Zpracování videa

Video

- **Video** – časová posloupnost snímků

- Video je reprezentováno datovou strukturou, kterou si v nejjednodušší podobě můžeme představit jako datový soubor, kde jsou uloženy jednotlivé snímky v nějakém formátu za sebou s tím, že snímky pořadí snímků odpovídá okamžiku, kdy byly zaznamenány.
- Při dostatečné frekvenci snímání/ukládání, resp. přehrávání (fps - frames per second) dostáváme vizuální vjem spojitě probíhající události.

Video

- Vzhledem k tomu, že často potřebujeme ukládat velké množství snímků, prosté uložení snímků za sebou by bylo paměťově náročné a proto využíváme vhodné techniky **kódování** a **komprese**.

- Koncepčně nejjednodušší je kódovat a komprimovat jednotlivé obrázky bez kontextu s ostatními (sousedícími) obrázky videa.
- Pro kompresi se využívá nejčastěji standardu JPEG nebo JPEG2000.
- Video, které používá JPEG formát se nazývá Motion JPEG. Není to ale mezinárodní standard. Mohou být problémy s různou implementací JPEG.
- Při použití JPEG2000 je kompatibilita zaručena, kódování je o něco efektivnější, ale implementace JPEG2000 je složitější.

Video

- Prostorové/časové kódování videa (Spatial/temporal video coding)
  - Frame Replenishment Coding
  - Differential Frame Coding
  - Unitary Transform Interframe Coding
  - Motion Compensatioan Interframe Coding

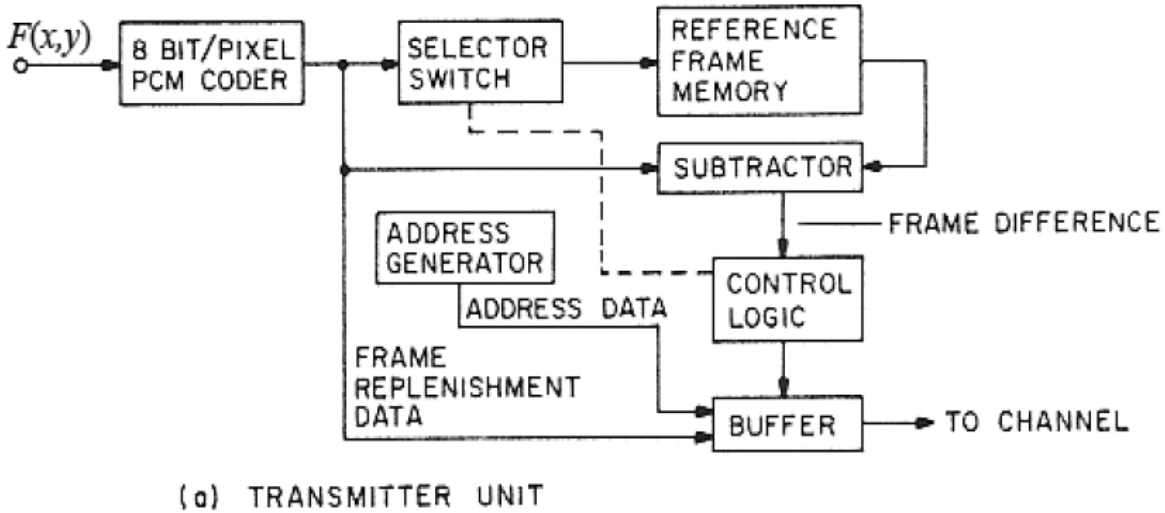
- Uvedené techniky vedou ke standardizovaným formátům MPEG-1 a MPEG-2.

Video

- Frame Replenishment Coding
  - Většinou se sousední snímky i při pohybu příliš nemění, takže místo přenášení celého snímku stačí přenášet pouze rozdíl mezi oběma snímky.
  - Předpokládejme, že každý pixel je reprezentován 8 bity.
  - První referenční snímek je uložen celý (frame reference memory)
  - Následující snímky jsou pixel po pixelu porovnávány s referenčním snímkem
  - Pokud existuje významný rozdíl mezi aktuálním a referenčním pixelem, je referenční pixel nahrazen aktuální pixelem a současně je aktuální pixel zařazen do bufferu pro záznam/přenos.
  - Současně je zaznamenána i příslušná pozice změny v rámci zaznamenávaného/přenášeného řádku pixelů.
  - Pro zvýšení efektivity přenosu se nepřenáší jednotlivé pixely, ale shluky pixelů (např. celé řádky změn)

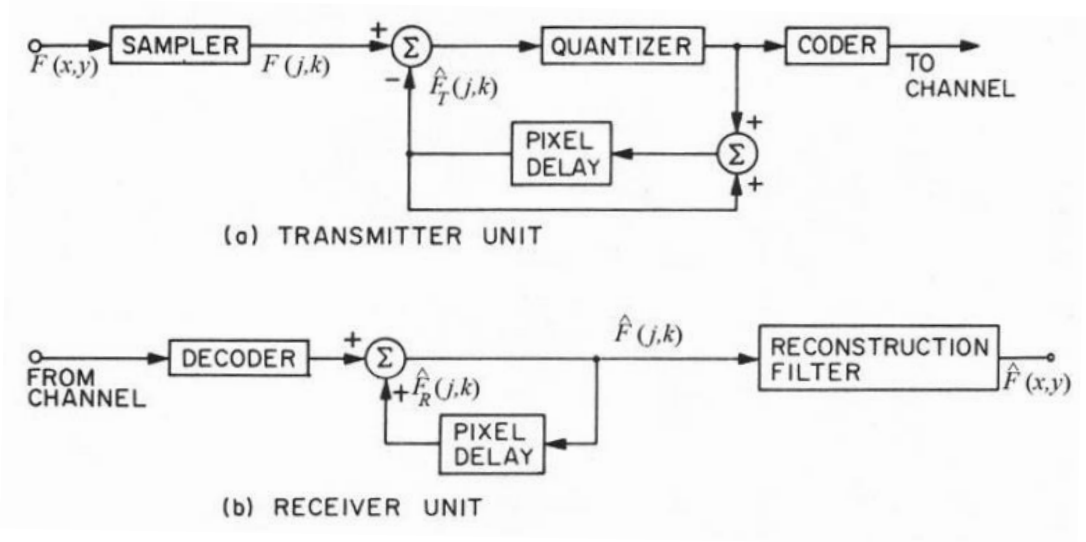
Video

- Frame Replenishment Coding



Video

- Differential Frame Coding
  - Předchozí technika Frame Replenishment Coding může být rozšířena o tzv. **prediktivní kódování**, aby se využila redundance mezi snímky.
  - Diferenční pulzní kódová modulace (DPCM) je využita pro predikci hodnoty pixelu v přenášeném řádku na základě z předchozího snímku



Video

- Unitary Transform Interframe Coding
  - V běžném video záznamu je významná korelace mezi časovým a prostorovým rozdílem (posunem) mezi obrázky.
  - Lze proto využít 3D unitární transformaci, která odstraní korelaci mezi snímky a zefektivní kódování.
  - Označme F(j,k,i) třírozměrnou matici (tenzor) JxKxI pixelů extrahovanou z posloupnosti I snímků. 3D unitární transformace je pak definována obecným vztahem

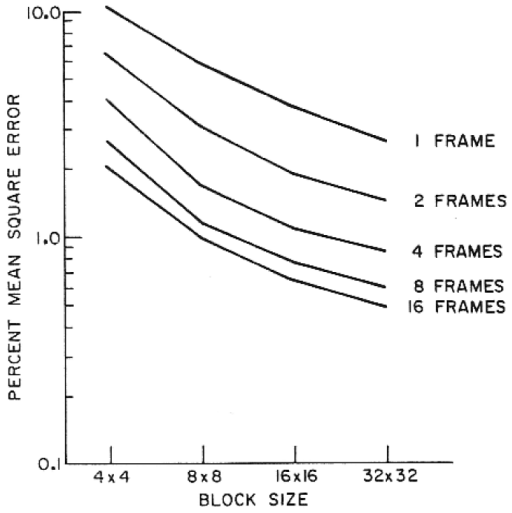
$$\mathcal{F}(u, v, w) = \sum_{j} \sum_{k} \sum_{i} F(j, k, i) A_C(j, u) A_R(k, v) A_T(i, w)$$

  - kde $A_C(j, u)$, $A_R(k, v)$ a $A_T(i, w)$ jsou postupně sloupcové, řádkové a časové transformační jádro (kernel).

Video

- Unitary Transform Interframe Coding
  - Při využití unitární transformace pro kódování rozdílů sousedních snímků je každý koeficient kódován na základě zónového vzorkování, zónového kódování nebo pomocí prahování a to s využitím podobných technik jaké se využívají pro kódování pomocí dvojrozměrné transformace.
  - Inverzní transformací získáme rekonstruované bloky snímků.
  - Nevýhodou této metody je potřeba úschovy velkého množství snímků.

Závislost chyby na velikosti transformovaného bloku pro kosinovou transformaci

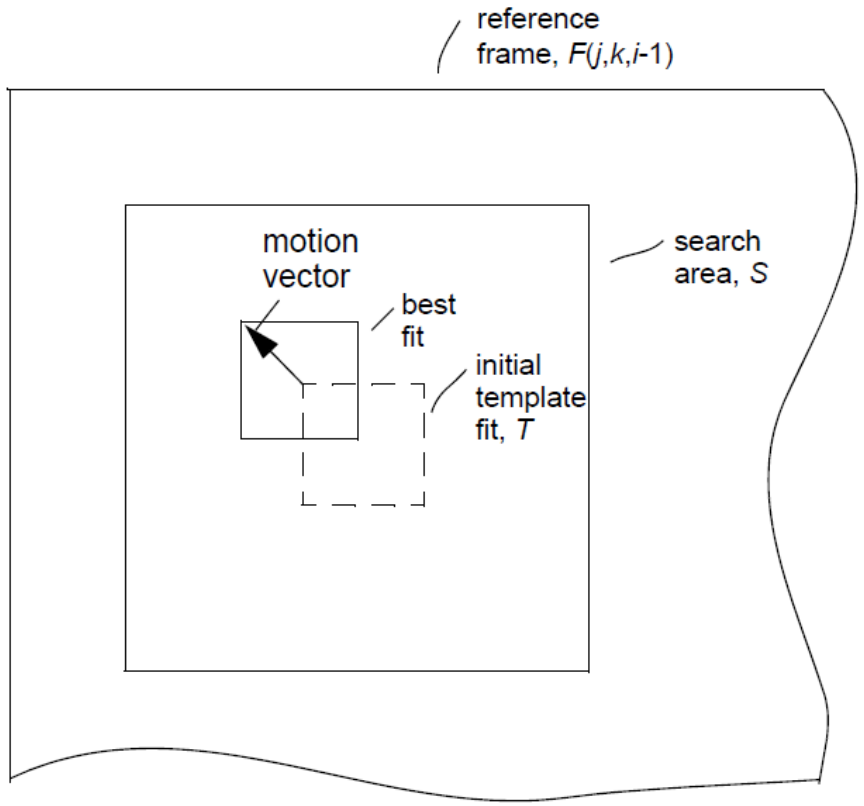


Video

- Motion Compensatioan Interframe Coding, 1969
  - Pozorování: I při pomalém pohybu snímaného objektu nebo pohybu kamery dochází k poměrně výrazným změnám mezi snímky, což omezuje využití předchozích technik založených na rozdílových snímcích.
  - Tento problém může být snížen pomocí tzv. **kompenzace pohybu**.
  - Myšlenka spočívá v nalezení oblasti T (zvolené v aktuálním snímku, tzv. template) v definované oblasti S v referenčním snímku.
  - Nejlepší (nejpodobnější) nalezená oblast je použita pro následný výpočet rozdílového snímku.
  - Nejčastěji je jako míra podobnosti použita křížová korelace (korelace mezi různé posunutými oblastmi).
  - Velikost oblasti S určuje míru předpokládaného posunu objektu/kamery mezi dvěma snímky.

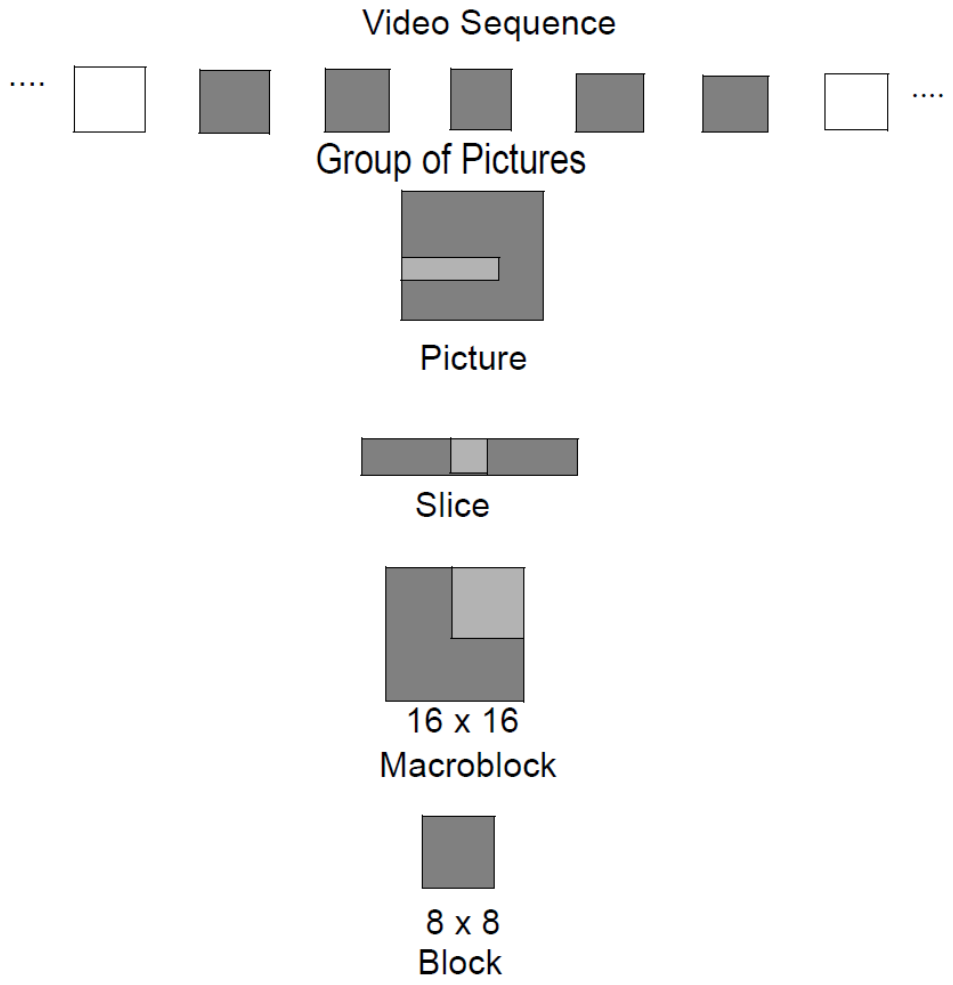
Video

- Motion Compensatioan Interframe Coding, 1969



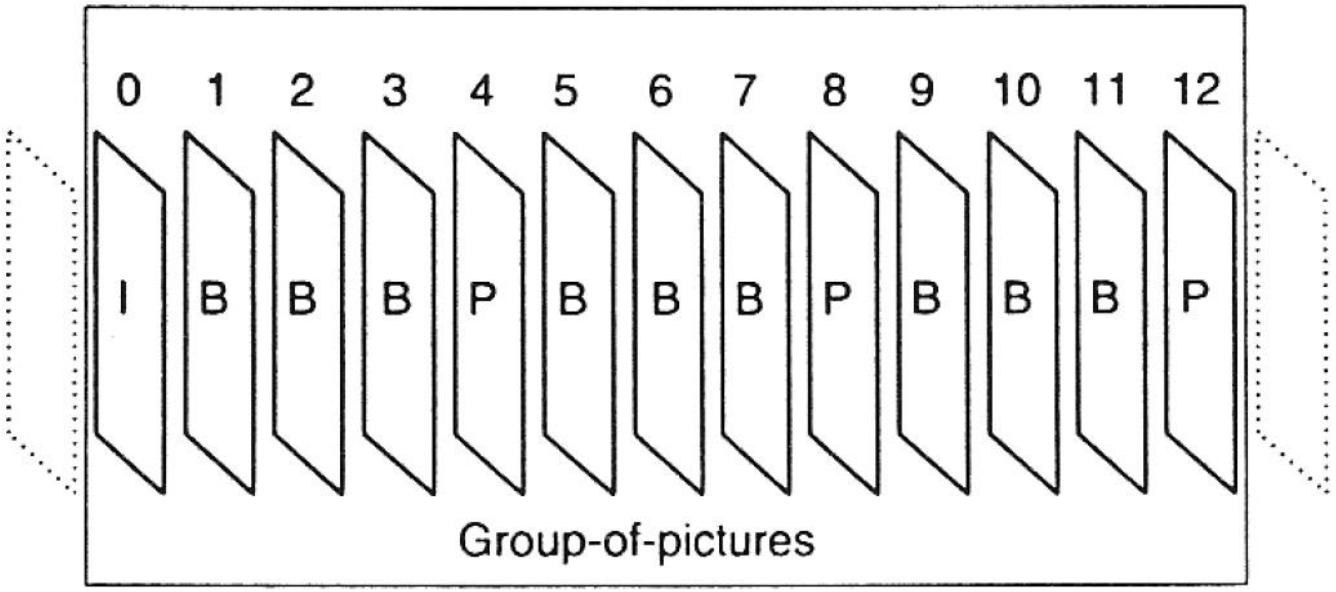
Video

- MPEG-1
  - Formát pro ukládání videí
  - Postaven na hierarchické vrstevnaté datové struktuře, která slouží ke kompresi časové posloupnosti snímků.
  - Na každé úrovni od sekvence snímků až po malý blok je barevný snímek popsán jasovým polem Y(j, k) a dvěma poli s intenzitami barev (chrominancemi) $C_b(j, k)$ a $C_r(j, k)$.



Video

- MPEG-1
  - Datové struktury
    - Posloupnost snímků (Video Sequence) – časová kolekce jedné nebo více skupin snímků
    - Skupina snímků (Group of Pictures) - časová kolekce jednoho nebo více snímků. Začíná tzv. I-snímkem nebo B-snímkem.



Video

- MPEG-1
  - MPEG-1 snímky (pictures)
    - **I-snímek** – 3 bloky velikosti 8x8 obsahující jas Y a intenzity barev C_b a C_r. Všechny reprezentují rozdílové snímek, který je kódován pomocí kosinové transformace. I-snímek může být použit pro predikci P-snímků a B-snímků.
    - **P-snímek** – 3 bloky velikosti 8x8 obsahující jas Y a intenzity barev C_b a C_r. Všechny reprezentují kompenzaci pohybu od I- nebo P-snímku. P-snímek může být použit pro predikci P-snímků a B-snímků.
    - **B-snímek** – 3 bloky velikosti 8x8 obsahující jas Y a intenzity barev C_b a C_r. Všechny reprezentují referenční snímek a to buď předchozí a/nebo následující snímek pro obousměrnou kompenzaci pohybu. B-snímek nemůže být použit jako referenční snímek pro jiné snímky (slouží je pro kompenzaci pohybu nikoliv jako výchozí snímek vůči kterému by se počítal rozdíl pro jiné snímky).
    - **D-snímek** – 3 bloky velikosti 8x8 obsahující jas Y a intenzity barev C_b a C_r. Všechny reprezentují stejnosměrné složky (offset, posun).

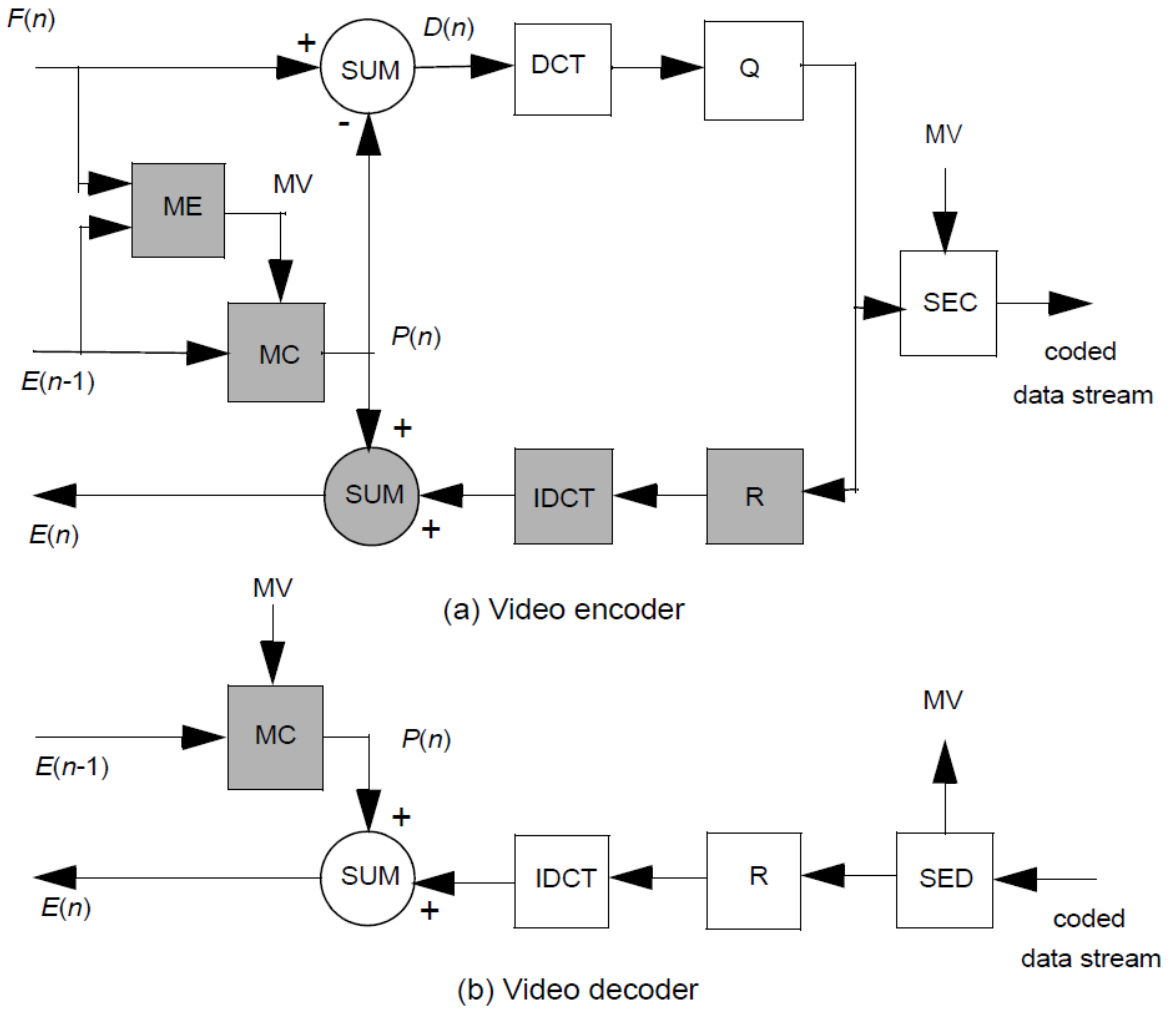
Video

- MPEG-1
  - MPEG-1 **řez** (slice) – souvislá posloupnost makrobloků v obrázku. Výška řezu je 16 pixelů. Na hraně snímku může řez přesahovat do dalšího řádku makrobloku (wrap around). Účelem řezu je umožnit změny v algoritmu kodéru za běhu. Je také užitečný pro kompenzaci/zotavení se z chyb.
  - MPEG-1 **makroblok** (macroblock) – skládá se ze 3 bloků velikosti 16x16, které obsahují jas Y a intenzity barev C_b a C_r. Jas Y je v makrobloku rozložen do 4 polí o velikosti 8x8. Jas Y může být využit pro odhad pohybu.
  - MPEG-1 **blok** (block) – blok velikosti 8x8 pixelů, který obsahuje buď jas Y nebo nějakou složku intenzity barev C_b nebo C_r.

Video

- MPEG-1
  - Standard MPEG-1 nepředepisuje žádný konkrétní **kodér** ani **dekodér**.
  - Nicméně kodér musí splňovat určitou syntaxi a sémantiku. Podobně dekodér musí být schopen dekódovat zakódované video.
  - Na obrázku vpravo je schématicky znázorněn kodér a dekodér.
    - Pokud vynecháme šedé bloky, dostaneme kodér/dekodér pro I-snímky.
    - Pokud ponecháme šedé bloky, dostaneme kodér/dekodér pro P- a B-snímky.

DCT = Discrete Cosine Transform; Q = Quantizer; R = Reconstructor; IDCT = Inverse DCT; ME = Motion Estimate; MC = Motion Compensate; SEC = Symbol and Entropy Coder; SED = Symbol and Entropy Decoder; MV = Motion Vector



Video

- MPEG-1: Kódování I-snímků
  - 4 jasové bloky velikosti 8x8 a 2 bloky velikosti 16x16 s intenzitami barev C_b a C_r makrobloku jsou postupně extrahovány ze vstupního snímku F(n) a přivedeny na vstup bloku diskrétní kosinové transformace (DCT).
  - DCT koeficienty jsou následně kvantovány pomocí kvantizační tabulky níže, tj. hodnota každého DCT koeficientu je vydělena hodnotou v tabulce a zaokrouhlena na celé číslo.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Video

- MPEG-1: Kódování I-snímků
  - Následně jsou kvantované koeficienty vhodně zakódovány v bloku Symbol and Entropy Coder (SEC).
  - Nejprve je odečtena stejnosměrná složka (DC koeficient) aktuálního snímku od stejnosměrné složky předchozího snímku.
  - Tento rozdíl je zakódován v bloku entropického kódování (Entropy Coder), který používá např. metodu Huffmanova kódování – viz slajdy na konci za Literaturou.
  - DC koeficient a zbývajících 63 získaných AC koeficientů z Huffamnova kódování jsou v matici 8x8 cik-cak uspořádány a následně zakódovány pomocí RLE.

  - V následujících tabulkách značí
    - symbol *s* znaménkový bit, *s* = 0 pro kladné hodnoty a *s* = 1 pro záporné hodnoty,
    - EOB (end of block) informuje dekodér, že v matici nejsou další nenulové koeficienty,
    - ESC (escape) indikuje, že run/level pár není v tabulce a následných 16 bitů udává hodnotu páru.

Video

- MPEG-1: Kódování I-snímků

size	coefficient range	*Y* code	*C* code
0	0	100	00
1	-1,1	00	01
2	-3...-2,2...3	01	10
3	-7...-4,4...7	101	110
4	-15...-8,8...15	110	1110
5	-31...-16,16...31	1110	1111 0
6	-63...-32,32...63	1111 0	1111 10
7	-127...-64,64...127	1111 10	1111 110
8	-255...-128,128...255	1111 110	1111 1110

Huffmanovy kódy pro DC koeficienty jasu a intenzit barev C_b a C_r.

run/level	code
0/1	1s (first)
0/1	11s (next)
0/2	0100 s
0/3	0010 1s
0/4	0000 110s
0/5	0010 0110 s
0/6	0010 0001 s
0/7	0000 0010 10s
0/8	0000 0001 1101 s
.	.
.	.
26/1	0000 0000 1101 1s
27/1	0000 0000 0001 1111 s
28/1	0000 0000 0001 1110 s
29/1	0000 0000 0001 1101 s
30/1	0000 0000 0001 1100 s
31/1	0000 0000 0001 1011 s
EOB	10
ESC	0000 01

Huffmanovy kódy pro AC koeficienty.

Video

- MPEG-1: Kódování I-snímků
  - Při zpětné rekonstrukci snímků pomocí dekodéru se nejprve dekódují zakódované bity (RLE) aby se extrahoval DC koeficient a AC koeficienty.
  - Pomocí uvedených tabulek se zpětně rekonstruují jednotlivé symboly, které se následně de-kvantizují, čímž se získají skutečné hodnoty pro inverzní diskrétní kosinovou transformaci (IDCT)
  - Pomocí IDCT se získají makrobloky, z nichž je postupně sestaven výsledný snímek – odhad původního snímku F(n).

Video

- MPEG-1: Kódování P- a B-snímků
  - Snímek F(n) je porovnán s předchozím snímkem F(n-1).
  - Porovnávají se jasové makrobloky těchto dvou snímků. Makrobloky se vůči sobě posouvají a hledá se nejlepší shoda. Výsledkem je **vektor pohybu** (Motion Vector, MV).
  - Vektor pohybu se vede do bloku kompenzace pohybu, který vytvoří makroblok predikce P(n), který je odečten od aktuálního snímku F(n), čímž vznikne rozdílový makroblok D(n).
  - Každý jasový kvadrant velikosti 8x8 a jemu odpovídající bloky intenzit barev získaného rozdílového makrobloku D(n) jsou transformovány pomocí DCT transformace.
  - Následně jsou koeficienty DCT transformace kvantovány pomocí tabulky.
  - Kvantované koeficienty jsou následně zakódovány s využitím entropického kódování.
  - Pro P- a B-snímky jsou všechny koeficienty uvažovány jako AC koeficienty Huffmanova kódu.

  - Při **dekódování** je třeba dekodéru předložit zakódované video spolu s vektorem pohybu (MV).

Video

- MPEG-2
  - Na MPEG-2 může nahlížet jako na nadmnožinu MPEG-1.
  - MPEG-2 je zpětně kompatibilní s MPEG-1. Této žádoucí vlastnosti je dosaženo díky předchozímu obecnému požadavku na kódovací a dekódovací část v MPEG-1 standardu.
  - Schéma kódování a dekódování znázorněné na obrázku pro MPEG-1 je tedy stejné i pro MPEG-2.
  - MPEG-2 poskytuje větší flexibilitu co se týče možnosti ovlivňovat parametry.

Video

- MPEG-2: Hlavní rozdíly oproti MPEG-1

Vlastnost	MPEG-1	MPEG-2
Bit rate coding	max. 1,9 Mbps	max. 100 Mbps
Velikost snímku	288x352 @ 24 or 30 fps	288x352, 576x720, 1152x1440, 1152x1920
Prokládaný mód	jen progresivní	progresivní i prokládaný
Vzorkování intenzit barev	4:2:0	4:2:0, 4:2:2, 4:4:4
Profily a úrovně	-	5 profilů a každý do 4 úrovní

Video

- MPEG-4, 1993, ISO norma v roce 1999
  - Pozn. Standard MPEG-3 neexistuje, protože původně MPEG-3 měl kódovat HDTV, ale ukázalo se, že to MPEG-2 dokáže také.
  - MPEG-4 je skupina standardů pro ukládání video a audio dat.
  - Souběžně s vývojem MPEG-4 vznikal i standard komprese videa H.264.
  - Zastřešující nástroj (kontejner) pro řadu metod komprese video dat
  - Formát MPEG-4 mimo jiné podporuje:
    - Standardní snímky MPEG-1 a MPEG-2
    - Nepohyblivé obrázky JPEG a JPEG2000
    - Textové obrázky
    - Hybridní obrázky, tj. kompozice přirozených a synteticky vytvořených obrázků
    - 2D a 3D drátové objekty
    - Vysoká škálovatelnost jak z hlediska rozlišení tak hloubky

Video

- H.264
  - Norma H.264 byla navržena záměrně tak, aby se minimalizovala flexibilita ve prospěch lepšího kódovacího výkonu.
  - Rozšířena byla také šíře možných komunikačních kanálů a sítí.
- AVC
  - Na technologii H.264 byla postavena technologie Advanced Video Coding (AVC) , která se v roce 2003 stala standardem ve společném dokumentu H.264 a MPEG-4.
  - AVC standard má tři profily (módy)
    - Základní profil – videohovory, telekonference
    - Hlavní profil – podpora prokládaného videa, TV vysílání, archivace videa
    - Rozšířený profil – aplikace vyžadující online streaming
  - Uvedené schéma kodéru a dekodéru je použitelné i pro AVC vyjma možnosti přepínat mezi predikcí v rámci snímku a mezi snímky.

Video

- MPEG-4
  - Srovnání MPEG-4 a H.264

Comparison	MPEG-4 Visual	H.264
data types	rectangular fields and frames, video objects, still images, synthetic-natural hybrids, 2D and 3D mesh objects	rectangular fields and frames
profiles	19	3
compression efficiency	medium	high
motion compensation block size	8 x 8	4 x 4
motion vector	half or quarter pixel	quarter pixel
transform	8 x 8 DCT	4 x 4 DCT approximation
deblocking filter	no	yes

Video

- AVI (Audio Video Interleave), Microsoft, 1992
  - Podskupina RIFF (Resource Interchange File Format)
  - Dvě části: metadata a vlastní video data
  - Video data jsou kódována a dekódována pomocí tzv. kodeků (codec = coder/decoder) – lze tedy zahrnout zde zmíněné technologie Motion JPEG, MPEG-4.

- Omezení:
  - Nelze nastavit poměr stran zobrazení (aspect ratio).
  - Více soupeřících možností, jak zakódovat časovou osu.
  - Nepočítalo se s tím, že by algoritmus komprese využíval následující (budoucí) snímky za současným snímkem.
  - Neumožňuje proměnný bit rate.

Video

- VMW (Windows Media Video), Microsoft, (2003) 2006
  - Komprimovaný souborový videoformát pro několik kodeků vyvinutých společností Microsoft.
  - Původní kodek známý jako WMV byl navržen pro internetové streamingové aplikace.
  - WMV video je obvykle zapouzdřeno do kontejneru ASF (Advanced Systems Format) spol. Microsoft, ale může být také vložen do formátu AVI.

Literatura

- Pratt K. W., Introduction to Digital Image Processing, CRC Press, 2014
- Moeslund T. B., Introduction to Video and Image Processing: Building Real Systems and Applications, Springer, 2012
- Bartalmío M., Image Processing for Cinema, CRC Press, 2014
- Acharya T., Ray A. K., Image Processing: Principles and Applications, Wiley, 2005
- Sundararajan D., Digital Image Processing: A Signal Processing and Algorithmic Approach, Springer, 2017
- McAndrew A., Computational Introduction to Digital Image Processing, CRC Press, 2. vydání, 2016

Huffmanovo kódování

- Statistické rozdělení hodnot intenzit v obrázku není rovnoměrné, některé hodnoty jsou častější než jiné.
- To poukazuje na redundanci v reprezentaci hodnot, která může být definována jako celkový počet bitů aktuální číselné reprezentace mínus teoretický limit optimálního (nejefektivnějšího) kódování.
- Tuto redundanci nazýváme **entropie** kódovacího procesu.

Huffmanovo kódování

- Nejjednodušší statistické kódování spočívá v tom, že každé hodnotě intenzity pixelu přiřadíme kód na základě tabulky zvané kódová kniha.
- Pro efektivní kódování (co nejmenší celkový počet bitů reprezentace obrázku) je třeba, aby časté hodnoty měly co nejkratší (bitovou) reprezentaci a naopak.
- Průměrná délka kódu odpovídá entropii jednoho pixelu obrázku.
- Východiskem pro stanovení kódů je statistická analýza hodnot pixlů obrázku – pravděpodobnost výskytu dané hodnoty r_i intenzity pixelu v obrázku:

$$p_i = Pr\{F(j,k) = r_i\}$$

Huffmanovo kódování

- Průměrná délka kódu (average code length) je dána vztahem

$$Lc = \sum_{i=0}^{Q-1} p_i b_i$$

- kde b_i je počet bitů kódu kódujícího danou kvantizační úroveň (hodnotu jasu) a Q je počet kvantizačních úrovní (počet úrovní jasu).
- Entropie jednoho pixelu je dána vztahem

$$H = -\sum_{i=0}^{Q-1} p_i log_2(p_i)$$

Huffmanovo kódování

- Huffmanovo kódování spočívá v nalezení takových kódů pro dané hodnoty (kódované intenzity pixelů), aby celková bitová délka zakódovaného snímku (skupiny snímků) byla minimální.
- Huffmanovy kódy mají různou délku, což může způsobovat problémy během přenosu (proměnná rychlost přenosu jednotlivých hodnot).
- Pro snímky, které mají jiné statistické rozložení hodnot intenzit pixelů, než odpovídá předpokládanému výskytu, pro který byly kódy stanoveny, lze dostat výrazně neefektivní kódování.
- Pro typické šedotónové snímky se entropie jednoho pixelu pohybuje v rozmezí 5 až 7 bitů na pixel.

Huffmanovo kódování

- **Příklad**: Uvažujme obrázek, který má pouze 8 úrovní jasu. Statistickou analýzou zjistíme pravděpodobnosti výskytu jednotlivých hodnot intenzit pixelů ve snímku. V tabulce níže je vidět srovnání prostého PCM kódu (pulzní kódové modulace) a Huffmanova kódování.

Amplitude Index	Probability	PCM code	Huffman code
0	0.20	000	00
1	0.20	001	10
2	0.25	010	01
3	0.15	011	011
4	0.10	100	0111
5	0.05	101	01111
6	0.03	110	011111
7	0.02	111	111111

Huffmanovo kódování

- Sousední pixely v obrázku si jsou velmi podobné (korelované), což odpovídá vysoké redundanci pixelů např. v rámci jednoho řádku.
- Toho lze využít tím, že převedeme hodnoty intenzit pixelů na rozdíly sousedních hodnot (v rámci každého řádku) a kódujeme teprve tyto relativní hodnoty místo původních absolutních hodnot.
- Spíše se budou vyskytovat menší rozdíly v intenzitách sousedních pixelů než velké rozdíly a tedy prudké změny ve snímku.
- Díky tomu krátké kódy budou odpovídat malým rozdílům, kterých je hodně a naopak dlouhé kódy velkým rozdílům, kterých je ale málo.
- Další vylepšení může spočívat v tom, že pro rozdíly větší než definovaný práh zakódujeme přímo skutečnou hodnotu (těch nebude moc), viz obr.

Huffmanovo kódování

- V uvedené tabulce kódujeme malé rozdíly standardně.
- Hodnoty, které odpovídají velkým rozdílům kódujeme individuálně absolutní hodnotou intenzity pixelu s tím, že této hodnotě předchází rozlišovací prefix 0000.

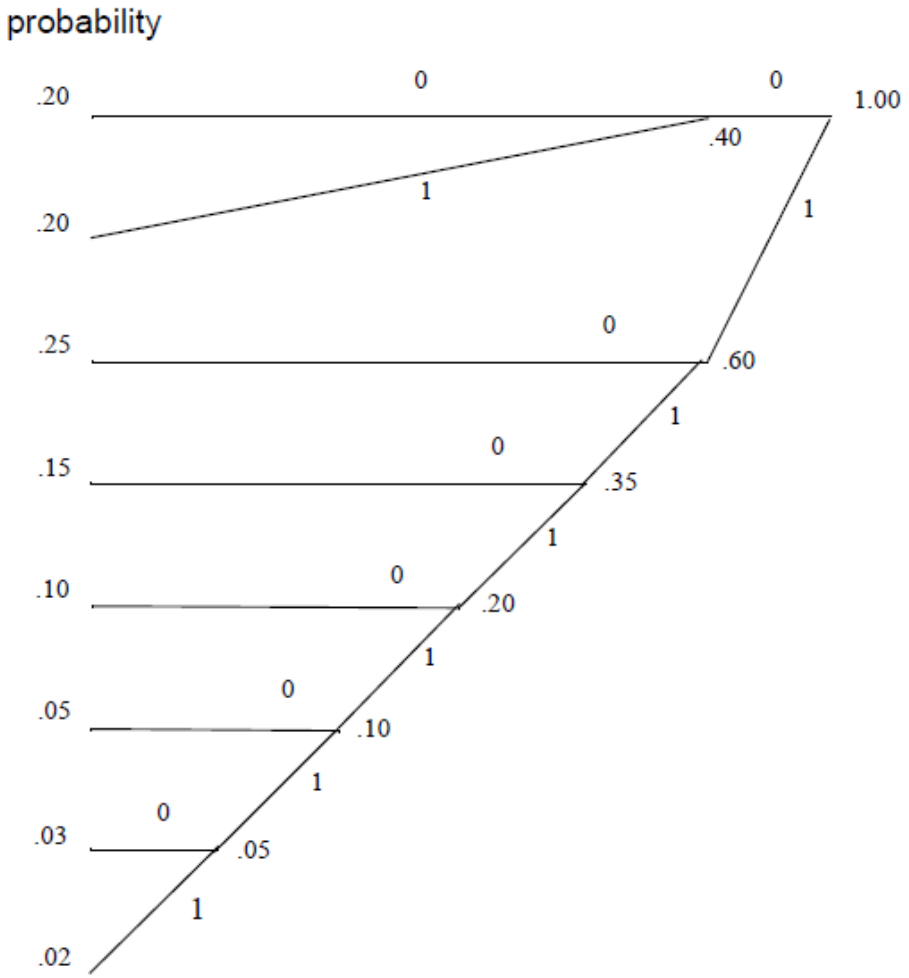
Difference, D	Code
0	1
+1	0100
-1	0101
+2	0110
-2	0111
+3	00100
-3	00101
+4	00110
-4	00111
$\|D\|$ =>5	0000 + 8 bit pixel

Huffmanovo kódování

- **Příklad**. Uvažujme snímek, který má Q=8 úrovní jasu.

1. Uvažujme Q úrovní jasu jako listy stromu a seřaďme je do klesající posloupnosti podle pravděpodobnosti jejich výskytu ve snímku.
2. Zkombinujme dva uzly s nejnižšími pravděpodobnosti do nového uzlu, jehož pravděpodobnost je součtem pravděpodobností jeho dvou větví.
3. Předchozí krok opakujeme pro zbývající Q-1 uzly.
4. Předchozí kroky opakujeme tak dlouho dokud nedostaneme jediný uzel – kořen stromu.
5. Označme všechny větve všech uzlů tak, že horní větvi přiřadíme 0 a dolní 1 (nebo naopak).
6. Spojíme kódy všech větví na cestě od kořene k listu. Získáme tak kód pro daný symbol.

Huffmanovo kódování



symbol	probability	code	length
1	0.200	00	2
2	0.200	10	2
3	0.250	01	2
4	0.150	011	3
5	0.100	0111	4
6	0.050	01111	5
7	0.030	011111	6
8	0.020	111111	6