

Multimediální systémy

Jan Outrata



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLMOUCI

přednášky



Komprese obrazu



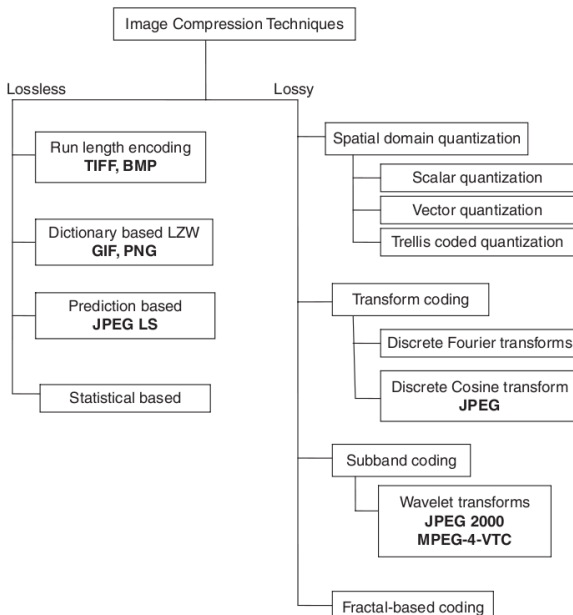
- obrazová média (obraz, video, grafika po rasterizaci) všudypřítomná, nárůstající na rozlišení a objemu → **potřeba komprese**
- např. 4288×3216 24bpp (foto) ~ 40 MB \rightsquigarrow 4 MB (JPEG)
- např. formáty (ISO) JPEG pro foto a video, GIF/PNG pro obrázky na internetu
- software pro práci s komprimovaným obrazem
- bezetrátové (GIF, PNG) i ztrátové, typicky hybridní kompresní metody (JPEG)



- **irelevance**: vizuální (parametry obrazu jsou **nad možnosti zobrazení nebo vnímání** za daných podmínek), aplikační (některé části obrazu nepotřebné)
- ztrátová komprese – neznatelné nebo přijatelné zkreslení, redukce entropie
- **redundance**: statistická, **barvy (sousedních) pixelů jsou (lokálně) korelované** – objekty, oblasti = „podobné“ pixely, postupné změny
 - **prostorová** = „podobné“ pixely v lokálních oblastech
 - **spektrální** = málo dominantních frekvencí
- bezztrátová komprese

- bezztrátové: využití prostorových redundancí
- ztrátové/hybridní: kvantizace v prostorové nebo frekvenční doméně (po transformaci) a bezztrátové statistické kódování

Obrázek: Taxonomie metod komprese obrazu



- omezená úroveň komprese (entropie)
- využití pro **zachování** (maximální, původní) **kvality**, např. u specifických fotografií (lékařských, astronomických apod., ztráty jsou nepříjemné) nebo ve filmovém průmyslu
- převod 2D pole pixelů na posloupnost hodnot pomocí (progresivního) **skenování**
- RLE: formáty např. **BMP**, **TIFF** (rozšiřitelný se specifikací komprese), RLA, PICT
- slovníkové (LZW): např. **GIF**, **PNG**, komprese **indexovaných barev** místo barev pixelů (po předchozím výběru optimální palety barev dané velikosti = vektorové kvantizaci, např. 8b)
- založené na predikci (DPCM): predikce barev sousedních pixelů, statistická komprese rozdílů, např. **JPEG LS** (lossless mode) – predikce ze sousedního levého, horního nebo/a levého horního pixelu → chybový obraz s menší entropií

Obrázek: Predikce v JPEG LS (lossless mode)

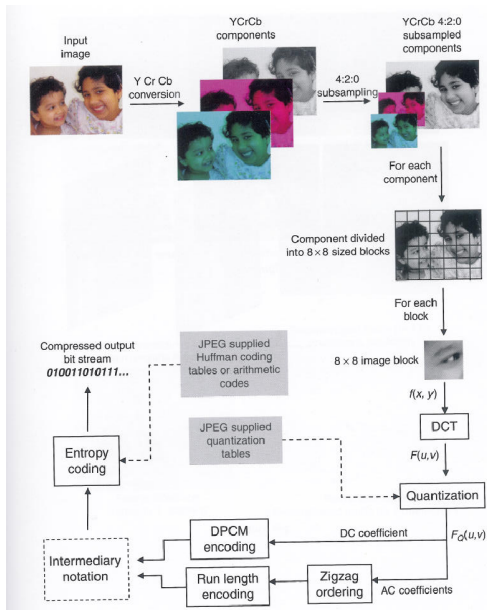
	C	B		
	A	X		

Prediction index	Prediction
0	No prediction
1	A
2	B
3	C
4	$A + B - C$
5	$A + ((B - C)/2)$
6	$B + ((A - C)/2)$
7	$(A + B)/2$

- ISO standard, 1993
- široce rozšířený, volné implementace a použití
- použití **diskrétní kosinové transformace (DCT)** – popis ve frekvenční doméně, hardwarová implementace

Obrázek: JPEG komprese (baseline mode)

- 1 transformace barev do **barevného prostoru YCbCr** (podobný YUV)
- 2 **podvzorkování** ve schématu 4:2:0
- 3 každý barevný kanál zpracováván samostatně
- 4 **rozklad** obrazu do **bloků 8×8 pixelů** (velikost zvolena experimentálně jako optimální pro využití prostorových a frekvenčních korelací) – příp. rozšíření obrazu o nulové okraje
- 5 každý blok zpracován samostatně, zpracování bloků v pořadí daném (progresivním) skenováním – v **baseline mode**



- 6 **DCT**: výpočet frekvenčních koeficientů $F(u, v)$ ze vzorků $f(x, y)$, první, největší, koeficient na pozici $\langle 0, 0 \rangle$ (nejnižší frekvence) = **DC koeficient**, ostatní = **AC koeficienty**

$$f(x, y) = \frac{1}{4} \left[\sum_{u=0}^{u=7} \sum_{v=0}^{v=7} C(u)C(v) \times F(u, v) \times \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \right]$$

$$F(u, v) = \frac{1}{4} C(u)C(v) \left[\sum_{x=0}^{x=7} \sum_{y=0}^{y=7} f(x, y) \times \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \right]$$

$$C(u), C(v) = \frac{1}{\sqrt{2}} \text{ pro } u, v = 0, \text{ jinak } 1$$

- 7 **kvantizace** koeficientů $F(u, v)$ na $F_Q(u, v)$: s využitím **kvantizační tabulky** Q (vyplněné experimentálně, kdy **nízké frekvence** jsou v obrazech **dominantní** a vnímání ztráty na nich je citlivější \rightarrow větší rozsah) – **kvantizační faktory** $Q(u, v)$ udávají kvantizační rozsah (menší hodnota = větší rozsah), **nastavení nižší „kvality“** komprese u software **zvětšuje faktory**, skoro všechny pro vyšší frekvence vysoké \rightarrow ztráta původních hodnot koeficientů po dekvantizaci

$$F_Q(u, v) = \left\lceil \frac{F(u, v)}{Q(u, v)} \right\rceil$$

Obrázek: DCT a kvantizace bloku

- 8 DC koeficient první, cikcak seřazení AC koeficientů – ke konci řady řetězce 0 = nižší entropie

Obrázek: Cikcak seřazení AC koeficientů

178	187	183	175	178	177	150	183
191	174	171	182	176	171	170	188
199	153	128	177	171	167	173	183
195	178	158	167	167	165	166	177
190	186	158	155	159	164	158	178
194	184	137	148	157	158	150	173
200	194	148	151	161	155	148	167
200	195	172	159	159	152	156	154

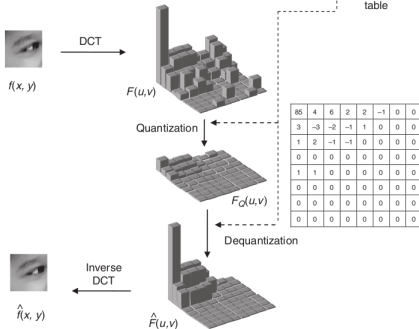
Pixel values $f(x, y)$

1359	46	61	26	38	-21	-5	-18
31	-35	-25	-11	13	10	12	-3
13	20	-17	-14	-11	-7	6	5
-5	5	2	-8	-11	-26	8	-4
10	15	-10	-16	-21	-7	8	7
-6	1	0	7	5	-7	-1	-3
-13	-8	1	10	8	4	-3	-4
-5	-5	-2	5	5	0	0	-3

DCT values $F(u, v)$

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	90	62
18	22	37	56	68	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

Quantization table



85	4	6	2	2	-1	0	0
3	-3	-2	-1	1	0	0	0
1	2	-1	-1	0	0	0	0
0	0	0	0	0	0	0	0
1	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

192	185	178	152	193	162	155	190
181	172	162	154	187	164	159	192
183	168	150	158	181	167	162	190
198	177	150	155	177	169	161	182
202	180	148	148	171	167	159	175
193	176	145	141	162	162	156	170
195	184	155	145	159	156	150	164
209	200	170	154	160	153	144	156

1360	44	60	32	48	-40	0	0
36	-36	-28	-19	26	0	0	0
14	26	-16	-24	0	0	0	0
0	0	0	0	0	0	0	0
18	22	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

- 9 **DPCM** DC koeficientů bloků – dočasná reprezentace jako dvojice velikost (v bitech) kódu rozdílu a rozdíl koeficientů pro sousední bloky
- 10 (varianta) **RLE** AC koeficientů bloku – kódovány pouze nenulové koeficienty, dočasná reprezentace jako dvojice dvojice počet předchozích nulových koeficientů v řadě a velikost (v bitech) kódu hodnoty koeficientu, a tato hodnota

Obrázek: Dočasná a kódová reprezentace koeficientů a bloku

- 11 kódování dvojic bloku: prvního prvku dvojice **Huffmanovým kódováním** (se statickým statisticky zjištěným modelem), druhého **celočíselným kódováním** s proměnlivou velikostí kódů – neprefixové, kódy kratší než u prefixového, jednoznačně dekódovatelné se znalostí velikosti kódů

Obrázek: Ilustrace JPEG komprese

DC coefficient representation:

symbol -1 symbol -2
 <SIZE> <AMPLITUDE>

AC coefficient representation:

symbol -1 symbol -2
 <RUNLENGTH, SIZE> <AMPLITUDE>

Intermediary stream

<2><3> <0,3><4> <0,2><3> <0,1><1> <0,2><-3> <0,3><6>
 <0,2><2> <0,2><-2> <0,2><2> <1,1><1> <1,1><-1> <0,1><-1>
 <0,2><2> <0,1><-1> <0,1><1> <0,1><-1> <1,1><1> EOB

Intermediary symbol	Binary representation of first symbol (prefixed Huffman Codes)	Binary representation of second symbol (non-prefixed variable integer codes)
<2> <3>	011	11
<0,3> <4>	100	100
<0,2> <3>	01	11
<0,1> <1>	00	1
<0,2> <-3>	01	00
<0,3> <6>	100	110
<0,2> <2>	01	10
<0,2> <-2>	01	01
<0,2> <2>	01	10
<1,1> <1>	11	1
<1,1> <-1>	11	0
<0,1> <-1>	00	0
<0,2> <2>	01	10
<0,1> <-1>	00	0
<0,1> <1>	00	1
<0,1> <-1>	00	0
<1,1> <1>	11	1
EOB	1010	

Binary Stream:

0111110010001110010100100110011001010110111100000110000010001111010



Original—24 bits per pixel



Compressed—2 bits per pixel



Compressed—0.5 bits per pixel



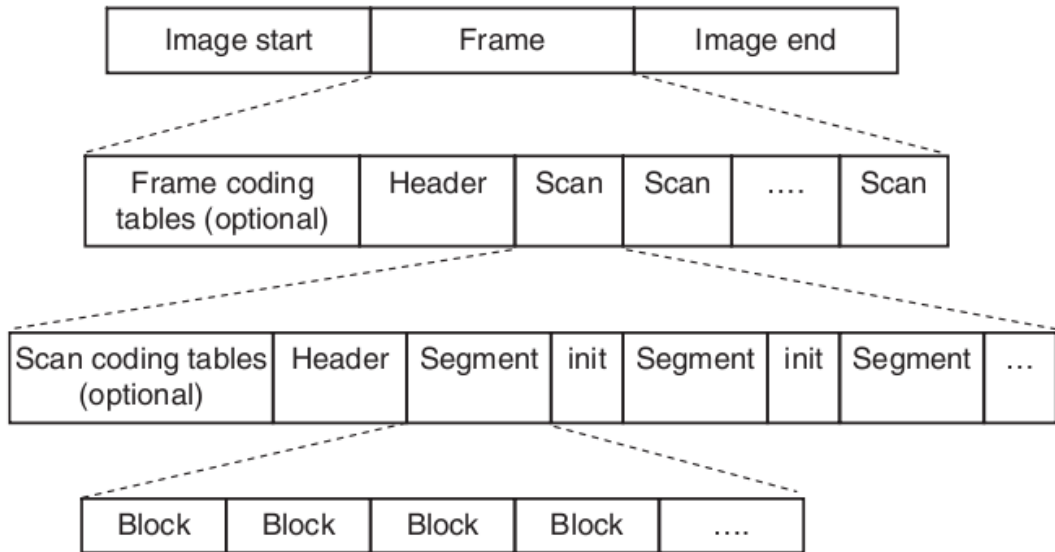
Compressed—0.15 bits per pixel

12 složení kódů bloků do bitového proudu – podle JPEG standardu

Obrázek: JPEG bitový proud

Nevýhody

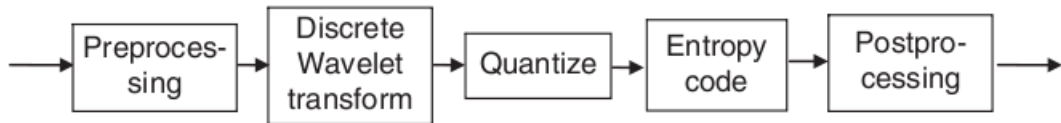
- pro narůstající rozlišení obrazu a potřebu úprav komprimovaného obrazu je dnes potřeba lepších metod
- vysoké **zkreslení na nízkých bit rate** (velmi nízké na středních a vyšších), **nemožnost náhodného přístupu** k částem obrazu (závislé kódování bloků), komprese max. 64kB bloků dat obrazu, kolem 40 různých **módů** (většinou aplikačně specifických, neimplementovaných dekodery), neodolnost proti chybám přenosu, neoptimální pro počítačově generované obrazy (vytvořený pro foto)



- ISO standard
- pro různé typy obrazů a různé modely přenosu a zobrazování
- použití **diskrétní waveletové transformace (DWT)** – lepší popis ve frekvenční doméně než DCT

Obrázek: Obr. JPEG 2000 komprese

- 1** (volitelně) **dlaždicování**: rozklad obrazu na čtvercové stejně velké bloky, s příp. rozšířením obrazu o nulové okraje, bloky zpracováváné samostatně
- 2** transformace barev do **barevného prostoru YCbCr** (podobný YUV), každý barevný kanál zpracováván samostatně
- 3** **posunutí úrovní**: odečtení poloviny rozsahu hodnot od hodnot pixelů pro rozsah s 0 uprostřed, kvůli DWT



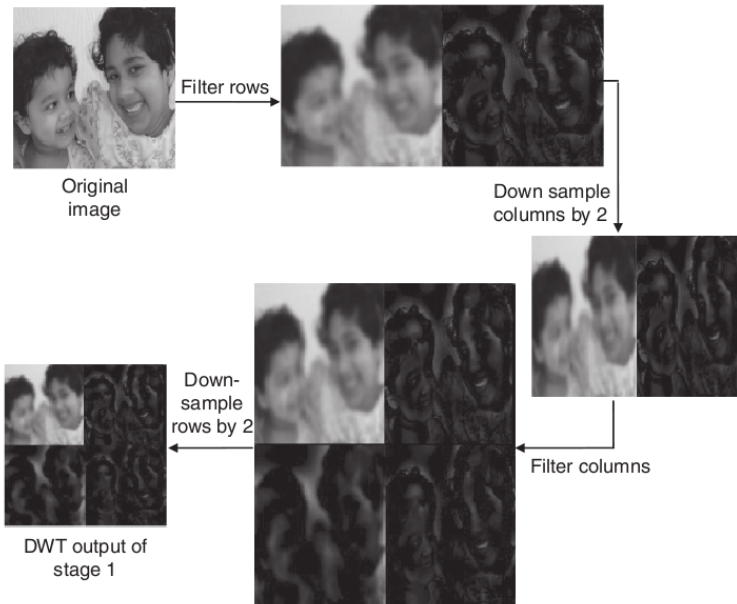
- 4 DWT: jednorozměrná**, krok rekuze nejdříve po řádcích, pak po sloupcích → 4 **kvadranty pásem LL, LH, HL a HH**, rekurzivní opakování na kvadrant LL, až 32-krát (v praxi 4–8-krát)

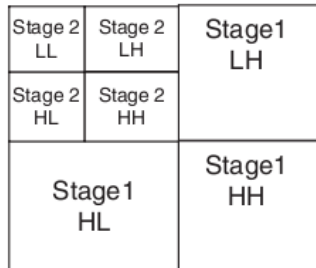
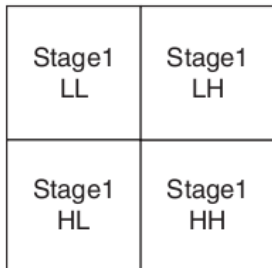
Obrázek: Obr. DWT

- 5 kvantizace** frekvenčních koeficientů v kvadrantech a **přeskládání** hodnot do paketů s hodnotami pro dané rozlišení od nejnižšího
- 6** pakety hierarchicky uspořádány do bloků a bloky komprimovány **statistickým kódováním**

Výhody (oproti JPEG)

- DWT poskytuje **lepší kompresní poměr** než DCT
- komprese a dekomprese (vybraných částí) obrazu v **různých kvalitách** (hierarchická organizace, region of interest, ROI, coding), obrazové **operace** (geometrické transformace, filtrace) na komprimované reprezentaci

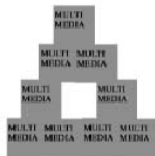
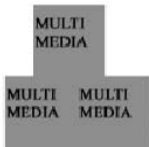
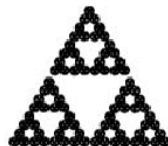
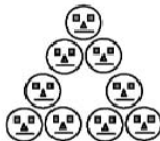
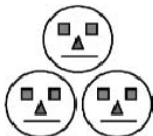




- = vyhledávání **podobných částí obrazu a transformací** (predikcí) vzorových na podobné a kódování vzorových částí a transformací
- vyhledávání výpočetně náročné
- použití **fraktálů** = sobě podobné geometrické objekty, kde části vypadají stejně jako celek – vytvořen a plně určen **rekurzivní afinní transformací** libovolného vzoru (tzv. seed), např. Sierpinskyho transformace (trojúhelník)

Obrázek: Obr. Fraktál definovaný Sierpinskyho transformací

- ideálně nalezení transformace, kde obraz = fraktál – „inverzní transformace“, v praxi (pro běžné obrazy) kombinace **více transformací pro části obrazu**



- = **rozklad** obrazu na čtvercové stejně velké bloky a pro každý blok **vyhledávání** vzorového bloku (v obrazu) a jeho **transformace** na blok s minimálním zkreslením

Obrázek: Obr. Fraktálová komprese

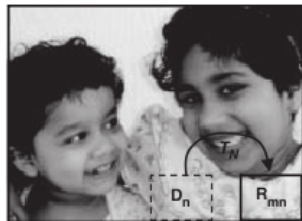
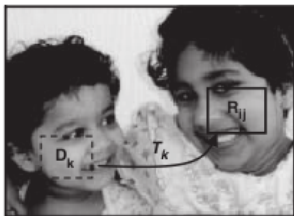
- experimentální stav (vyhledávání transformací), formát **FIF**), výsledky srovnatelné s DCT a DWT až mírně lepší



Construct
range blocks



For each range block R_{ij} , compute the domain block D_k and the affine transform T_k

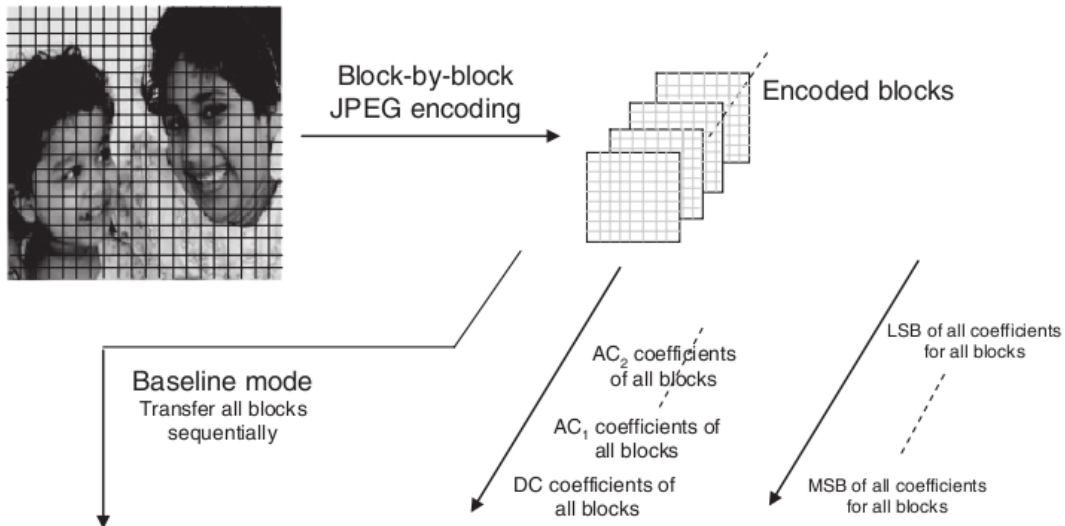


- problém: uložení a zobrazení (dekomprese) obrazu v plné kvalitě sekvenčně po blocích
 - pro celý obraz potřeba všechny
- **uložení a zobrazení obrazu postupně po úrovních kvality od nejhorší (nejhrubší obraz) po plnou = progresivní**
- JPEG: **progresivní módy** s více skeny bloků, např.
 - **spektrální selekce** – v 1. skenu jen DC koeficienty všech bloků, ve 2. skenu první AC koeficienty atd.
 - **postupná bitová aproximace** – v 1. skenu nejvýznamnější bity všech koeficientů všech bloků, ve 2. skenu druhé nejvýznamnější bity atd.

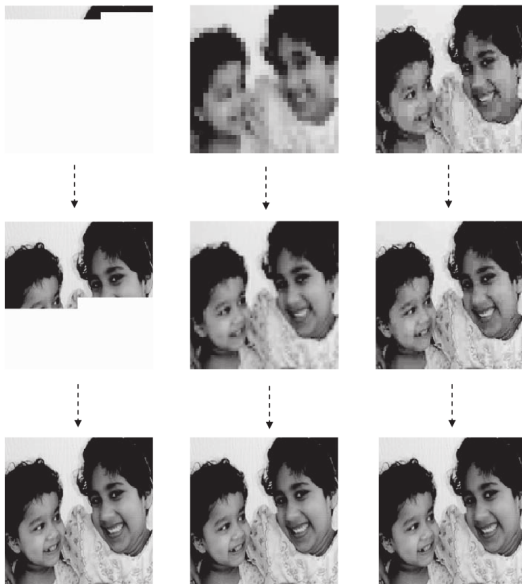
Obrázek: Obr. Baseline a progresivní uložení v JPEG

- JPEG 2000: samo o sobě (**hierarchická organizace**), na dané úrovni rekurze (rozlišení) nejdříve nízkofrekvenční pásma

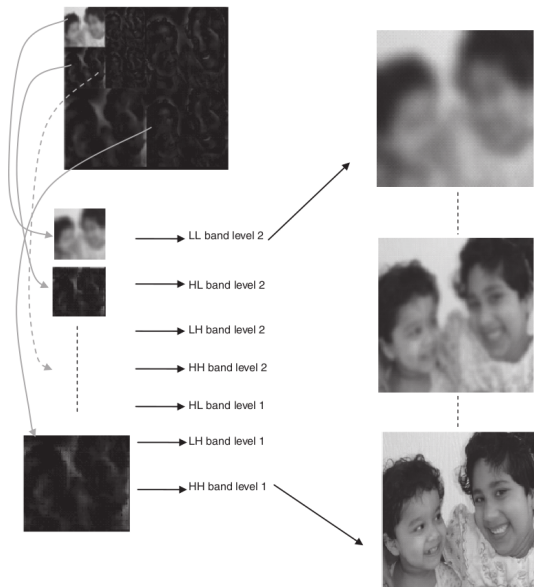
Obrázek: Obr. Progresivní uložení v JPEG 2000



Progresivní uložení komprimovaného obrazu



Progresivní uložení komprimovaného obrazu





Kompresa videa



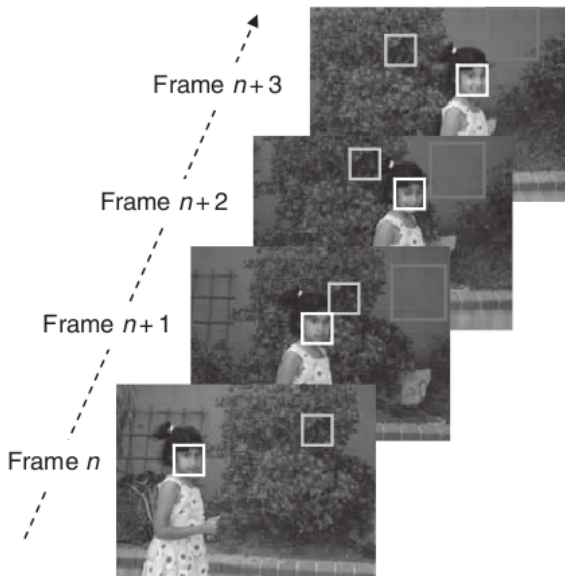
- video významné podobně jako obraz – zábavní průmysl, nárůstající na rozlišení a objemu, real-time distribuce → **(nezbytná) potřeba komprese**
- např. 720×576 , 12bpp (DVD), 25fps prokládaně ~ 120 Mbps → 9.8 Mbps max, běžně 2 Mbps (MPEG-2), 1920×1080 , 12bpp (HDTV), 25fps prokládaně ~ 590 Mbps → 8 Mbps (MPEG-2), 4 Mbps (MPEG-4)
- např. standardy (ISO) MPEG-x, (ITU) H.26x
- metody založené na technikách **predikce a kompenzace pohybu**

- u obrazu prostorová (a spektrální) redundance – oblasti podobných pixelů (málo frekvencí)
- video = sekvence obrazů → **kompresi jednotlivých snímků**, tzv. **intra-frame** – např. formáty **M-JPEG (Motion JPEG) (2000)**, **DV**, u aplikací s nízkou frame rate (1–2 fps), např. instant messaging, videokonference, kamery pro ostrahu, nízká komprese (pro video)
- = **(lokální) korelace barev pixelů v čase/mezi snímky** – pixely zůstávají podobné, např. pozadí, postupné změny

Obrázek: Obr. Časová redundance, podobné snímky

- uložení pouze **změn mezi snímky**, tzv. **rozdílových snímků** – menší entropie, podobné DPCM

Obrázek: Obr. Rozdílový snímek





Frame n



Frame $n+1$



Frame $(n+1) - \text{Frame } n$



Frame n



Frame $n+1$



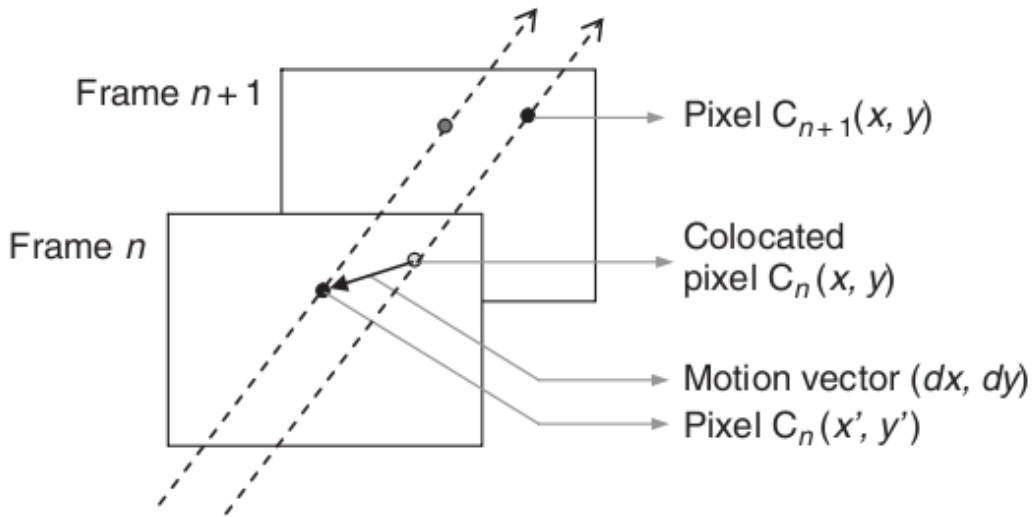
Frame $(n+1) - \text{Frame } n$

- další **časová redundance**: **pohyb (lokálních) oblastí pixelů** – objektů, při pohybu kamery

Obrázek: Obr. Časová redundance, pohyb oblastí

- entropie rozdílového snímku nepřímo závislá na množství a rychlosti pohybů – plynulý, malý vzhledem k frame rate \Rightarrow predikovatelný \rightarrow **predikce pohybů oblastí**
 - barva pixelu na pozici (x, y) ve snímku n : $C_n(x, y)$
 - $C_{n+1}(\mathbf{x}, \mathbf{y}) = C_n(\mathbf{x} - \mathbf{dx}, \mathbf{y} - \mathbf{dy}) + e(\mathbf{x}, \mathbf{y})$ – posunutí pixelu o **vektor pohybu** (dx, dy) (v důsledku pohybu objektu nebo kamery) plus chyba $e(x, y)$ (kvantizační chyba, šum, změna světelných podmínek apod.)
- \rightarrow predikce $C_{n+1}(x, y)$ jako $C_n(x - dx, y - dy)$ a opravení o $e(x, y)$

Obrázek: Obr. Predikce pohybu pixelu



- stejný pohyb (lokálních) oblastí pixelů
- = rozklad snímku na tzv. **makrobloky** (čtvercové, stejně velké) a **predikce makrobloků** ze stejně velkých (libovolně umístěných) oblastí předchozího, **referenčního, snímku**
- ne vždy možná – když pro makroblok není oblast
- predikce (cílových) snímků z referenčních snímků = **kompensace pohybu (motion compensation)**

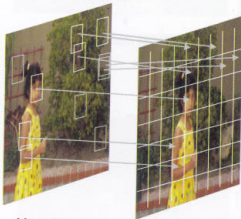
Obrázek: Obr. Predikce snímků



Frame n



Frame $n+1$



Macroblock prediction of frame $n+1$ from regions in frame n



Reconstructed frame $n+1$ by macroblock prediction



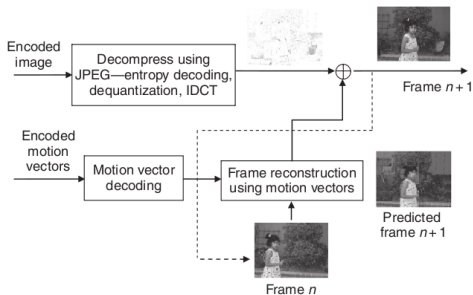
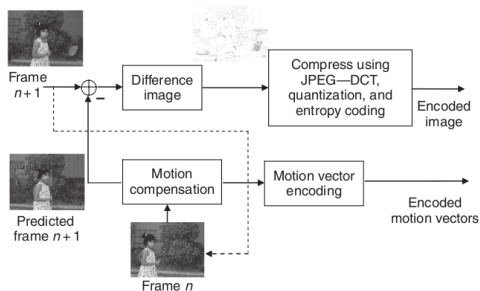
Frame difference
frame $n+1$ - frame n



Frame difference
Reconstructed frame $n+1$ - frame $n+1$

- 1 první snímek přímo komprimován **obrazovou kompresí**, např. JPEG – tzv. **intra mód**
- 2 pro další snímky – tzv. **inter mód** – **nalezení vektorů pohybu** (a tedy oblastí v referenčním snímku) pro makrobloky
- 3 **vytvoření predikovaného** (pohybově kompenzovaného) **snímku** = zkopírování oblastí z referenčního snímku daných vektory pohybu do makrobloků
- 4 **odvození chybového snímku** = rozdíl mezi skutečným a predikovaným snímkem
- 5 **kompresa chybového snímku** obrazovou kompresí, např. JPEG, (ztrátové) **a vektorů pohybu** pomocí statistického kódování (Huffmanova nebo aritmetického, bezztrátové, se statickým empirickým modelem)

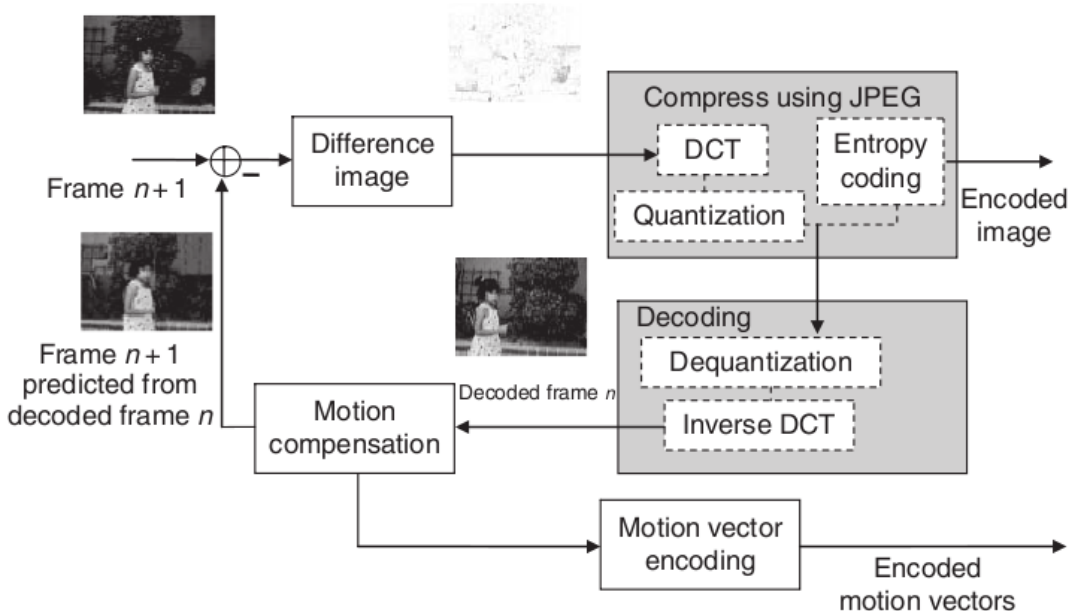
Obrázek: Obr. Kompresa (open loop případ) a dekomprese



- open loop vs. closed loop případ – jako DPCM pro snímky jako hodnoty signálu

Obrázek: Obr. Closed loop případ komprese

Obrázek: Obr. Pseudokódy komprese a dekomprese



```
referenceFrame = NULL;
While input frames are available
{
    currentFrame = getNextFrame();
    if (intramode)
    {
        codedFrame = codeFrame (currentFrame); //Using JPEG pipeline
        dumpToStream (codedFrame);
        reconstructedFrame = decodeFrame (codedFrame);
        referenceFrame = reconstructedFrame;
    }
    else if (intermode)
    {
        for each macroblock in image
        {
            // motion vector computed for each macroblock
            mVs[i] = computeMotionVector(macroblock, referenceFrame);
        }

        // Create a motion compensated frame by copying blocks from reference
        // frame using computed motion vectors
        predictedFrame = compensateFrame(mVs, referenceFrame);

        // Compute the error Image
        errorFrame = currentFrame - predictedFrame;

        codedFrame = codeFrame(errorFrame); //Using JPEG pipeline
        codedMvs = codeMotionVectors (mVs); // entropy code motion vectors
        dumpToStream (codedMvs);
        dumpToStream (codedFrame);
    }
}
```

```
referenceFrame = NULL;
While encoded data is available
{
    currentCodedFrame = getNextCodedFrame();
    currentCodedMvs = getNextCodedMotionVectors();

    if (intramode)
    {
        frame = decodeFrame (currentCodedFrame); // Using JPEG pipeline
        referenceFrame = frame;
        sendToDisplay (frame);
    }
    else if (intermode)
    {
        errorFrame = decodeFrame (currentCodedFrame);
        // Using JPEG pipeline
        mVs = decodeMotionVectors (currentCodedMvs);
        predictedFrame = compensateFrame (mv, referenceFrame);
        frame = predictedFrame + errorFrame;
        sendToDisplay (frame)
    }
}
```

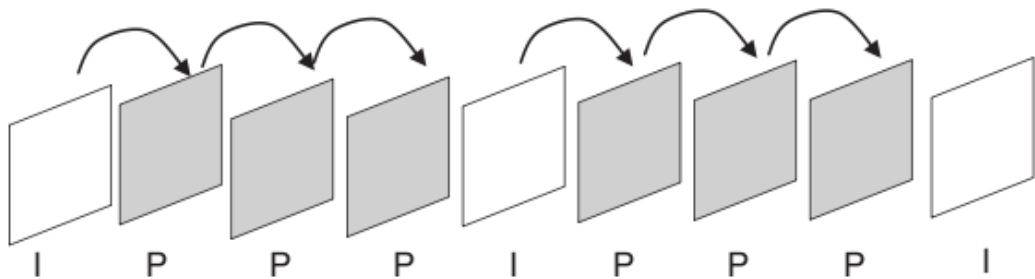
Snímky I

- = **komprimované** přímo, samostatně **bez predikce**, obrazovou kompresí, např. JPEG, v intra módu \Rightarrow snadná komprese, ale větší bit rate než predikované snímky
- (typicky) kódované **periodicky** v posloupnosti snímků, **referenční snímky** pro predikci
 - když by chybový snímek byl příliš obsáhlý (měl velkou entropii = velký bit rate, např. při změně scény) nebo by jeho výpočet trval příliš dlouho (real-time přenos) \rightarrow možnost detekce

Snímky P

- = **prediktivně (P) komprimované z (přímo) předchozího snímku I nebo P** \Rightarrow náročná komprese, ale nižší bit rate než snímky I
- chybový snímek má nižší entropii než snímek I \rightarrow hrubější kvantizace (např. vyšší kvantizační faktory u JPEG, typicky čtyřnásobné)
- vytváří dopřednou závislost mezi snímky

Obrázek: Obr. Snímky P

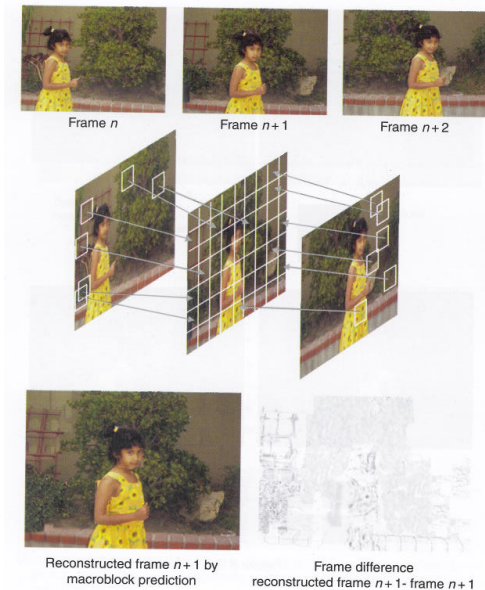


Snímky B

- = **dvousměrně** (bidirectionally, B) **prediktivně komprimované z předchozího a následujícího snímku I nebo P** \Rightarrow náročnější komprese, ale nižší bit rate než snímky P
- někdy je pro kompenzaci pohybu lepší následující než předchozí snímek – např. objevení se objektů při pohybu

Obrázek: Obr. Predikce snímků B

- vyhledání a **výběr z vektorů pohybu** (a tedy oblastí odpovídajících makroblokům) pro oba referenční snímky **nebo použití obou vektorů** (a interpolace oblastí jako predikce makrobloku)



Snímky B

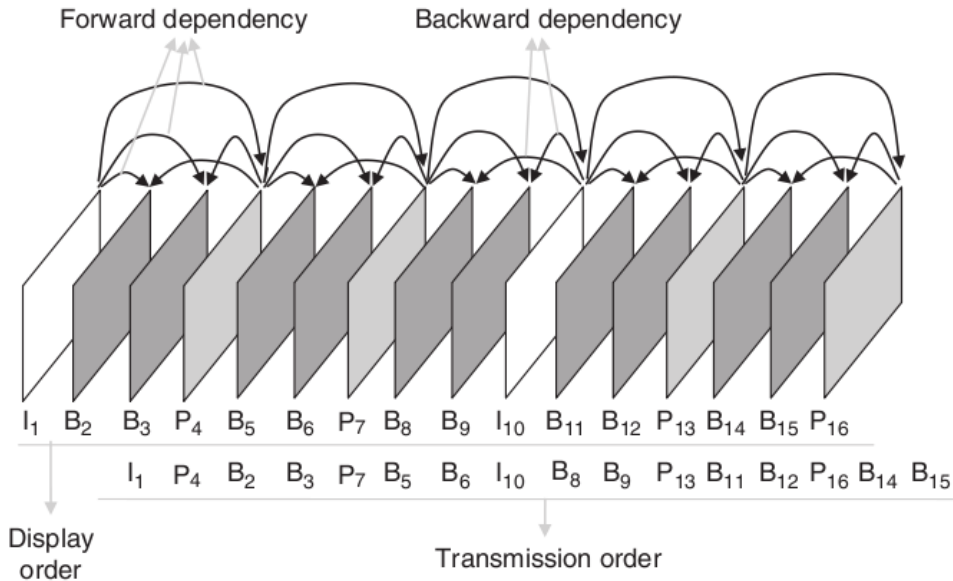
- vytváří dopřednou i **zpětnou závislost mezi snímky** → potřeba je při uložení/distribuci kvůli možnosti dekomprese **přeskládat** do tzv. **přenosového pořadí** (snímky B před referenčními snímky I nebo P) – může způsobovat prodlevy při zobrazení (v zobrazovacím pořadí), **bufferování snímků**

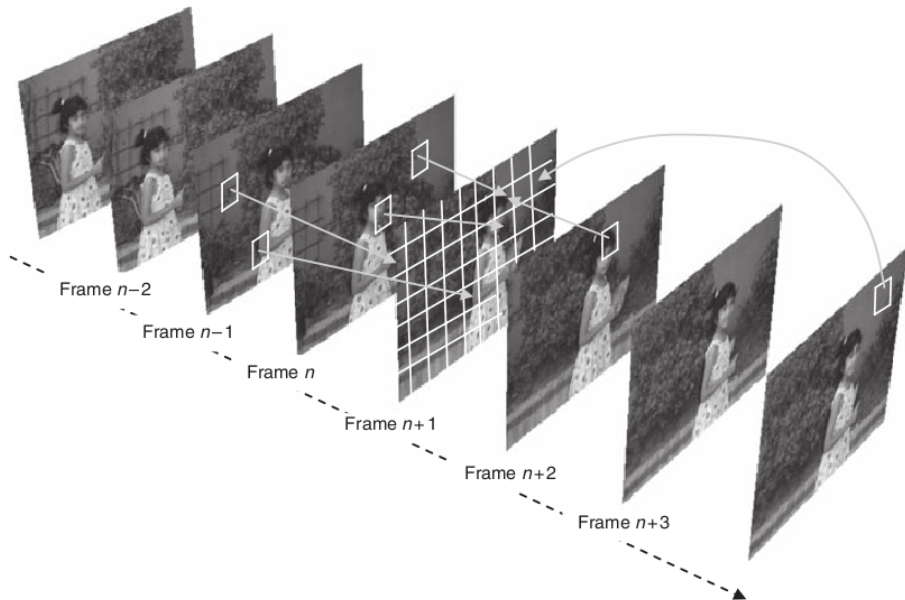
Obrázek: Obr. Snímky B

Multiframe predikce

- = **predikce z více předchozích i následujících snímků** ⇒ lepší, náročnější, ale menší bit rate

Obrázek: Obr. Multiframe predikce





- = posloupnost kódovaných snímků daná frekvencí a použitím typů snímků, **opakující se skupiny snímků**, tzv. **group of pictures (GOP)**
- první snímek I, pak několik (včetně 0) snímků P a mezi nimi snímky B
- reprezentována počtem snímků a vzdáleností (v počtu snímků + 1) mezi nejbližšími I/P snímky nebo počty snímků P mezi I a B mezi P

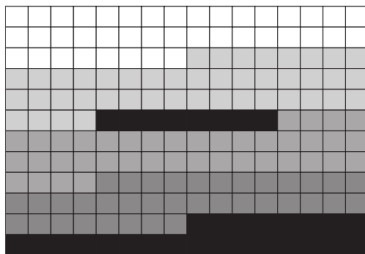
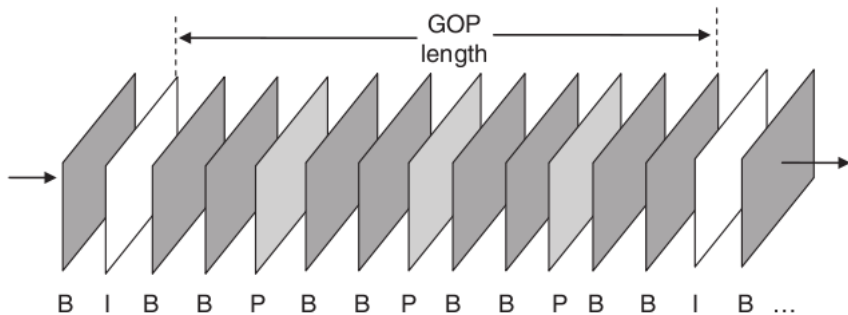
Obrázek: Obr. GOP (MPEG-1)

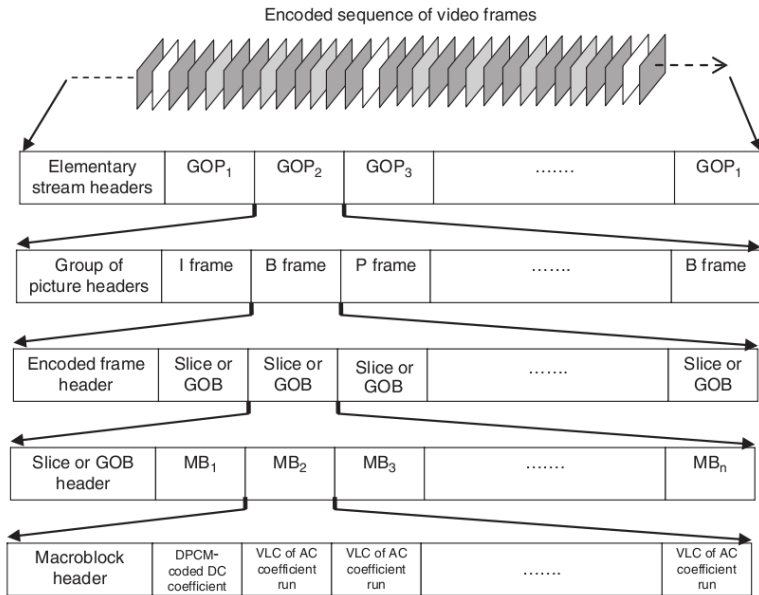
- podporuje **náhodný přístup** (např. „přetáčení“) – k snímkům I
- **slice/GOB** (group of macroblocks) = skupina makrobloků ze snímku, pro variabilní kódování částí snímků

Obrázek: Obr. Slices/GOBs (MEPG-1)

- standardově závislé

Obrázek: Obr. Bitový proud



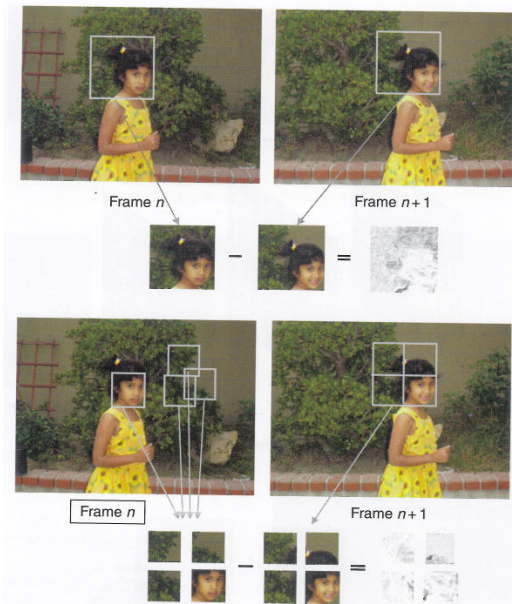


- = prohledávání plochy v referenčním snímku kolem korespondující (kosituované) pozice makrobloku v cílovém snímku na nejpodobnější oblast (**referenční makroblok**) a určení vektoru pohybu
- barvy pixelů makrobloku: $C_{n+1}(x, y)$, kde $x, y \in [0, m]$ a m je velikost makrobloku

Velikost makrobloků

- = kompromis mezi menšími (větší počet makrobloků a vektorů pohybu, ale vektory menší = lepší chybový snímek) a většími (méně vektorů, ale horší chybový snímek – potenciálně různé oblasti pro části makrobloku)
- ve standardech **16 × 16 pixelů** nebo variabilní, pak např. pro snímek 1920 × 1080 (HDTV) je 8100 makrobloků

Obrázek: Obr. Velikost makrobloku



Míry nepodobnosti/rozdílu oblasti a makrobloku

- mean absolute difference (MAD):

$$MAD(dx, dy) = \frac{\sum_{x=1}^m \sum_{y=1}^m |C_{n+1}(x, y) - C_n(x + dx, y + dy)|}{m^2}$$

- **sum of absolute difference (SAD)**, používanější): $MAD \cdot m^2$
- mean square difference (MSD): $MAD^2 \cdot m^2$
- **pel difference** classification, kde t je zvolený práh shody a $\|p\| = 1$ pro pravdivé p , jinak 0:

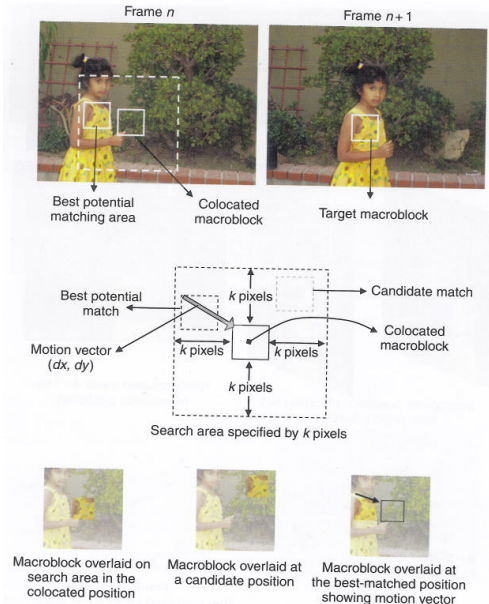
$$PEL(dx, dy) = \sum_{x=1}^m \sum_{y=1}^m \mathbb{1}_{\|C_{n+1}(x, y) - C_n(x + dx, y + dy)\| \leq t}$$

- pro $dx, dy \in [-k, k]$

- **prohledávaná plocha** = čtverec s hranami k pixelů od hran makrobloku, k je parametr vyhledávání, typicky 0 až 32, volený podle rychlosti pohybu objektů/oblastí mezi snímky

Obrázek: Obr. Vyhledávání vektoru pohybu

- nalezení oblasti/vektoru – kritický, **centrální problém**: lepší vektory pohybu (= podobnější oblast) = lepší predikce makrobloku = lepší chybový makroblok (menší entropie) = menší bit rate, ale delší doba vyhledávání (která nemusí být dostupná, např. u real-time komprese/přenosu), typicky 60–80 % celkového času komprese



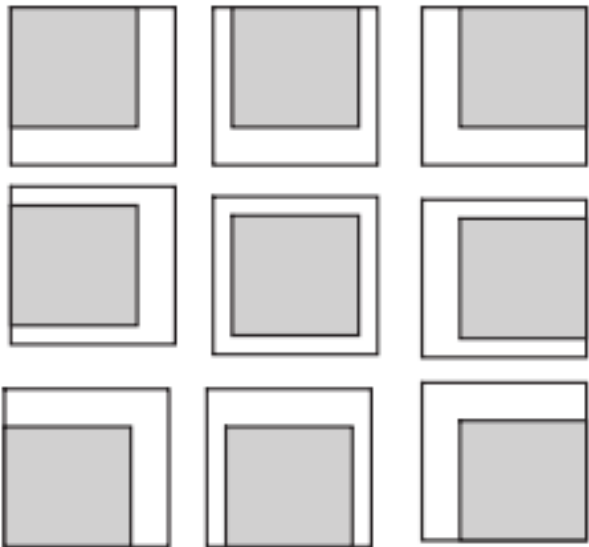
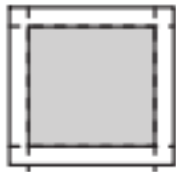
- složitost porovnání oblasti a makrobloku (výpočet míry rozdílu): $O(m^2)$
- složitost v počtu porovnání při hledání nejpodobnější oblasti/nejlepšího vektoru pohybu
- algoritmus jako další parametr vyhledávání, vedle k

Sekvenční/Hrubá síla

= porovnání na **všech** $(2k + 1)^2$ **pozicích**

Obrázek: Obr. Sekvenční vyhledávání pro $k = 1$

- např. pro snímek 1920×1080 (HDTV) s makrobloky 16×16 a $k = 32$ je přes 34 mil. porovnání
- nepoužitelné pro real-time přenos, ale záruka nejlepších vektorů = nejlepší komprese



Logaritmické/Binární (Fast motion estimation, FME)

- rozdíl oblasti a makrobloku se často monotónně zmenšuje k nejpodobnější oblasti, ve směru pohybu
- = opakované porovnání na 9 (v 1. iteraci) a 8 pozicích s $dx, dy \in \{-\frac{k}{2}, 0, \frac{k}{2}\}$ bez $(0, 0)$ (od 2. iterace), výběr nejlepší a zmenšení k na polovinu = **zpřesňování vektoru pohybu**: $8 \log_2 k + 1$ porovnání

Obrázek: Obr. Logaritmické vyhledávání pro $k = 16$

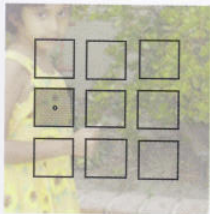
- např. pro snímek 1920×1080 (HDTV) s makrobloky 16×16 a $k = 32$ je přes 330 tis. porovnání



Search area around the colocation



Macroblock of frame $n+1$



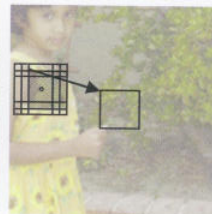
Iteration 1



Iteration 2



Iteration 3



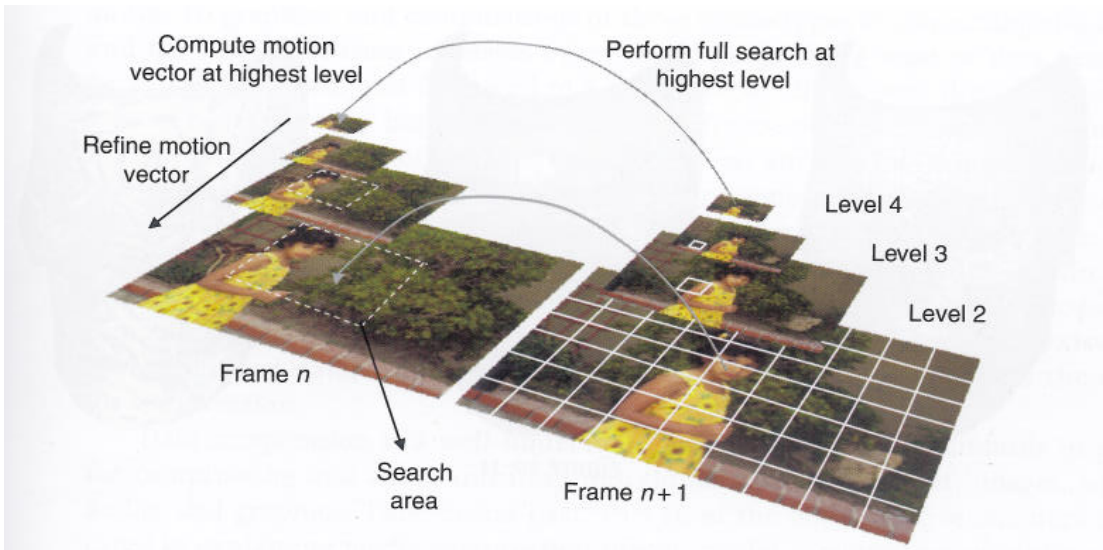
Iteration 4 showing motion vector

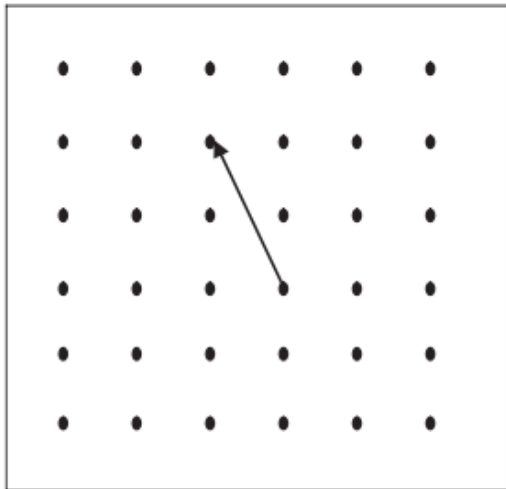
Hierarchické

- vytvoření **hierarchie verzí snímků** pro cílový i referenční snímek **se zmenšujícím se rozlišením** na polovinu (podvzorkování a filtrace) – zmenšuje se i velikost makrobloků/oblastí a k
- = sekvenční vyhledání na nejvyšší úrovni (s nejnižším rozlišením a nejmenšími makrobloky/oblastmi a k), pak logaritmické na nižších úrovních = zpřesňování vektoru pohybu

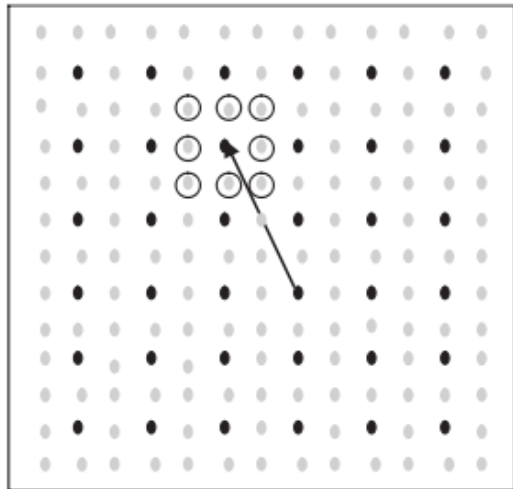
Obrázek: Obr. Hierarchické vyhledávání

- např. pro snímek 1920×1080 (HDTV) s makrobloky 16×16 , $k = 32$ a čtyřmi úrovněmi je přes 850 tis. porovnání





Level $k + 1$



Level k

- od 1984 (H.120), pro aplikace dané doby
- některé standardizované – ISO MPEG-x, ITU H.26x
- standardy definují komprimovaný bitový proud (**pro dekódování**), ne konkrétní algoritmy kodeku (implementace)
- patentované, s poplatky za užití (MPEG-x/H.26x) i bez (Theora, VPx), svobodné (AV1) – otázka pro web

H.261 (kodek FFmpeg)

- 1988, \sim px64 – pro ISDN (rychlost v násobcích 64 kbps)
- až 352×288 (CIF), jen progresivní, snímky jen I (JPEG) a P

MPEG-1 (FFmpeg)

- 1991, i audio, použití ve **Video CD (VCD)** – srovnatelné s VHS
- $352 \times 240/288$ (SIF), jen progresivní, až 1.5 Mbps
- **half pixel** kompenzace pohybu (**HPEL**, zvýšení přesnosti, lepší predikce), použití slices/GOBs u predikovaných snímků

H.262/MPEG-2 (FFmpeg, x262)

- 1994, i audio, pro **digitální TV** (SIF/**SDTV**, pak **HDTV**), použití v **DVD**
- **kontejnerové formáty program stream (PS)** pro uložení na disková média, a **transport stream (TS)** pro přenos chybovým kanálem se synchronizací
- i prokládané – **predikce pulsů** z pulsů/snímků, až 20 Mbps (2 SDTV, 15–18 HDTV)
- half pixel kompenzace pohybu, **škálovatelné kódování** – možnost dekódovat různá rozlišení, base (nižší) a enhancement (vyšší frekvence) **vrstva videa**

H.263/MPEG-4 (Advanced Simple Profile) (FFmpeg, DivX, Xvid)

- 1999, i audio a 2D/3D grafika – **media objekty** = více proudů (videa, audia, grafiky)
- i prokládané, rozsah bit rate od 64 kbps (streaming, wireless) do 4 Mbps
- **quarter pixel (QPEL)** a globální kompenzace pohybu, popisy rozmístění objektů, **roviny videa** (video object planes, VOPs) – libovolný neobdélníkový tvar snímků, různé bit rate, skládání rovin v přehrávači (např. klíčování), synchronizovaná **textová a metadatová stopa** (na úrovni snímků)
- nekompatibilní varianta MS MPEG-4v3 (kodek DivX)

H.264/MPEG-4 AVC (Advanced Video Coding) (x264, FFmpeg, OpenH264, DivX, Quicktime)

- 2004/2003, pro **digitální TV (HDTV)**, **Blu-ray** (a HD-DVD), internet streaming (Youtube, Netflix aj.), web atd.
- **menší bloky** u JPEG až do 4×4 pixely, v intra módu slices/GOBs bloků s **prostorovou predikcí** bloků z předchozích a DCT rozdílů, tzv. **deblocking filter** – prevence blokových artefaktů DCT na hranicích bloků, v enkoderu i dekoderu
- **variabilní velikost makrobloků**: hierarchický rozklad na 8×16 , 16×8 a 8×8 a poslední na 8×4 , 4×8 a 4×4 – lepší predikce (na hranicích pohybujících se objektů)
- multiframe predikce, kontextově adaptivní binární aritmetické kódování (CABAC, kontextově adaptivní model) vektorů pohybu a chybových snímků
- varianty MPEG-4 Web Video Coding, Scalable Video Coding, XAVC

DODELAT

H.265/MPEG-H HEVC (High Efficiency Video Coding) (x265)



Theora (libtheora, FFmpeg)

- Xiph.org

VP6–9 (libvpx, FFmpeg)

- Google (On2), použití v Adobe Flash

AV1 (aomenc)



Další

- Windows Media Video (WMV), Apple ProRes, Apple Sorenson, Cinepak (~ Apple Compact Video), RealVideo, Flash, Dirac

- **variabilní (VBR)** – „pomalé“ a „rychlé“ pasáže videa, čím více pohybu na sousedních snímcích, tím vyšší bit rate (větší, nepřesnější, vektory pohybu, obsáhlejší chybové snímky)
- **konstantní (CBR)** – ideálně, pro přenos, zvláště real-time – omezená rychlost (šířka pásma) síť → **řízení bit rate**

Řízení

- **pevná GOP a bit rate pro GOP**
 - pevné bit rate pro snímky – empiricky snímek I = 3× snímek P = 6 – 15× snímek B ⇒ horší kvalita (variabilní složitost obsahu)
 - **variabilní bitrate pro snímky** – cache snímků, adaptivní kódování podle množství pohybu, méně = více bitů pro snímky I, více = více bitů pro snímky P, méně bitů pro snímky B, (proprietární) heuristiky
- variabilní bit rate pro GOP – omezené rychlostí a velikostí bufferu dekodéru

- předzpracování videa – low-pass filtry pro **odstranění šumu a chyb** (při digitalizaci z analogového signálu)

Parametry

- **rozlíšení, frame rate** – standardizované (PAL/NTSC, DVD apod.)
- **bit rate, CBR/VBR, počet průchodů** – více pro CBR, první pro statistickou analýzu pro variabilní alokaci bit rate, adaptace kompresních modelů atd.
- struktura GOP: počty snímků P mezi I – u „rychlého“ videa menší (horší predikce), B mezi P – kompromis mezi bit rate a dobou komprese, lepší odolost proti chybám přenosu při více snímcích I a P
- obraz: kvantizační tabulky (J/MPEG, ITU) a faktor („JPEG kvalita“)
- kompenzace pohybu: velikost prohledávané plochy a vyhledávací algoritmus, half/quarter pixel vyhledávání
- přeskokování snímků – při omezené rychlosti přenosu
- **informace pro dekoder (DSI)** – vkládat parametry komprese průběžně, nejen do záhlaví, např. u streamu
- ...