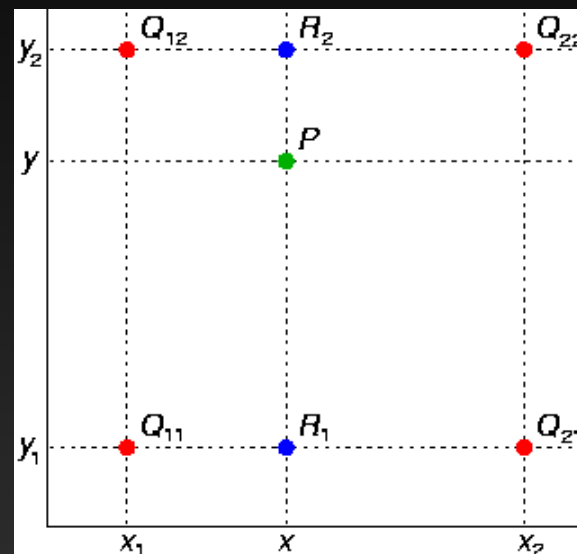
# Lineární interpolace

- Interpolovaná hodnota vybraná proporcionalně podle vzdálenosti od původních hodnot
  - Nejjednodušší
- (př. pro jednu dimenzi)



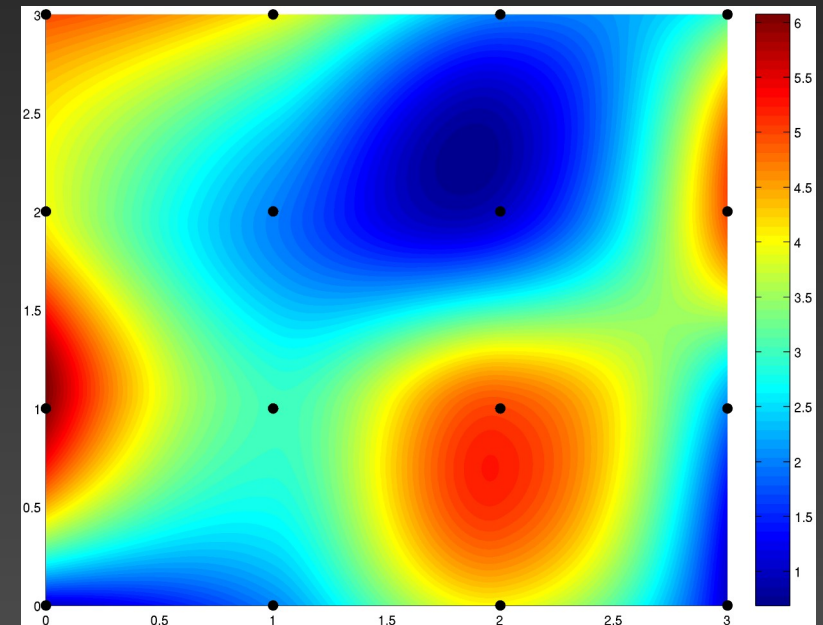
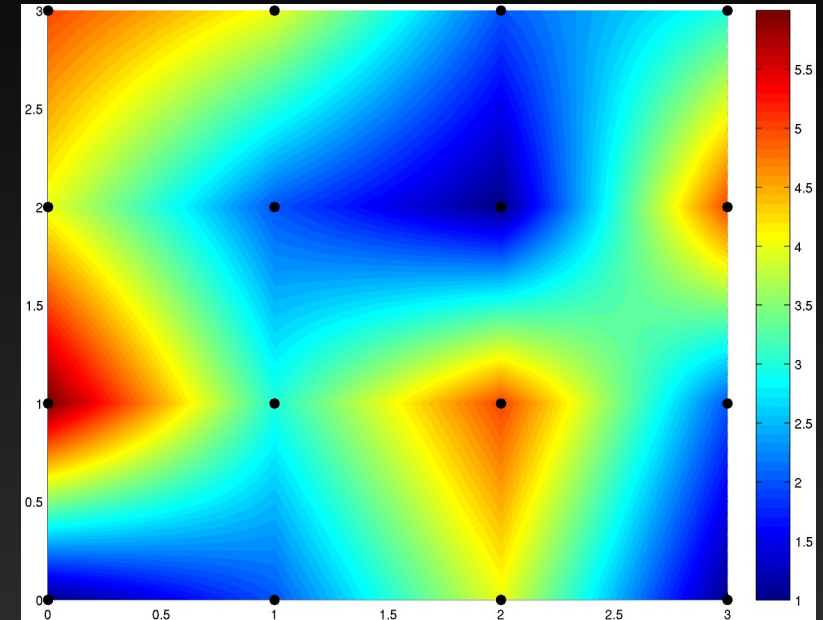
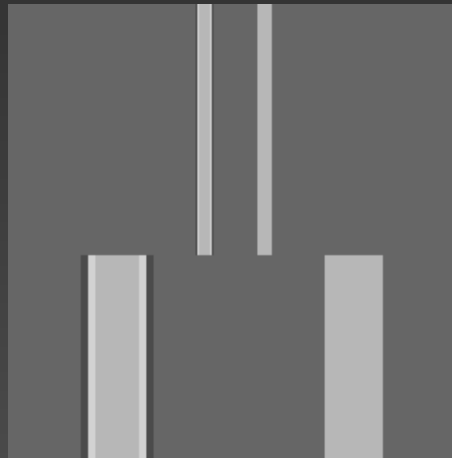
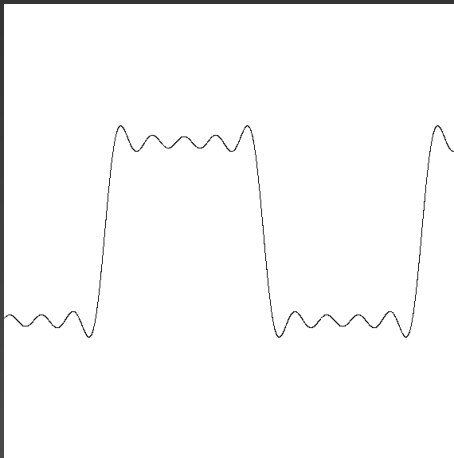
# Bilineární interpolace

- Lineární interpolace dvou proměnných
  - $X, Y \rightarrow 2D$
- 4 sousední vzorky
- 3 lineární interpolace
- Rozmazává hrany (text, čáry, ...)



# Bikubická interpolace

- Spline-based
- 16 sousedních vzorků samples
- výpočetně náročná
  - 16 rovnic o 16 neznámých
- Plynulejší
- Méně artefaktů
  - zachovává detaily
- Generuje překmity



# Změna rozlišení

- Rekonstrukce obrazu převzorkováním
  - nejbližší soused (nearest neighbour)
  - bilineární interpolace
  - kubická interpolace, atd.
- Nová vzorkovací frekvence může být
  - vyšší → zvětšení
  - nižší → zmenšení
- Interpolátory neustále využívány v GPU
  - specializovaný HW blok

# Dynamický rozsah

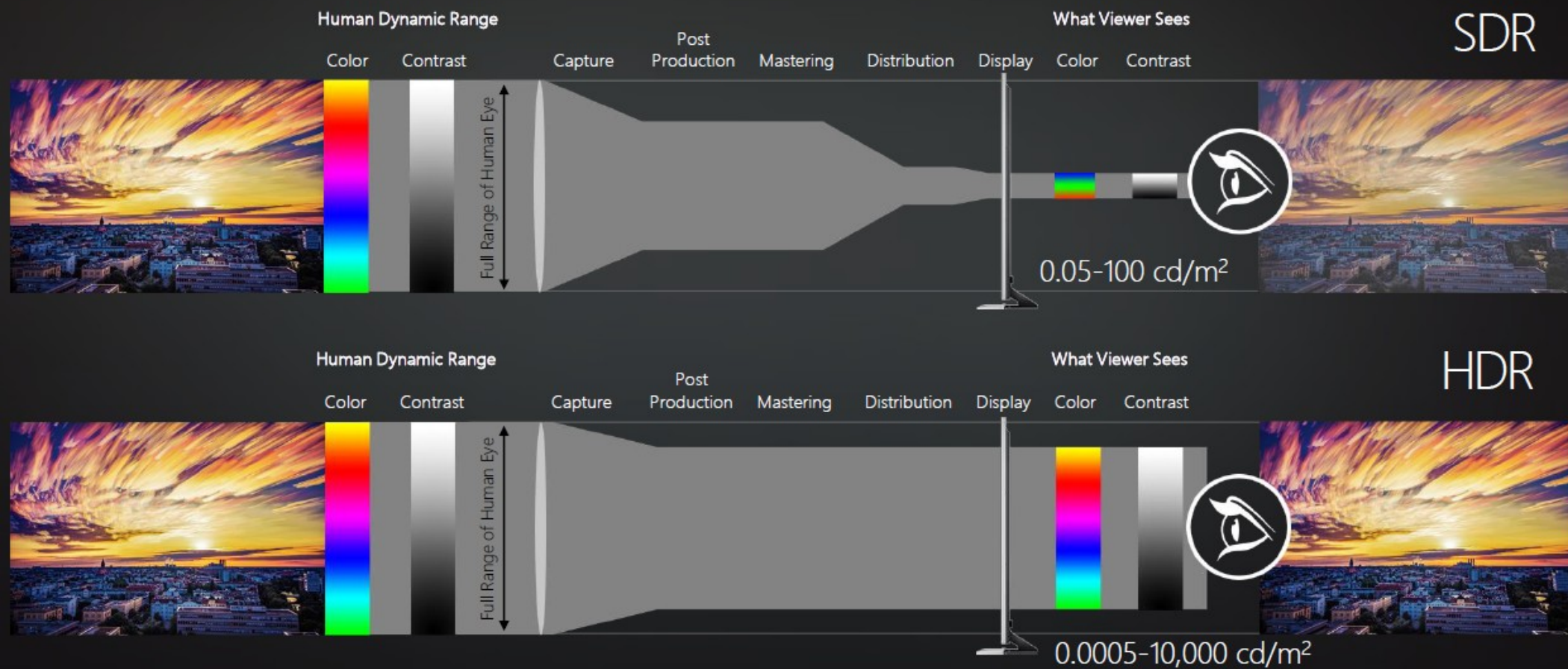
- Udává počet úrovní pro jeden barevný kanál
- Počet úrovní je určen počtem bitů, které slouží pro uložení barevného kanálu
  - 1 bit ... 2 úrovně (černá/bílá)
  - 6 bitů ... obyčejné LCD displeje (TN)
  - 8 bitů ... 256 úrovní
  - 10 bitů ... lepší LCD displeje (S-IPS), UHD-TV
  - 12 bitů ... 4096 úrovní (digitální zrcadlovky)
  - 16 bitů ... 65 536 úrovní (half-float, rendering)
  - 32 bitů ... přes 4 miliardy úrovní (float)

# Obrazy s vysokým dynamickým rozsahem – HDR

- Rozsah jasu 0 – 255 někdy nestačí
  - potřebujeme rozlišit velké množství jasových úrovní
  - LCD 1:1000, DSLR 1:4000(6000), negativ 1:5000(8000)
    - oko až 1:16000
  - nutné více bitů na pixel (36bpp, 48bpp ... )
- Velký kontrast ve scéně
  - tmavý kout a svíčka (žárovka, slunce ... )
- Problém **mapování tónů** do rozsahů zobrazitelných běžnými monitory
- Využití: osvětlování virtuální 3D scény, postprocessing filmové produkce
- OpenEXR, Radiance HDR, LogLuv TIFF ...



# Chasing the human vision system



# HDR



HDR rendering



Mapování tónů (globální vs. lokální)

např. gamma komprese  $V_{out} = A \cdot V_{in}^{\gamma}$



# Kompresa rastrových dat

- $\text{data [byte]} = \text{šířka} * \text{výška} * \text{bpp} / 8$ 
  - např. fullHD rozlišení:  $1920 * 1080 * 24 / 8 = \text{cca } 6 \text{ MB}$
  - nutné komprimovat
- **Bezeztrátová** komprese
  - RLE kódování (např. TGA, JFIF)
  - Statistické (Huffmanovo, aritmetické) (např. JFIF)
  - LZW (Lempel-Ziv-Welch) (např. GIF)
  - Deflate (např. PNG)
- **Ztrátová** komprese
  - využívá nedokonalosti lidského zrakového aparátu
  - vyšší citlivost na jas než na barevnou informaci
  - DCT (diskrétní kosinová transformace, JFIF)
  - Wavelety (vlnková transformace, JPEG2000)

# Běžné (zajímavé) 2D rastrové formáty



# PBM, PGM, PPM (PAM)

- Pro skriptování a pokusy
- Nejjednodušší formáty
- Bez komprese
- Textová i binární verze
- **PBM** – Portable Bitmap
  - 1 bpp (jen černobílé)
- **PGM** – Portable Greymap
  - 8bpp – 16bpp (256 – 65535 stupňů šedi)
- **PPM** – Portable Pixmap
  - 24bpp – 48bpp (TrueColor a víc)
- PAM – Portable Anymap
  - složitější hlavička

P1

# toto je komentář – B/W bitmapa

# rozměry 256 x 256

256 256

0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1 0 1 0

1 0 0 0 0 0 0 ...

P3

# PPM – barevný obrázek

# rozměry 625 x 576

# max. hodnota složky je 255

625 576

255

224 136 122 224 136 122 224 136 124

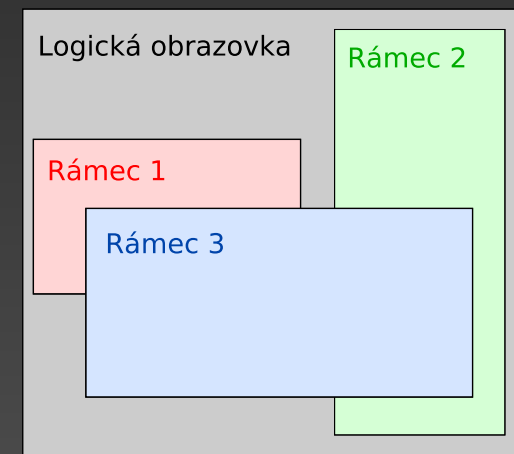
122 224 136 122...

# BMP

- Microsoft (Windows), IBM (OS/2)
- Zbytečně komplikovaný formát
  - několik verzí hlaviček, mění se význam položek...
- **Neúsporný** – zbytečné položky, komprese zvětšuje
- Možnosti
  - paleta (několik formátů)
  - 1bpp - 24bpp (16,7 miliónů barev)
  - průhlednost (některé verze, nedokumentováno)
- Volitelně jednoduchá komprese
  - příliš se nepoužívá, omezení na vybrané režimy
  - algoritmus RLE – špatně navržený
    - může obrázek i výrazně zvětšit!
  - Huffman – snad nikde ...
- **Běžné použití: nekomprimovaný 24bpp RGB**

# GIF

- Graphics Interchange Format
- Účinná **neztrátová komprese** LZW
- Barevná **paleta** s 256 barvami na jeden rámeček
- Možnost **více rámečků** v jednom obrázku
  - Samostatná paleta pro každý rámeček
    - možnost simulace TrueColor
  - Umožňuje vytvářet jednoduché **animace**
    - postupné zobrazování rámečků se zpožděním
- **Prokládání** – přenesen každý 8, 4, 2, 1 řádek
- Jednobitový **alpha** kanál
  - jedna barva v paletě může být plně průhledná
- LZW do roku 2004 patentováno → vymáhání licenčních poplatků → vývoj PNG
- **Běžné použití: krátké, jednoduché animace**



# PNG

- Portable Network Graphics, náhrada za GIF
- Jednoduchý, velmi dobře dokumentovaný formát
- Účinná (ale pomalejší) **neztrátová komprese** (deflate)
- **Prokládání** Adam7
- Barvové profily **ICC**, **gamma** korekce
- Možnosti
  - Barevná paleta (256 položek z RGBA)
  - 8bpp – 16bpp stupně šedi
  - 24bpp – 48bpp RGB, RGB + **Alpha** kanál
- **Běžné použití: statické obrázky ve vysoké kvalitě**
- Ale:
  - **neumí** animace (MNG, APNG)
  - pro velmi jednoduché obrázky je větší (1x1 B/W)



# TIFF

- Tagged Image File Format
- **Komplikovaný** a 'funkčně' bohatý formát (kontejner)
  - Neztrátová i ztrátová komprese – RLE, LZW, JPEG...
  - Speciální komprese pro čb. obrázky – faxy (CCITT)
  - Nejrůznější bitové hloubky (1bpp – 64bpp)
  - RGB, **CMYK**, **YCbCr**, **CIE L\*a\*b**, **RAW**
  - Vysoký dynamický rozsah (**HDR**)
  - **Vícestránkové** dokumenty
    - Ukládání naskenovaných materiálů, faxů
- Problémy s kompatibilitou
  - příliš složitý
  - několik verzí formátu (aktuálně v.6)
  - každý program implementuje jen něco
- **Běžné použití: pokud potřebujeme speciality**

# JFIF/JPEG

- Formát souboru = JFIF
  - JPEG File Interchange Format
  - Data obrázku + dodatečná data (metadata)
    - EXIF data (expoziční časy, datum, GPS...)
    - náhledový obrázek
    - audio stopa, textový popis ...
- Formát komprese = JPEG
  - Joint Picture Experts Group
  - JPEG může být i v TIFF
- Vhodné pro ukládání fotografií, skenů
- Nevhodné pro ukládání grafů, textu, screenshotů
  - rozmazání + artefakty na ostrých hranách, liniích

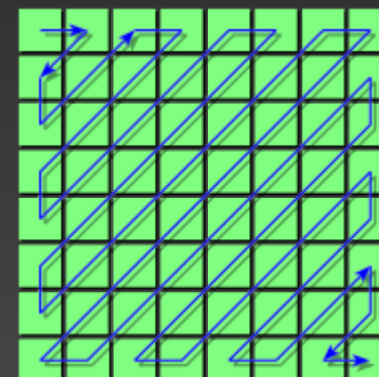
# JPEG

- Nejedná se o souborový formát ale o metodu **ztrátové** komprese
- Možnosti
  - 8bpp, 12bpp – stupně šedi
  - 24bpp, 36bpp – RGB
  - umožňuje i neztrátovou kompresi – není běžné
  - hierarchická komprese – víc rozlišení (mapy)
  - možnost prokládání
- Komprese v blocích 8x8 pixelů
- **Běžně: ztrátové 24bpp RGB bez hierarchie, často prokládání**



# Kroky komprese JPEG

- 1) Převod do barevného prostoru YCbCr
  - jas + dva barvové kanály
- 2) Převzorkování barev (4:4:4, 4:2:2, 4:2:0)
  - **ztráta informace**: 75% barvy
- 3) Rozdělení obrázku do makrobloků 8x8
- 4) DCT: Diskrétní kosinová transformace
  - převod do frekvenční oblasti
- 5) Kvantizace pomocí kvantizační tabulky
  - **ztráta informace**:  
vysoké frekvence = detaily
- 6) Linearizace (zig-zag)
- 7) RLE kódování (**neztrátové**)
- 8) Huffmanovo nebo aritmetické kódování (**neztrátové**)



Originální obraz



Dekompozice

$$\begin{aligned} Y &= 0,59R + 0,30G + 0,11B \\ Cr &= -0,17R - 0,33G + 0,5B \\ Cb &= +0,5R - 0,42G - 0,08B \end{aligned}$$

Makroblok



každý pixel – 128

107	103	98	84	45	8	4	1
109	93	114	97	90	66	11	-3
118	90	110	96	93	89	72	33
106	97	99	100	96	97	92	79
98	108	106	97	93	106	94	94
103	103	98	105	98	106	87	109
99	107	109	98	103	93	98	97
84	80	89	103	100	105	92	81

DCT

DC	AC <sub>01</sub>	AC <sub>07</sub>					
705	109	-41	8	-1	7	14	1
-94	118	-23	1	3	9	6	9
-70	52	-17	-11	10	1	-4	3
-12	16	24	-29	7	-16	-5	-10
-18	-11	13	-11	-1	4	-11	-7
3	-2	23	-12	2	6	-14	7
-5	-8	10	3	4	9	-1	-5
2	5	7	-3	-12	4	-4	8
AC <sub>70</sub>							

Kvantizační tabulka

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	66	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Koeficienty kvantizační tabulky, zde pro kvalitu 50 %, mohou být násobeny konstantou  $\alpha$ , která určuje míru ztrátovosti.

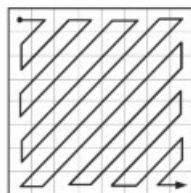
Kvalita Q:  
 $1 \leq Q \leq 50, \alpha = 50 / Q,$   
 $51 \leq Q \leq 100, \alpha = 2 - 100 / Q$

Kvantizace

$$\begin{aligned} 705 / 16 &= 44, \\ 109 / 11 &= 10, \dots \end{aligned}$$

44	10	-4	1	0	0	0	0
-8	10	-2	0	0	0	0	0
-5	4	-1	0	0	0	0	0
-1	1	1	-1	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Linearizace



44, 10, -8, -5, 10, -4, 1, -2, 4, -1, -1, 1, -1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, -1, 0, 0, ...

RLE kódování nul  
( nul / kateg. / hodnota )

44; (0/4/10), (0/4/-8), (0/3/-5), (0/4/10), (0/3/-4), (0/1/1), (0/2/-2), (0/3/4), (0/1/-1), (0/1/-1), (0/1/1), (0/1/-1), (5/1/1), (5/1/-1), **EOB**

Huffmanovo kódování

(pouze AC členy)

1011110010110111100010101110010001100101011001000000000100011110101111101001010

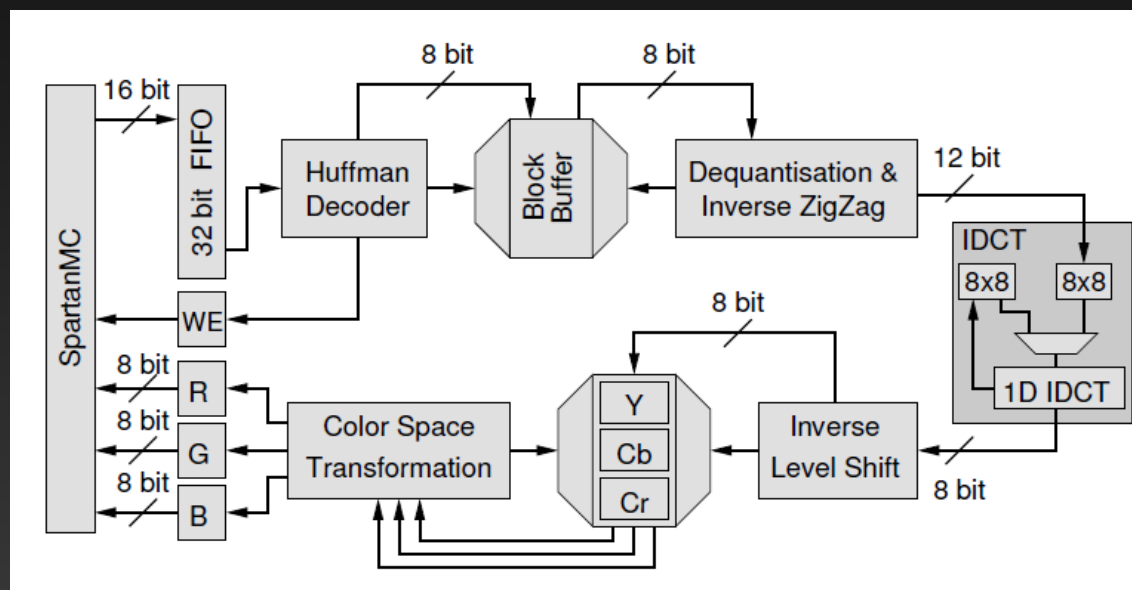
bitový tok



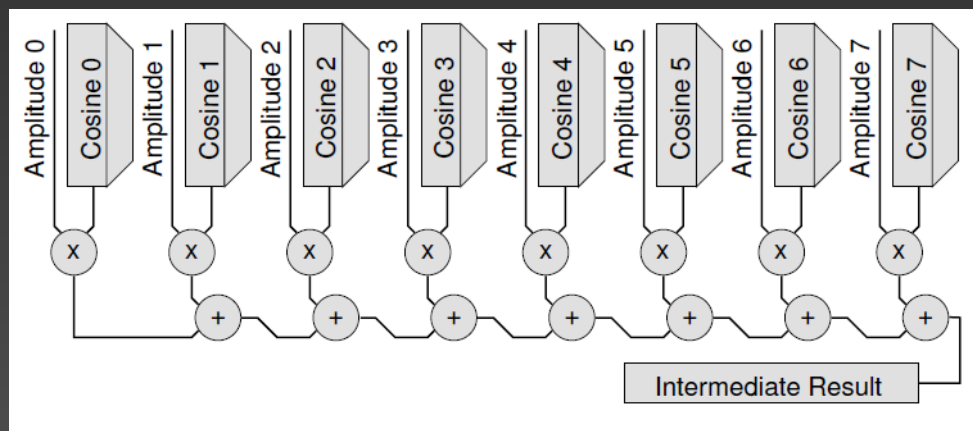
# Hardware JPEG dekodéru

- JPEG algoritmus navržen s ohledem na HW realizaci (jednoduché kroky, pipeline, ...)

# Celý dekodér (příklad)



# 1D DCT



# Nástupci JPEG

- Odvozené z videa
  - HW podpora v mobilních zařízeních (rychlost, spotřeba)
  - Větší schopnosti, vyšší kvalita → pravděpodobně budoucnost
- HEIF
  - Apple, Google, Microsoft; patentovaný
  - odvozený z 1 snímku z video formátu HEVC (h265)
  - 4:4:4 (plné barvy), HDR, dlaždice, 10bit, alpha
  - obrázky i sekvence, náhledy, metadata
- WebP
  - Google, otevřený formát, hlavně web
  - odvozený z 1 snímku z video formátu WebM (VP8)
  - ztrátový i neztrátový, RGBA8
  - makrobloky, YCbCr, DCT, Huffman, metadata



# Pokrok mezi video kodeky (a tedy i statickými snímky)

- H.264 Vs H.265

## H.264/AVC

16x16 Macroblock



### Block coding structure

3 Intra partitioning



4 Inter partitioning



+4 sub-partitioning 8x8



2 Transform sizes:  
4x4, 8x8



Up to 9 Intra prediction directions

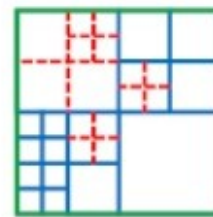


## HEVC



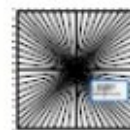
Coding Unit  
64x64 to 8x8

### Quadtree coding structure



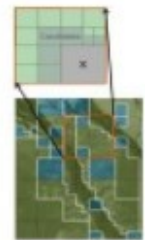
Prediction Unit and Transform Unit partitioning

⇒ Multiples sizes/forms: 64x64 to 4x4



35 Intra prediction directions

Efficient spatio-temporal mv prediction





# RAW data

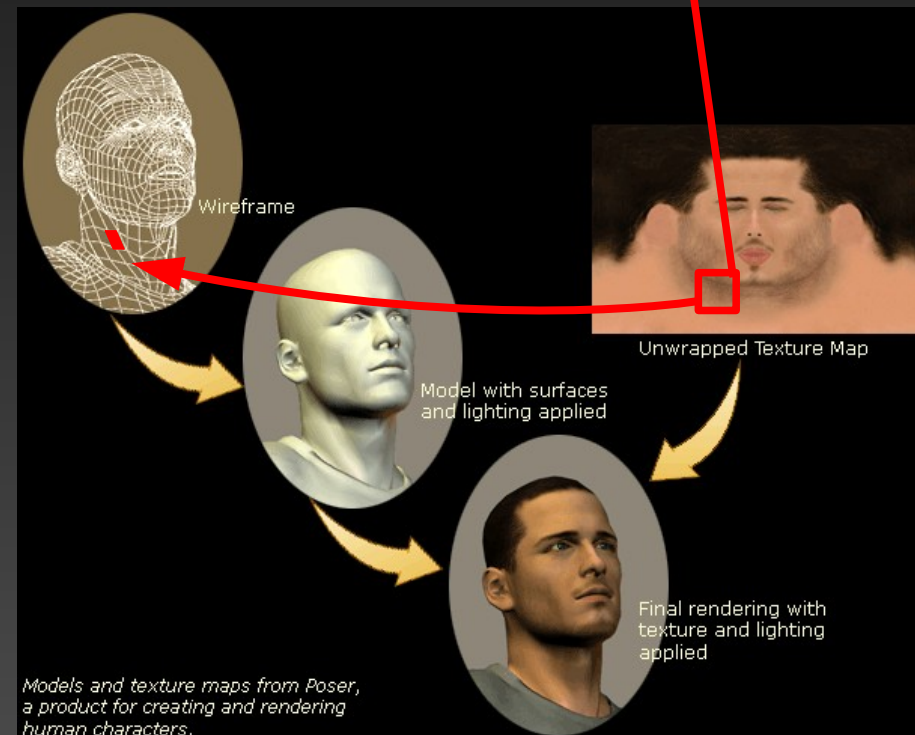
- Data načtená přímo z optického elementu fotoaparátu (kamery)
- Obsahuje **maximum obrazové informace**
  - typicky 12 nebo 14bpp (16bpp)
- Umožňuje provádět korekce až v počítači
  - vyvážení bílé, komprese, odšumění, ...
- Formát závislý na výrobci – existují **min. desítky**
  - bitová hloubka, barevný prostor, rozložení pixelů ...
  - pokusy o standardizaci
    - TIFF-EP – ISO Electronic Photography
    - DNG – Adobe, Digital Negative
    - NEF – Nikon, založený na TIFF
    - CRx – Canon, založený na TIFF
- **Nutné dále zpracovat – digitálně „vyvolat“**
  - software (Darktable, Lightroom...), výkonný počítač, disk
- Velký objem dat
  - nepoužívá se (většinou) ani neztrátová komprese

# Speciální rastrové formáty – textury

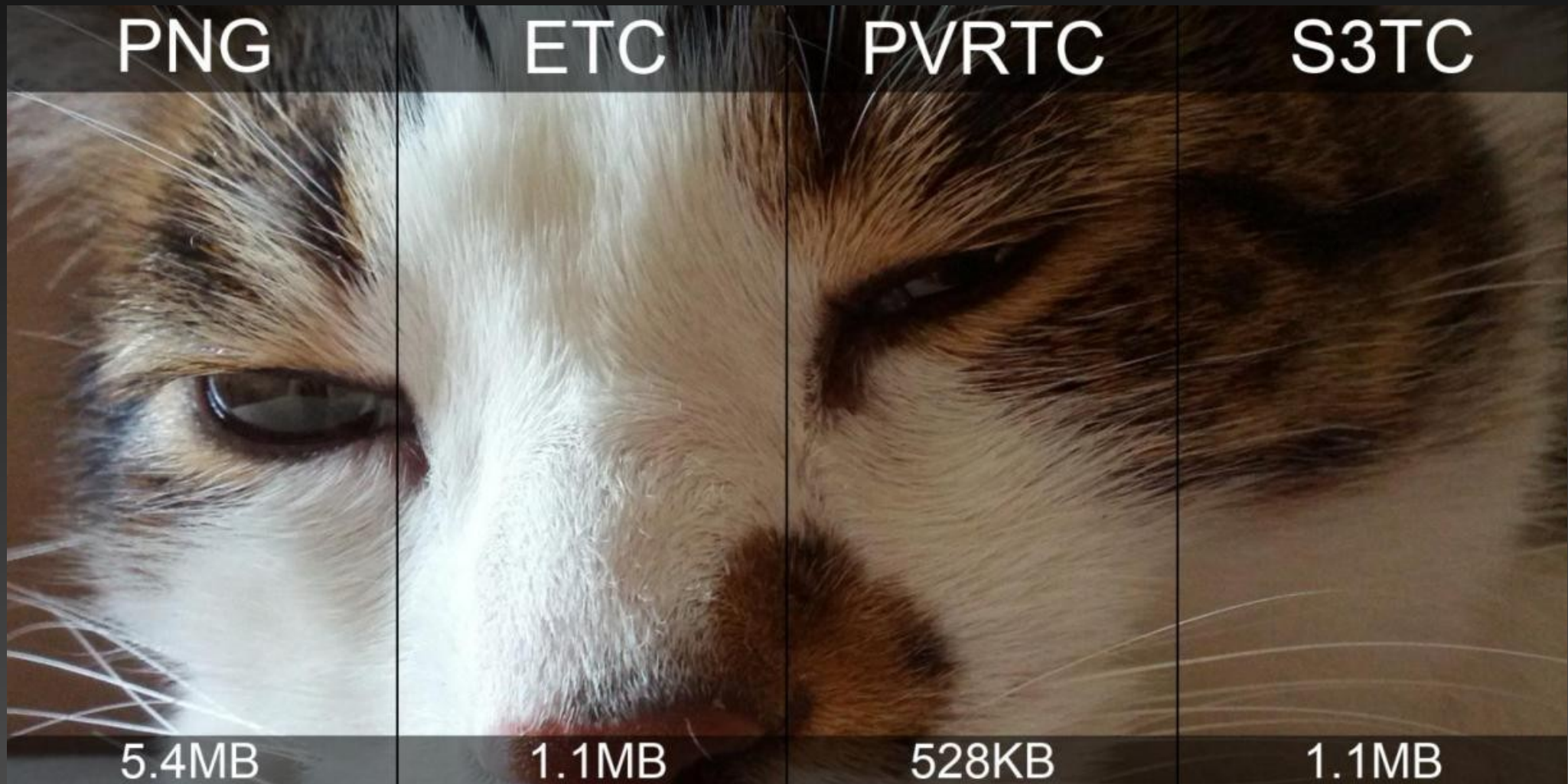
Potřebujeme náhodný, malý kousek velkého obrázku (textury). Nemáme dost paměti ani času na kompletní dekompresi → PNG, JPEG atd. nelze použít!

- Potřebujeme
  - náhodný přístup, rychlost, dobrý kompresní poměr, kvalitu
- Obvykle dlaždicový ztrátový formát. Každá malá dlaždice nezávisle komprimovaná s podporou HW → takřka náhodný přístup, rychlé, efektivní.
- S3TC, DDS, DXT, PWR, ETC, PVRTC, ASTC, BC, ATC, ...

Potřebujeme  
co nejrychleji  
jen tuto malou oblast.



- ETC – Ericsson Texture Compression (Android)
- PVRTC – PowerVR Texture Compression (Intel, Kindle, ...)
- ATITC – ATI Texture Compression (Qualcom, Nexus)
- a další (3DC, Arm ASTC, ...)





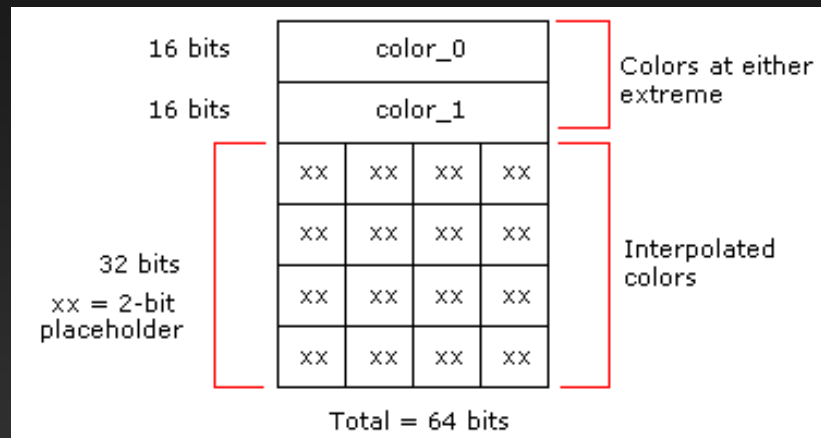
# Srovnání kvality

- Nekomprimovaný obraz vs. S3TC stejné velikosti v byte.
- Nekomprimovaný je nutné uložit ve velmi malém rozlišení a při vykreslování zvětšit.



# S3TC

- Nejběžnější, jednoduchý (ale překonaný); 5 variant (DXT1-5)
- Bloky 4x4 pixelů R8G8B8 komprimuje na 64 bitů
  - nekomprimovaně 4x4x24 = 384 bitů → komprese 1:6



color\_0, color\_1 = R5G6B5 (vybraná libovolně)

- if  $C_0 > C_1$  then  $C_2 = \frac{2}{3} * C_0 + \frac{1}{3} * C_1$   
 $C_3 = \frac{1}{3} * C_0 + \frac{2}{3} * C_1$   
else  
 $C_2 = \frac{1}{2} * C_0 + \frac{1}{2} * C_1$   
 $C_3 = \text{black}$