

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 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**.
- Konceptně 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 Compensation 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 snímek
- 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

- 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 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 tzv. **3D unitární transformaci**, která **odstraní korelaci** mezi snímky a zefektivní kódování.

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

konvoluční jádra

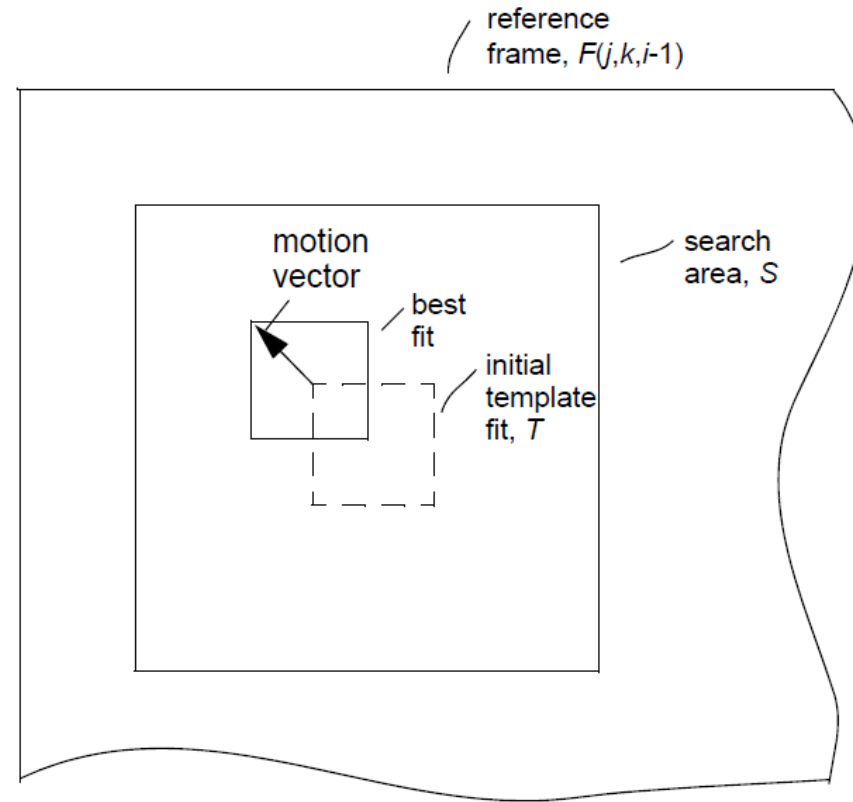
- Každý koeficient $\mathcal{F}(u, v, w)$ transformace je následně vhodně kódován.
- Inverzní transformací získáme rekonstruované bloky snímků $F(j, k, i)$.
- Nevýhodou této metody je potřeba úschovy velkého množství snímků.

Video

- Motion Compensation 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. **kompensace 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 různě posunutými oblastmi).
 - Velikost oblasti S určuje míru předpokládaného posunu objektu/kamery mezi dvěma snímky.

Video

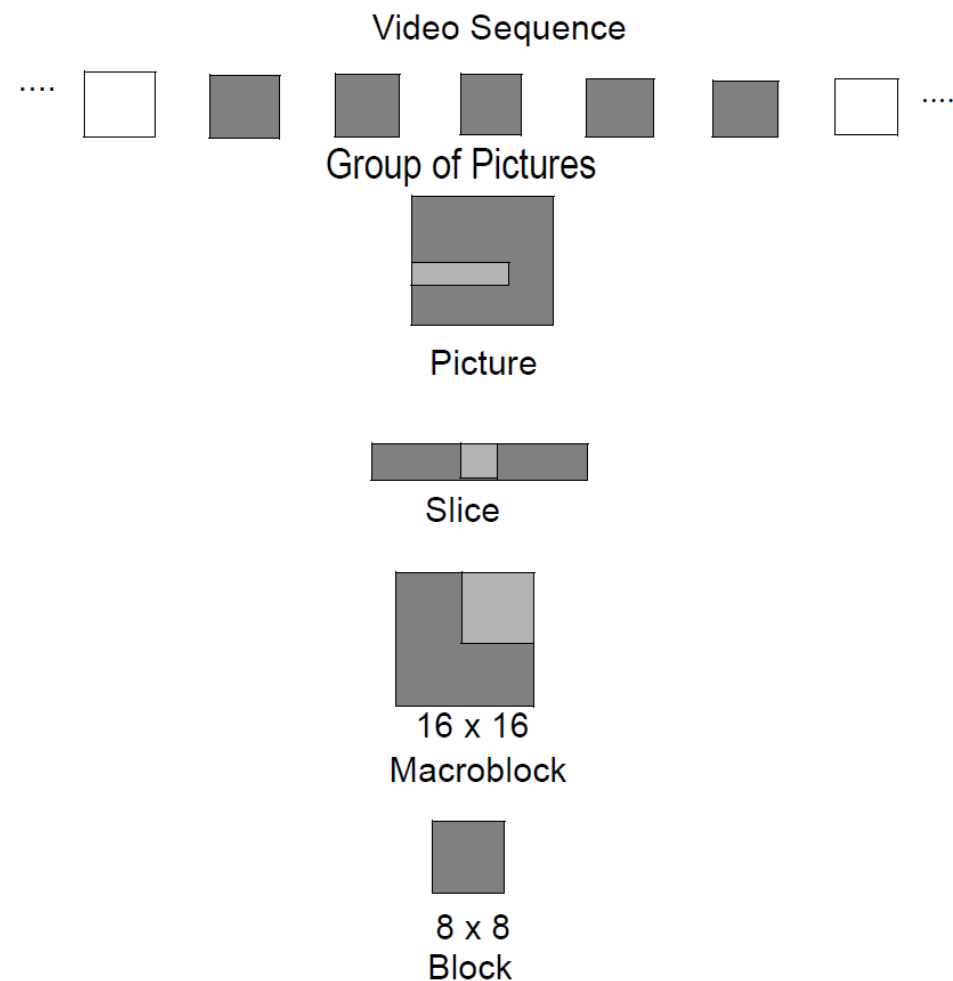
- Motion Compensation Interframe Coding, 1969



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 $C_r(j, k)$.

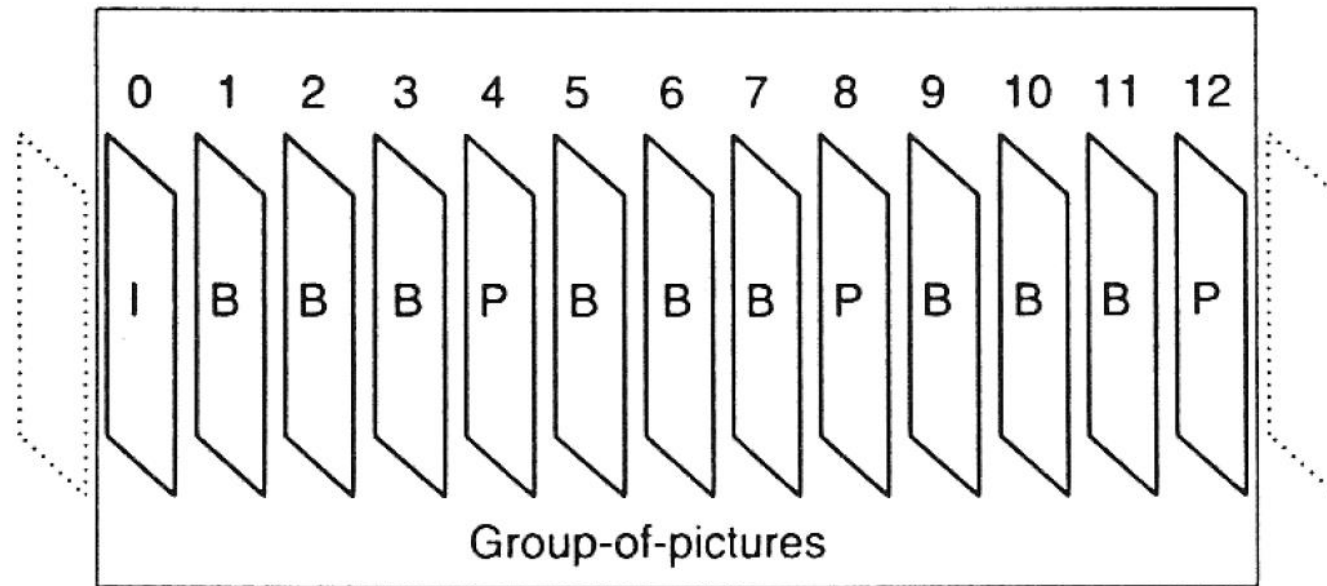


Video

- 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-snímekem nebo B-snímekem.



Video

- MPEG-1

- MPEG-1 snímky (pictures)

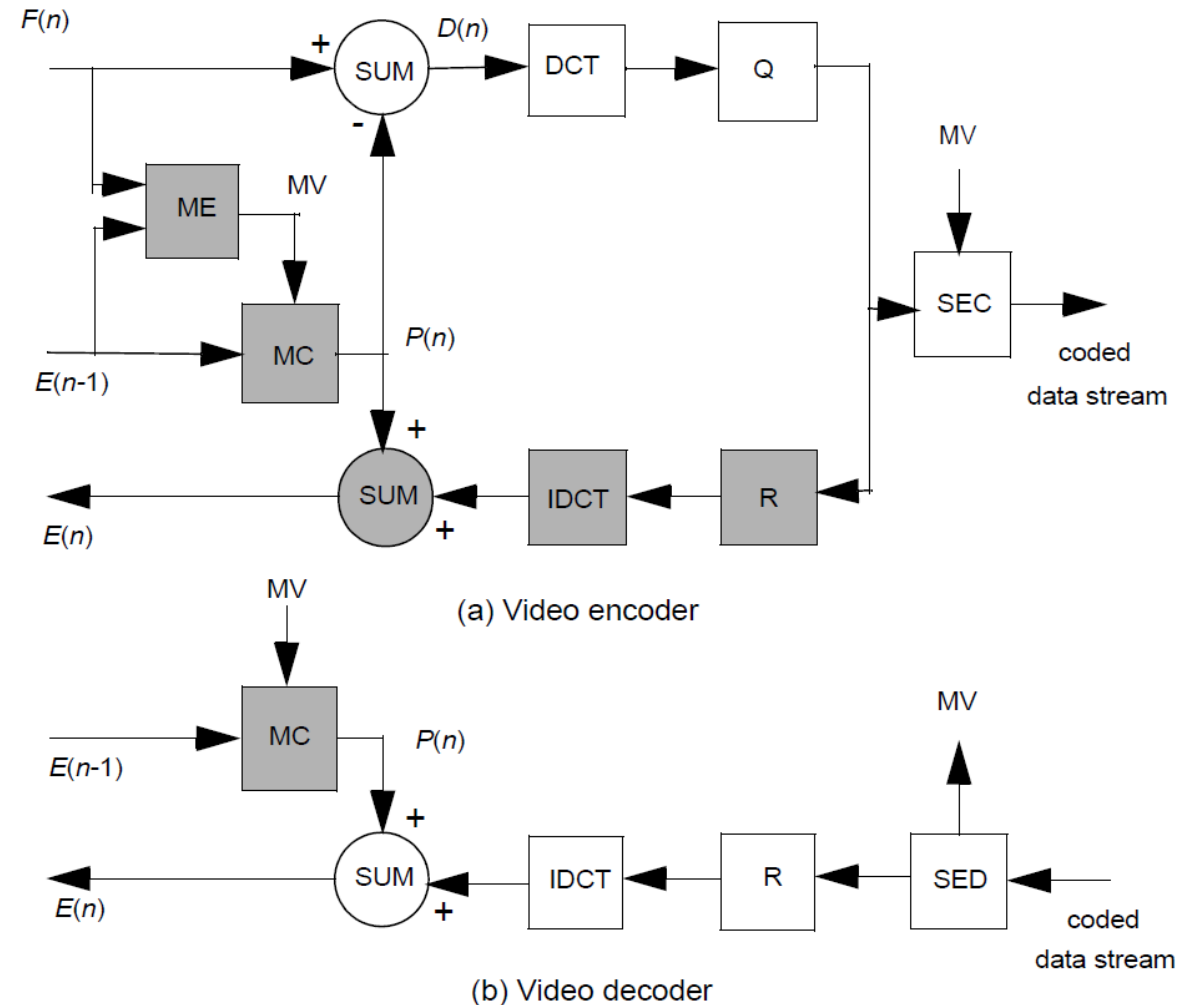
- **I-snímek** – 3 bloky velikosti 8x8 obsahující jas Y a intenzity barev C_b a C_r . Všechny reprezentují rozdílové snímek, který je kódován pomocí kosinové transformace. I-snímek může být použit pro predikci P-snímku a B-snímku.
 - **P-snímek** – 3 bloky velikosti 8x8 obsahující jas Y a intenzity barev C_b a C_r . Všechny reprezentují kompenzaci pohybu od I- nebo P-snímku. P-snímek může být použit pro predikci P-snímku a B-snímku.
 - **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

- 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 P- a B-snímků

- Snímek $F(n)$ je porovnán s předchozím snímkem $F(n-1)$.
- Porovnávají se jasové makrobloky těchto dvou snímků, které 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 blok intenzit barev získaného rozdílového makrobloku $D(n)$ je transformován 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. Těto žádoucí vlastnosti je dosaženo díky předchozímu obecnému požadavku na kódovací a dekodovací část v MPEG-1 standardu.
- Schéma kódování a dekodování 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 širě 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.

Segmentace a detekce pohybu ve videu

- 2 základní rozdíly mezi snímkem a videem
 - Video zahrnuje časovou složku informace (rozdíl mezi dvěma snímky v čase) – lze využít např. pro detekci pohybu a segmentaci pohyblivého objektu
 - Záznam videa a jednotlivého snímku může být provedeno různě
- Typické parametry videa
 - Frekvence snímání (framerate, fps)
 - **Kvalita snímání** (rozlišení, reprezentace informace – počet bitů na vyjádření hodnoty pixelu, ...) a rychlostí/kvalitou vyhodnocování v reálném čase (složitější algoritmy bývají přesnější, ale pomalejší)
 - **Komprese** (komprimované snímky jsou snazší na přenos, tj. lze mít větší frekvenci snímání, ale kvalita je většinou nižší) a kapacita přenosového pásma (bandwidth) [bits/s]

Segmentace a detekce pohybu ve videu

- Komprese
 - **Bezeztrátová** – např. RLE, kódování pomocí entropie apod.
 - **Ztrátová** – většinou postavena na vlastnostech/nedostatecích lidského vnímání obrazu
 - Takto komprimované video/obraz nemusí být ideální z hlediska strojového vidění/zpracování!
 - Např. lidské oko/mozek je citlivější na změny jasu než na změnu barvy. Proto je výhodné kódování barvy v systému YCbCr a barevné složky jsou komprimovány více (hruběji) než jasová složka.
- Při kompresi videa lze využít i skutečnosti, že dva po sobě jdoucí snímky nevykazují velké změny.
- Dokonce některé části nevykazují žádné změny -> rozdělení obrazu na bloky (blocks, templates) a hledání stejných (podobných) bloků mezi dvěma snímky a zaznamenání jen posunu těchto bloků
- Aplikace bezeztrátové komprese na redukovanou obrazovou informaci

Segmentace a detekce pohybu ve videu

- Komprese

- Každý snímek může dle svého charakteru vyžadovat jiný počet bitů na jeho zakódování.
- Může se proto stát, že narazíme na kapacitu přenosu (nedokážeme protlačit tolik informace za danou dobu)
- Proto musíme nějakou informaci zahodit, např. snížíme počet bitů na reprezentaci barvy.
- Tím ale dojde ke **ztrátě detailu** a tedy rozmazání/rozostření obrazu (blocking artifacts).
- Otázka kvalitnějšího (a dražšího) kamerového vybavení versus nutnost použití propracovanějších algoritmů pro zpracování obrazu.
- Kompromis: Levnější vybavení a následná detekce blocking artifacts a vypuštění takovýchto snímků ze zpracování.

Segmentace a detekce pohybu ve videu

- Rozmazané snímky
 - Pohybem (motion blur)
 - Nevhodnou hloubkou ostrosti (depth-of-field)
 - Špatné zaostření
- Rozmazané snímky se špatně vyhodnocují -> eliminovat rozmazání, je-li to možné, resp. detekce a odstranění rozmazaných snímků
- Detekce rozmazaných snímků – určení míry rozmazání
 - Na základě **analýzy hran** (ostré snímky -> více hran)
 - Na základě **porovnání snímku s** jeho cíleně **rozmazanou verzí** (původní rozmazaný snímek se příliš neliší od jeho dalšího rozmazání, ale u ostrých snímků je rozdíl oproti jejich rozmazané verzi výraznější)

Segmentace a detekce pohybu ve videu

- Detekce změn (pohybu) ve videu
 - 5 základních kroků
 - Výběr/vytvoření referenčního snímku
 - Záznam současného snímku
 - Vytvoření rozdílového snímku
 - Segmentace/prahování
 - Filtrace šumu
- Dva základní přístupy k vytvoření rozdílového snímku
 - **Statická scéna** (pozadí se nemění) -> odečtení pozadí (background subtraction). Snímek pozadí je přitom nasnímán předem a je stále stejný
 - **Dynamická scéna** (pozadí se mění, např. změna osvětlení) -> odečtení předchozího snímku, tj. difference dvou po sobě jdoucích snímků (image differencing)

Segmentace a detekce pohybu ve videu

- Rozdílový snímek
 - Pokud můžeme ovlivnit snímání, pak je vhodné zajistit pro statickou scénu takové pozadí, které je co nejkontrastnější vůči detekovanému objektu.
 - Rozdílové hodnoty dvou snímků mohou nabývat **záporných** hodnot -> pozor na reprezentaci rozdílu – může dojít ke ztrátě informace! Vhodné je využít např. absolutní hodnotu rozdílu.
- Segmentace rozdílového snímku
 - Cílem je nalézt objekt, který se pohybuje
 - Využívá se prahování rozdílového snímku, které převádí šedotónový snímek na binární
- Filtrace
 - Filtrujeme oprahovaný rozdílový binární snímek
 - Využívá se např. **mediánový filtr** nebo **morfologické operátory** (většinu uzavření)

Segmentace a detekce pohybu ve videu

- Odečtení pozadí (background subtraction)



1. Save reference image



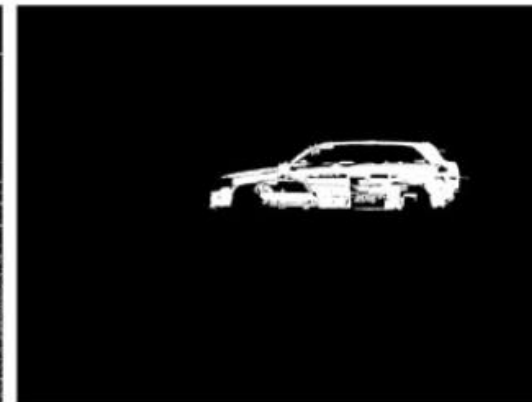
2. Capture current image



3. Perform image subtraction



4. Thresholding



5. Filter noise

Segmentace a detekce pohybu ve videu

- Odečtení pozadí (background subtraction)
 - Vhodné, pokud je pozadí uniformní, neměnné, dostatečně kontrastní
 - Využití hlavně pro snímání objektů v průmyslu (snadno definovatelné pozadí)
- Každý snímek ve videu ale zachytí mírně odlišné hodnoty pixelů pozadí (citlivost čipu, šum apod.). Snímky pozadí se proto mírně, ale přesto liší.
 - Proto je dobré za referenční snímek pozadí brát průměr prvních N snímků (během několik prvních sekund snímání).
- Pozadí se může měnit i vlivem drobných postupných změn během snímání, např. změnou denního světla. Díky tomu dochází k degradaci referenčního snímku a tedy i rozdílové snímky se začínají postupně zhoršovat.
 - Řešením může být **automatická aktualizace referenčního snímku** (paramet α se bývá $\alpha = 0.95$)

$$r_{\text{new}}(x, y) = \alpha \cdot r_{\text{old}}(x, y) + (1 - \alpha) \cdot f(x, y)$$

Segmentace a detekce pohybu ve videu

- Odečtení pozadí (background subtraction)
 - Metoda odečítání pozadí má své limity
 - Např. pokud snímáme v krajině, bývají přítomny pohyblivé objekty, takže je obtížné získat referenční snímek pozadí.
 - Možným řešením je stanovit referenční snímek pozadí tak, že nasnímáme více snímků a provedeme **mediánovou filtraci pro každý pixel** zvlášť. Tím se zbavíme pohyblivých objektů. Každý pohyblivý objekt se ale musí vyskytovat ve snímcích méně než v polovině případů.
 - Rozšířením předchozí metody stanovení statického pozadí je místo mediánu použít **průměr hodnot okolo mediánu** – větší robustnost

Segmentace a detekce pohybu ve videu

- Odečtení pozadí (background subtraction)
 - Dalším problémem může být **střídání pozadí**. Např. pokud fouká vítr, pohybují se listy, které jsou primárně součástí pozadí (nejsou pohyblivými objekty, které chceme detekovat) a ovlivňují tak referenční snímek pozadí. Podobně tomu může být s výskytem stínů, které se objevují a mizí.
 - Pokud bychom použili mediánovou filtraci každého pixelu, jak bylo popsáno výše, dostaneme pozadí, které je platné jen pro část snímků (s listem nebo bez něj).
 - Pokud bychom použili i rozšířenou verzi s průměrem sousedů mediánu, můžeme dostat pozadí, které je někde „na půl cesty“ od aktuální situace a opět si nepomůžeme.
 - Řešením je využít **více (dvou) referenčních snímků**: jeden pro situaci s výskytem listu a druhý bez. Uvažovat pak budeme ty hodnoty pixelů z jednoho nebo druhého snímku, které jsou blíže k aktuálnímu snímku.
 - <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.9209&rep=rep1&type=pdf>

Segmentace a detekce pohybu ve videu

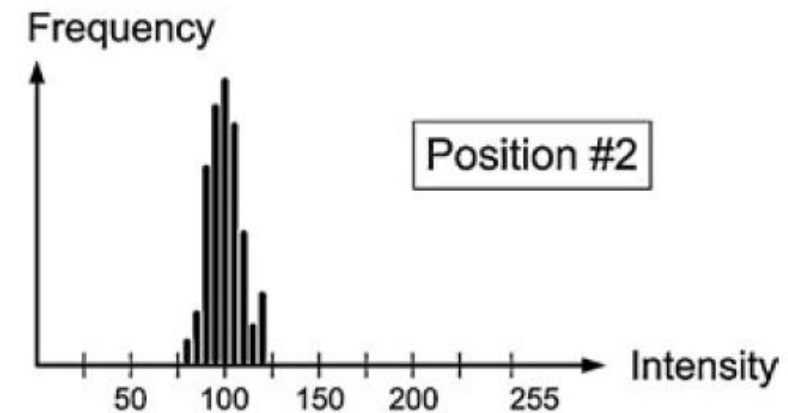
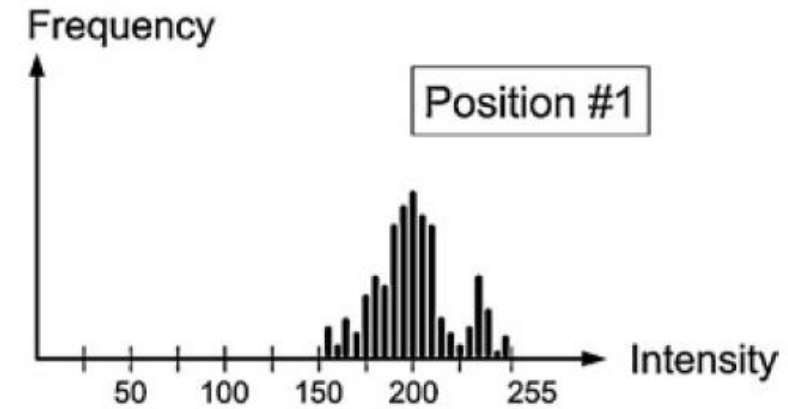
- Prahování

- Prahováním vysegmentováváme pohyblivý objekt (tu část scény, která se změnila)
- Rozdílový snímek odpovídá míře odchylky a ta je základem pro výpočet směrodatné odchylky/rozptylu.
- Protože zpracováváme video, počítáme řadu rozdílových snímků. I pro statickou scénu bez pohyblivých objektů, kterou opakovaně snímáme, dostáváme různé rozdílové hodnoty pro daný stále stejný pixel. Pro každý pixel scény můžeme sestavit histogram rozdílových hodnot.
- Pro ideální podmínky bychom měli dostávat stále stejnou hodnotu rozdílu hodnoty daného pixelu, ale díky různým vlivům (citlivost čipu, šum, drobné změny osvětlení a polohy apod.) dostáváme různé hodnoty rozdílu. Histogram tedy bude mít tvar podobný Gaussově funkci, viz následující obrázek.

Segmentace a detekce pohybu ve videu

- Prahování

-



Segmentace a detekce pohybu ve videu

- **Prahování**

- Pokud používáme globální práh na rozdílový snímek, předpokládáme, že histogramy odpovídající jednotlivým pixelům jsou stejné.
- V praxi tomu ale tak nemusí být. V některých částech snímku dochází k většímu rozptylu hodnot a proto i histogramy těchto oblastí budou širší (budou mít větší rozptyl). Globální práh tak nemusí fungovat dobře.
- Řešením může být stanovení a následné využití **lokálních prahů** pro každý pixel obrázku. Těchto lokálních prahů bude tolik, kolik je pixelů v obrázku.
- Hodnoty těchto lokálních prahů nastavíme algoritmicky na základě statistických výpočtů směrodatných odchylek/rozptylů zvoleného počtu po sobě jdoucích snímků.

$$\text{Binary image} = \begin{cases} 0, & \text{if } \text{Abs}(g(x, y)) < \beta \cdot \sigma(x, y); \\ 255, & \text{otherwise} \end{cases}$$

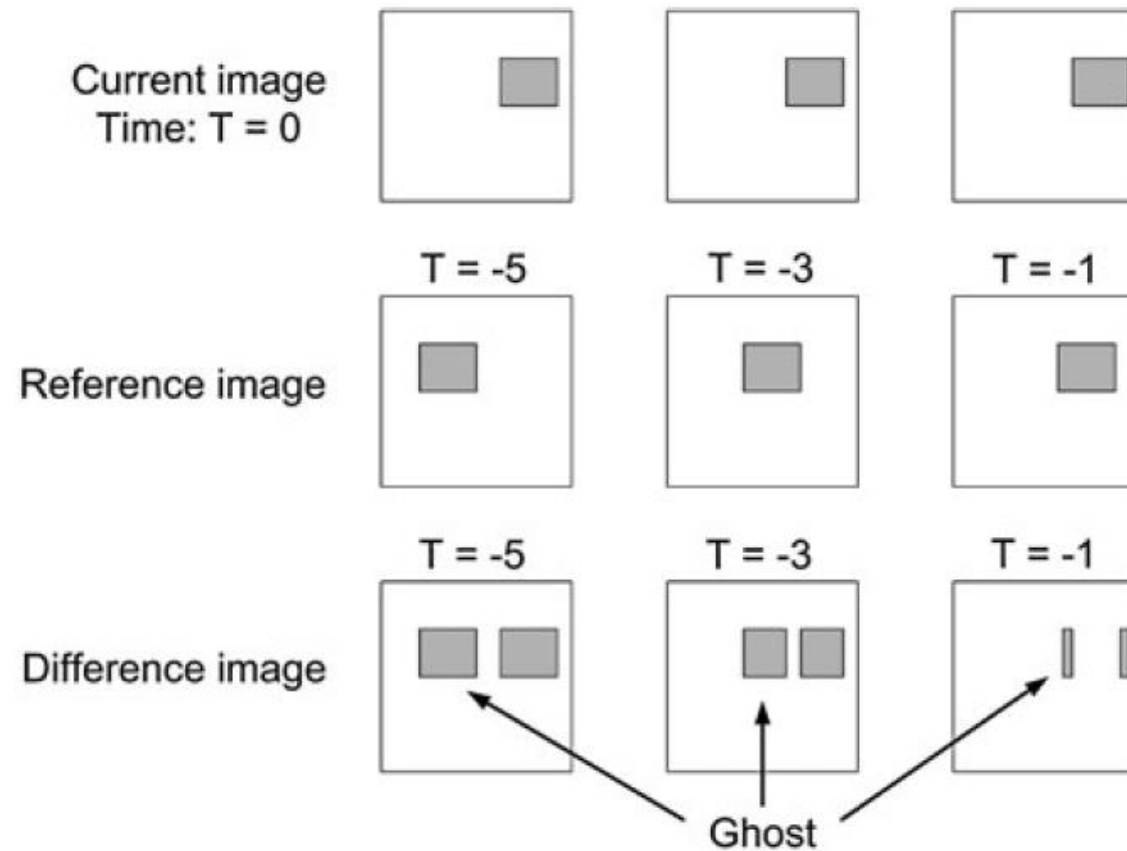
- Kde β je parametr a $\sigma(x, y)$ je matice standardních odchylek pro každou pozici pixelu.

Segmentace a detekce pohybu ve videu

- Diferenční snímky (image differencing)
 - Pokud se pozadí významně mění, není možné použít konstantní referenční snímek pro výpočet rozdílu.
 - Za referenční snímek se bere předchozí snímek (nebo jiný nepříliš časově vzdálený snímek od toho současného, aby byl zřetelný výraznější rozdíl). Referenční snímek není už statický, ale mění se podle situace.
 - 2 problémy:
 - Pokud se objekt zastaví, pak není detekovatelný
 - Výskyt „**duchů**“ (ghost objects) – objektů, které nejsou detekovanými objekty (zdvojení objektů)
- Odstranění „duchů“
 - Pokud známe rychlost a směr, lze dovodit, který objekt je pravý a který ne.
 - Skutečný objekt má jiné hodnoty v rozdílovém snímku (např. kladné, pokud je světlejší než pozadí), než má „duch“ (který je pak má záporné).

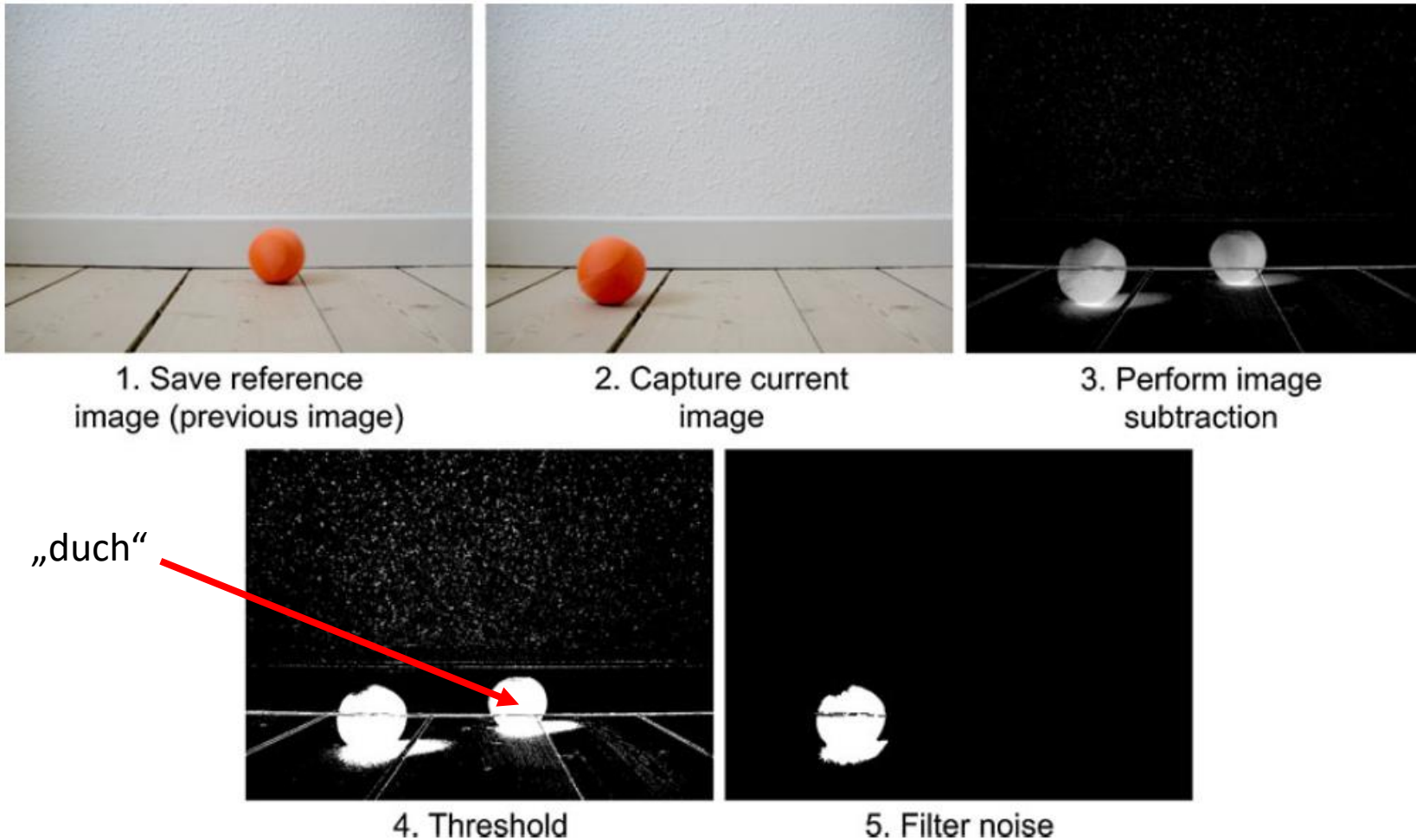
Segmentace a detekce pohybu ve videu

- Diferenční snímky (image differencing)



Segmentace a detekce pohybu ve videu

- Diferenční snímky (image differencing)



Segmentace a detekce pohybu ve videu

- Diferenční snímky (image differencing)
 - Pokud je rychlost snímání velká a pohyb objektu malý, může dojít k tomu, že objekt se částečně bude **překrývat** se svým duchem.
 - To způsobí, že pouze část objektu je detekovatelná.
 - Možným řešením je určit, který z předchozích snímků dává rozdílový snímek, kde nedochází k překryvu a za referenční snímek brát vždy ten, který je takto dostatečně vzdálený od současného snímku.

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