



Segmentace III

Pokročilá analýza obrazu

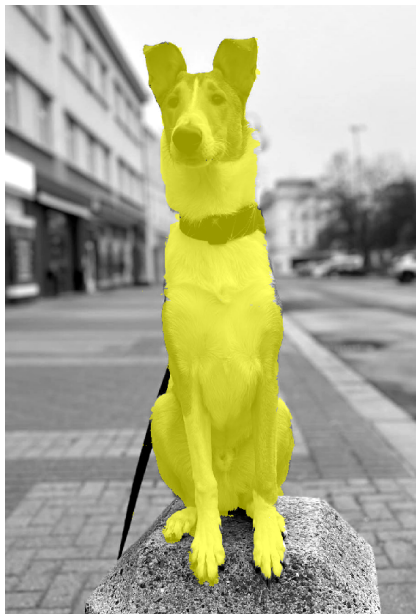
Mgr. Markéta Trnečková, Ph.D.

Obraz

- Segmentace (dělení obrazu na nějaké logické části)
 - dělení obrazu dle nějakých prudkých změn
 - dělení obrazu na regiony, které mají společné vlastnosti
 - jejich kombinace

Segmentace

- R – celý region (obraz)
- **Segmentace** – rozdělení obrazu na jednotlivé regiony R_1, \dots, R_n :
 - $\bigcup_{i=1}^n R_i = R$
 - R_i jsou spojité
 - $R_i \cap R_j = \emptyset$, pro všechna $i \neq j$
 - $Q(R_i) = TRUE$ – všechny pixely v oblasti splňují nějakou podmínku
 - sousední regiony stejnou podmínku nesplňují
 $Q(R_i \cup R_j) = FALSE$
- dělení obrazu na části, které splňují nějakou podmínku:
 - pixely v oblasti jsou „shodné“
 - pixely v sousedních oblastech jsou „rozdílné“



Segmentace

- Doposud jsme využívali princip:
 - detekce hran
 - prahování
 - extrakci oblastí
- U každého z těchto přístupů jsme zjistili, že má své výhody i nevýhody

Příklad

Jaké mají dříve zmíněné metody výhody a nevýhody?

Segmentace

Příklad

Jaké mají dříve zmíněné metody výhody a nevýhody?

■ Příklad výhod:

- Rychlost výpočtu u globálního prahování

■ Příklad nevýhod:

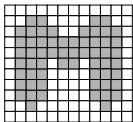
- Potřeba spojování bodů při detekci hran
- Výpočetní náročnost pokročilých metod

■ Segmentace pomocí morfologických watershedů

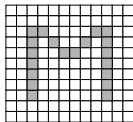
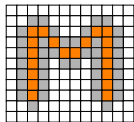
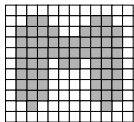
- zahrnuje mnohé myšlenky z předchozích tří přístupů
- poskytuje stabilnější výsledky segmentace
- možnost zahrnout do segmentace omezení, která jsou založená na znalosti obrazu

Matematická morfologie

- Známe z kurzu KMA/OBR, nepracujeme přímo s intenzitami, ale s geometrií objektů v obraze **Morfologie** poskytuje **nástroje založené na teorii množin**, které pracují přímo s tvary (množinami bodů)
- **dva typy množin** (v matematické morfologii):
 - objekty – pixely popředí a pozadí



- strukturní element (SE) – pixely popředí, pozadí a „don't care“

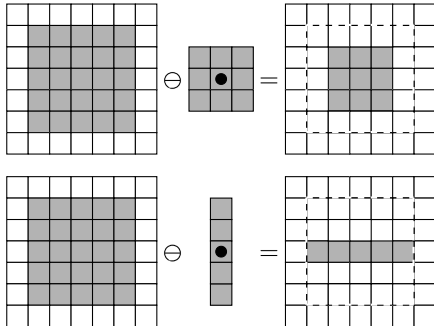


Základní morfologické operace

- SE posouváme po obrázku obdobně, jako v konvoluci
- provádíme množinové operace s pixely popředí SE (B) a pixely popředí obrázku (A)
- základní operace
 - eroze
 - dilatace
- jejich kombinace tvoří složitější operace
- výsledky operací vždy záleží na tvaru a velikosti SE

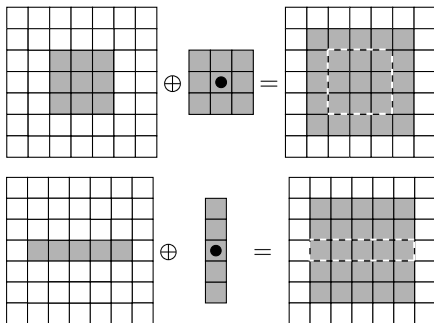
Eroze

- zmenšuje objekty v obraze, „ožírá“ jejich okraje
- $A \ominus B = \{z \mid (B)_z \subseteq A\}$
- bod z patří do výsledku, pokud je **celý strukturální element** B při posunutí do bodu z **obsažen uvnitř objektu** A
- **Výsledek:**
 - zmenšuje objekty
 - odstraňuje malé části a zužuje tenké spoje



Dilatace

- rozšiřuje objekty v obraze, „přidává“ pixely k jejich hranám
- $A \oplus B = \{z \in E \mid (\hat{B})_z \cap A \neq \emptyset\}$
- bod z patří do výsledku, pokud se **překrývá alespoň jeden bod reflektovaného strukturálního elementu \hat{B}** posunutého do bodu z s objektem A
Často se volí symetrický SE. V takovém případě reflexe není nutná.
- **Výsledek:**
 - rozšiřuje objekty
 - vyplňuje malé díry a spojuje blízké objekty

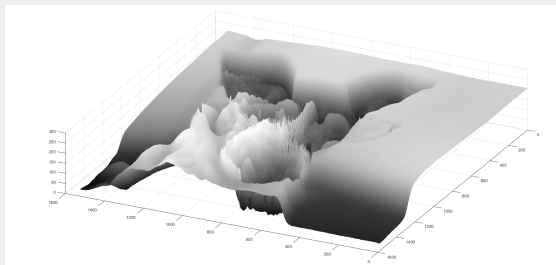


Segmentace založená na morfologických watershadech

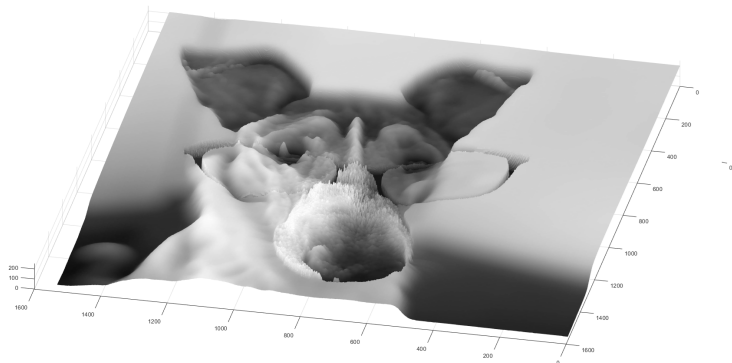
- Každý pixel je reprezentován 3 rozměrným vektorem (2 prostorové souřadnice a intenzita)

Příklad

Jaký obrázek je reprezentován tímto způsobem níže?



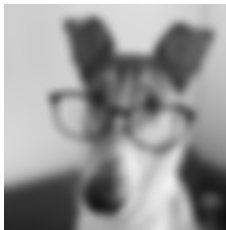
Segmentace založená na morfologických watershadech



Segmentace založená na morfologických watershedech

- Každý pixel je reprezentován 3 rozměrným vektorem (2 prostorové souřadnice a intenzita)
- Typy bodů:
 - 1 body patřící do regionálního minima
 - 2 body, v nichž by kapka vody umístěná do libovolného z těchto bodů s jistotou stekla do jednoho konkrétního minima
 - 3 body, z nichž by voda se stejnou pravděpodobností stekla do více takových minim
- Pro dané regionální minimum se množina bodů (2) nazývá **povodí** nebo watershed tohoto minima
- Body (3) vytvářejí na topografickém povrchu hřebenové linie a nazývají se **dělicí linie** nebo **watershed lines**
- Segmentační algoritmy hledají watershed linie

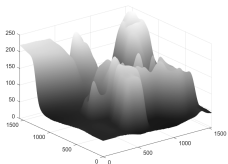
Segmentace založená na morfologických watershadech



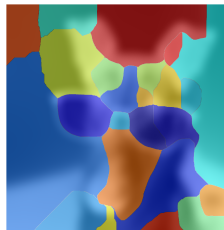
Vyhlazený obrázek



Komplement



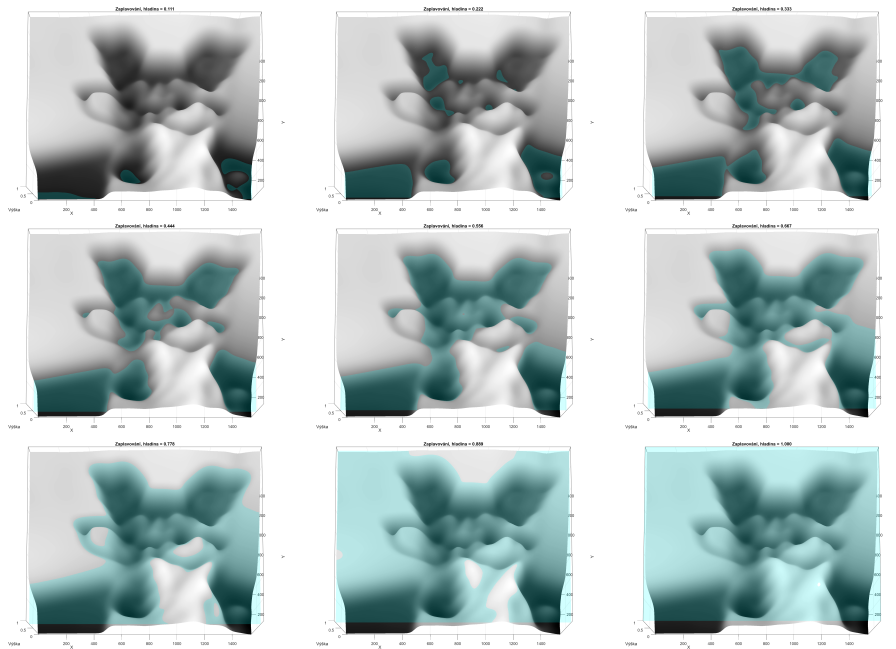
Povrchová reprezentace



Watershed

Každou oblast zespodu zaplavujeme. Jakmile by se dvě oblasti měly slít dohromady, tak zde vytvoříme hráz (watershed). Takto pokračujeme, dokud jsme nezaplavili, celou oblast.

Segmentace založená na morfologických watershadech



Segmentace založená na morfologických watershedech

- Watershed linie tvoří souvislé cesty = vytvářejí spojité hranice mezi regiony
- Jednou z hlavních aplikací watershed segmentace je extrakce téměř homogenních (blob-like) objektů z pozadí
- Oblasti charakterizované malými změnami intenzity mají malé hodnoty gradientu
- Proto se v praxi watershed segmentace často aplikuje na gradient obrazu, nikoli přímo na samotný obraz
- K hledání hrází využíváme matematickou morfologii

Segmentace založená na morfologických watershedech

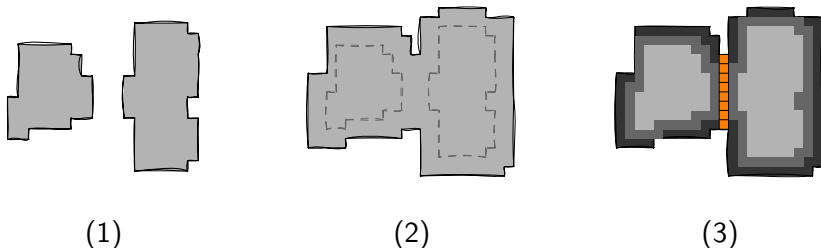
Hledání hrází

- Watershed linie tvoří souvislé cesty = vytvářejí spojité hranice mezi regiony
- Jednou z hlavních aplikací watershed segmentace je extrakce téměř homogenních (blob-like) objektů z pozadí
- Oblasti charakterizované malými změnami intenzity mají malé hodnoty gradientu
- Proto se v praxi watershed segmentace často aplikuje na gradient obrazu, nikoli přímo na samotný obraz
- K hledání hrází využíváme matematickou morfologii

Segmentace založená na morfologických watershadech

Hledání hrází

- Hráze konstruujeme na binárních obrázcích a využíváme k tomu dilataci (3×3)



- 1 Části dvou povodí ve fázi zaplavování $n - 1$
- 2 Výsledek v následujícím kroku zaplavování n
- 3 V kroku n voda přetekla z jednoho povodí do druhého, a proto musí být vybudována hráz, která tomu zabrání

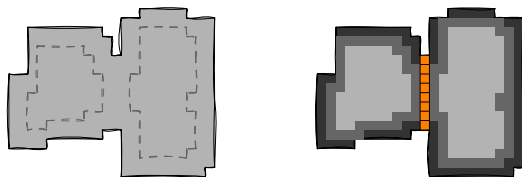
Segmentace založená na morfologických watershadech

Hledání hrází

- Označme M_1 a M_2 množiny souřadnic bodů ve dvou regionálních minimech
- $C_{n-1}(M_1)$, $C_{n-1}(M_2)$ = množina souřadnic bodů v povodí odpovídajícím těmto dvěma minimům ve fázi zaplavování $n - 1$
- Označme $C[n - 1]$ sjednocení těchto dvou množin
- $C[n - 1]$ je tvořena dvěma souvislými komponentami
- V další fázi je oblast tvořena pouze jednou souvislou komponentu = došlo ke slití
- Označme tuto souvislou komponentu q .
- Dvě komponenty z kroku $n - 1$ lze z q získat provedením logické operace AND:
 $q \cap C[n - 1]$
- Oblasti M_i postupně dilatujeme za splnění dvou podmínek:
 - 1 Dilatace musí být omezena na množinu q (tj. střed SE může být během dilatace umístěn pouze v bodech množiny q)
 - 2 Dilatace nesmí být provedena na bodech, které by způsobily sloučení dilatovaných množin (tj. vznik jedné souvislé komponenty)

Segmentace založená na morfologických watershadech

Hledání hrází

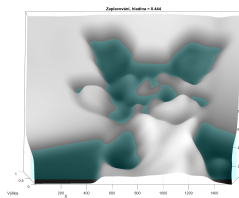
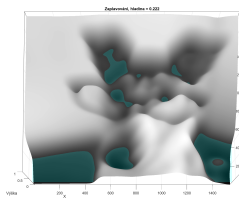


- První provedení dilatace (světlejší šedá) rozšířilo obě oblasti za splnění podmínky (1) a přitom nedošlo k porušení (2)
- Druhá dilatace (tmavší šedá) už nerozšířila celou oblast pravidelně, navíc některé body porušili podmínku (2)
- Body, které splňují obě podmínky tvoří jednopixelovou souvislou cestu (oranžová část)
- Tato cesta představuje hledanou hráz
- V této fázi zaplavování konstrukci hráze ukončíme tak, že všem jejím bodům nastavíme maximální výšku v obraze
- Tím zamezíme tomu, že by v dalších fázích tato hráz přetekla

Segmentace založená na morfologických watershadech

Segmentace

- Necht M_1, M_2, \dots, M_R jsou množiny označující souřadnice bodů v regionálních minimech obrazu $g(x, y)$ (často obraz gradientu)
min a max minimální a maximální hodnoty funkce $g(x, y)$
- Necht $C(M_i)$ označuje množinu souřadnic bodů v povodí příslušném k regionálnímu minimu M_i
- $T[n]$ je množina souřadnic (s, t) , pro které platí $g(s, t) < n$, tedy
 $T[n] = \{(s, t) \mid g(s, t) < n\}$
Geometricky je $T[n]$ množina souřadnic bodů, které leží pod rovinou $(g(x, y) = n$
- Zaplavujeme po celočíselných krocích, od $(n = \min + 1$ do $n = \max + 1$
- $T[n]$ můžeme chápat jako binární obrázky (1 pixely splňující $g(s, t) < n$, 0 jinde)



Segmentace založená na morfologických watershadech

Segmentace

- $C_n(M_i)$ označuje množinu souřadnic bodů v povodí příslušném k minimu M_i , které jsou zaplaveny ve fázi n
$$C_n(M_i) = C(M_i) \cap T[n]$$
- B označuje počet zaplavených povodí ve fázi n a necht' $C[n]$ označuje sjednocení těchto povodí
$$C[n] = \bigcup_{i=1}^B C_n(M_i)$$
- Je zřejmé, že
 - $C[\max + 1]$ je sjednocení všech povodí
 - $C[n - 1] \subseteq C[n]$
 - $C[n] \subseteq T[n]$
 - $C[n]$ je obsažena právě v jedné spojitě komponentě $T[n]$

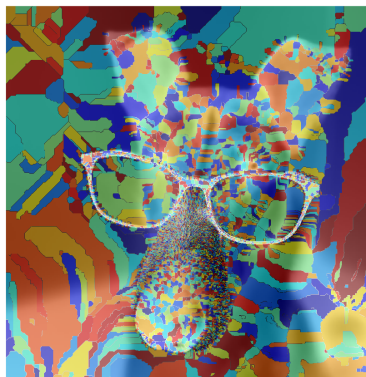
Segmentace založená na morfologických watershadech

Algoritmus

- Nastavíme $C[\min + 1] = T[\min + 1]$
- Rekurzivně počítáme $C[n]$ z $C[n - 1]$:
 - $Q[n]$ označuje množinu spojitých komponent v $T[n]$
 - Pro každou $g \in Q[n]$ existují 3 možnosti:
 - 1 $g \cap C[n - 1]$ je prázdná množina
Nalezeno nové minimum, g je přidána do $C[n - 1]$, čímž vznikne $C[n]$
 - 2 $g \cap C[n - 1]$ obsahuje jednu souvislou komponentu množiny $C[n - 1]$
 g je uvnitř nějakého povodí (jednoho), g je přidána do $C[n - 1]$, čímž vznikne $C[n]$
 - 3 $g \cap C[n - 1]$ obsahuje více než jednu souvislou komponentu množiny $C[n - 1]$
Dosáhli jsme hřebene, který odděluje povodí, zaplavení by způsobilo slití, vytvoříme v g hráz (pomocí dilatace, viz dříve)
- Efektivita algoritmu se dá zvýšit tak, že budeme za n brát pouze hodnoty obsažené v obraze

Použití markerů

- Přímá aplikace watershed segmentace vede k **přesegmentování** (over-segmentation)
- To je způsobeno zejména šumem a lokálními nepravidelnostmi
- Výsledek algoritmu je prakticky nepoužitelný



Použití markerů

- Omezíme počet povolených oblastí pomocí předzpracování (to do segmentace přidává další znalosti)
- Jedním z přístupů je použití **markerů** (souvislí komponenta patřící obrazu)
 - **vnitřní markery** (internal markers), spojené s objekty zájmu
 - **vnější markery** (external markers), spojené s pozadím
- Hledání markerů:
 - Předzpracujeme obraz
 - Definujeme kritéria, která musí markery splňovat

Příklad

Napadá vás, jak bychom mohli obraz předzpracovat a proč?

Použití markerů

- Např. malé detaily odstraníme vyhlazením
- Velice se také hodí metody matematické morfologie
- Výběr markerů může být založen na intenzitách, na vzdálenosti, textuře a dalších
Později se budeme bavit o různých features
- Do procesu segmentace tak díky markerům přidáváme další informaci (kontext)

Použití markerů

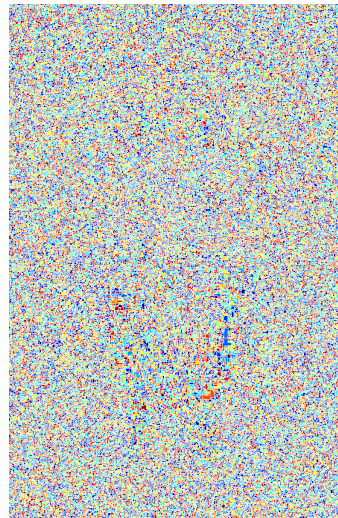
Konkrétní příklad



Originál



Gradient

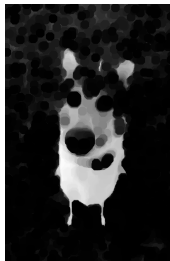


Watershed

Použití markerů

Konkrétní příklad

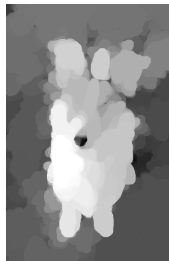
- Aby se odstranily malé lokální nerovnosti a potlačil šum, použije se morfologické předzpracování: eroze, otevření rekonstrukcí a uzávěr rekonstrukcí
- Smyslem tohoto kroku je vyhladit obraz tak, aby v něm zůstaly pouze významné struktury objektů a pozadí.



Eroze



Otevření rekonstrukcí



Dilatace



Uzavření rekonstrukcí

Použití markerů

Konkrétní příklad

- Z takto vyhlazeného obrazu se určí *foreground markers*, tedy značky odpovídající objektům zájmu
- Tyto markery se mnohdy také čistí pomocí morfologických operací



Markery



Markery



Po úpravě



Po úpravě

Použití markerů

Konkrétní příklad

- Nyní najdeme background markery
- Binarizujeme vyhlazený obraz a na něm spočítáme distanční transformace
- Ta každému bodu přiřazuje vzdálenost od nejbližšího objektu
- Na výsledný obrázek aplikujeme watershed – rozhraní mezi oblastmi pozadí a použijeme je jako background markery



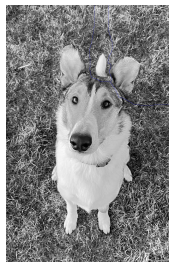
BW obraz



Distanční transformace



Watershed



Background markery

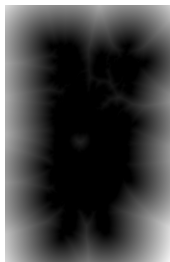
Použití markerů

Konkrétní příklad

- Nyní najdeme background markery
- Binarizujeme vyhlazený obraz a na něm spočítáme distanční transformace
- Ta každému bodu přiřazuje vzdálenost od nejbližšího objektu
- Na výsledný obrázek aplikujeme watershed – rozhraní mezi oblastmi pozadí a použijeme je jako background markery



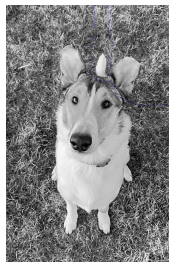
BW obraz



Distanční transformace



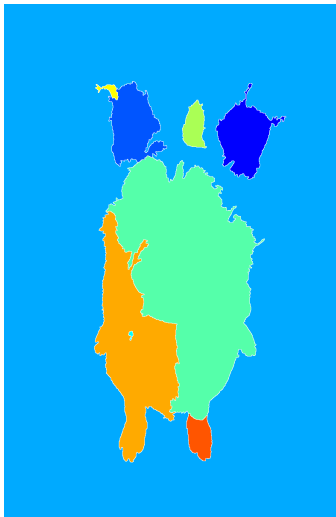
Watershed



Background markery

Použití markerů

Konkrétní příklad



Watershed



Hranice



Objekt

Využití pohybu při segmentaci

- Pohyb je silným vodítkem, které lidé i mnoho živočichů využívají k extrakci objektů nebo oblastí zájmu z pozadí obsahujícího nerelevantní detaily
- V obrazových aplikacích vzniká pohyb v důsledku relativního posunu mezi snímacím systémem a pozorovanou scénou
- Například v robotických aplikacích, autonomní navigaci nebo při analýze dynamických scén
- Ukážeme si využití pohybu při segmentaci:
 - v prostorové oblasti
 - ve frekvenční doméně

Využití pohybu při segmentaci

Prostorová doména

- Detekce změn mezi dvěma snímky $f(x, y, t_i)$ a $f(x, y, t_j)$, pořízenými v časech t_i a t_j – jejich porovnání pixel po pixelu

- Vytvoříme **rozdílový obraz**

$$d_{ij}(x, y) = \begin{cases} 1, & \text{jestliže } |f(x, y, t_i) - f(x, y, t_j)| > T, \\ 0, & \text{jinak,} \end{cases}$$

T je nezáporný práh

- Pokud bychom měli referenční obraz (jen statické pozadí), pak by rozdílový obraz byl maskou, kde je pohyblivý objekt
- Správně to funguje jen v případě, že jsou obrazy zarovnány a nemění se osvětlení (je relativně konstantní úměrně prahové hodnotě)
- V praxi se objevují odlehlé body, které vznikly vlivem šumu
- Toho se můžeme vyvarovat, pokud odstraníme malé souvislé komponenty
- Díky tomu, že předpokládáme, že je pozadí konstantní (neměnné), využívá se tato metoda jen v kontrolovaných prostředích

Využití pohybu při segmentaci

Prostorová doména

- Uvažujme posloupnost obrazových snímků $f(x, y, t_1), f(x, y, t_2), \dots, f(x, y, t_n)$
 $f(x, y, t_1)$ je referenční obraz (dále značím $R(x, y)$)
- **Kumulativní rozdílový obraz** (ADI – Accumulative Difference Image) vznikne tak, že tento referenční obraz porovnáme se všemi následujícími obrazy v posloupnosti
- Pro každou polohu pixelu v kumulativním obraze udržujeme čítač, který zvýšíme pokaždé, když se na dané pozici objeví rozdíl mezi obrazy

- Typy kumulativních obrazů:

- Absolutní ADI

$$A_k(x, y) = \begin{cases} A_{k-1}(x, y) + 1, & \text{jestliže } |R(x, y) - f(x, y, t_k)| > T, \\ A_{k-1}(x, y), & \text{jinak,} \end{cases}$$

- Pozitivní ADI

$$P_k(x, y) = \begin{cases} P_{k-1}(x, y) + 1, & \text{jestliže } R(x, y) - f(x, y, t_k) > T, \\ P_{k-1}(x, y), & \text{jinak,} \end{cases}$$

- negativní ADI

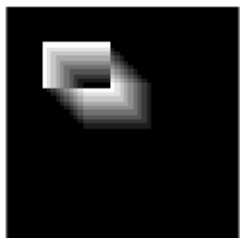
$$N_k(x, y) = \begin{cases} N_{k-1}(x, y) + 1, & \text{jestliže } R(x, y) - f(x, y, t_k) < -T, \\ N_{k-1}(x, y), & \text{jinak.} \end{cases}$$

- V předchozím předpokládáme, že intenzity pohybujících se objektů jsou větší, než pozadí

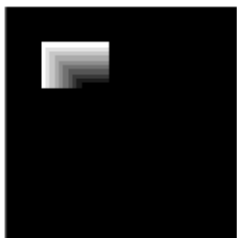
Využití pohybu při segmentaci

Prostorová doména

■ Pohyb bílého obdélníku



Absolutní ADI



Pozitivní ADI



Negativní ADI

Využití pohybu při segmentaci

Prostorová doména

Sekvence



Absolutní ADI



Pozitivní ADI



Negativní ADI

Využití pohybu při segmentaci

Prostorová doména

- Klíčem k úspěchu je nalezení referenčního obrázku vůči kterému porovnáváme následující obrázky
- Rozdíl mezi dvěma obrazy v dynamickém zobrazovacím problému má tendenci eliminovat všechny stacionární složky, takže zůstávají pouze obrazové prvky odpovídající šumu a pohybujícím se objektům
- Získání referenčního obrazu obsahujícího pouze stacionární prvky však není vždy možné
- Je nutné vytvořit referenční obraz z množiny snímků, které obsahují jeden nebo více pohybujících se objektů
- **Možný postup:**
 - První obraz v sekvenci se považuje za referenční obraz
 - Jakmile se nestacionární komponenta zcela přesune ze své původní polohy v referenčním snímku, lze odpovídající pozadí z aktuálního snímku zkopírovat do místa, které bylo v referenčním obraze původně obsazeno objektem
 - Jakmile se všechny pohybující se objekty zcela přesunou ze svých původních poloh, vznikne referenční obraz obsahující pouze stacionární komponenty
 - Posun objektů lze určit sledováním změn v kladném kumulativním rozdílovém obraze (positive ADI)

Využití pohybu při segmentaci

Frekvenční doména

- Určování pohybu pomocí formulace založené na Fourierově transformaci
- Na začátku předpokládejme, že všechny snímky mají homogenní pozadí s nulovou intenzitou
- Výjimkou je jediný objekt o velikosti jednoho pixelu s jednotkovou intenzitou, který se pohybuje konstantní rychlostí
- V prvním snímku ($t = 0$) se objekt nachází v poloze (x', y')
- Rovina obrazu je nyní promítnuta na osu x , tj. hodnoty intenzity pixelů se pro každý řádek sčítají přes všechny sloupce
- Vznikne jednorozměrné pole o M prvcích, které mají nulovou hodnotu všude kromě pozice x' (souřadnice objektu)
- Jestliže nyní vynásobíme všechny prvky tohoto pole výrazem

$$e^{[j2\pi a_1 x \Delta t]}$$

$x = 0, 1, 2, \dots, M - 1$ a výsledky sečteme, dostaneme

$$e^{[j2\pi a_1 x' \Delta t]}$$

Δt časový interval mezi snímky, a_1 kladné celé číslo

Využití pohybu při segmentaci

Frekvenční doména

- Pokud se objekt bude nadále posouvat o jeden pixel v každém snímku, pak v libovolném čase t dostaneme

$$e^{j2\pi a_1(x'+t)\Delta t} = \cos(2\pi a_1(x' + t)\Delta t) + j \sin(2\pi a_1(x' + t)\Delta t)$$

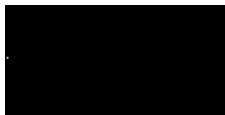
- Tento postup tedy generuje komplexní sinusoidu s frekvencí a_1
- Pokud by se objekt mezi snímky posouval o V_1 pixelů ve směru osy x , měla by sinusová složka frekvenci $V_1 a_1$
- V důsledku symetrie DFT, dostaneme 2 peaky (jeden má frekvenci $V_1 a_1$ a druhý na frekvenci $K - V_1 a_1$)
- Vyhledání maxima ve Fourierově spektru tedy poskytne hodnotu $V_1 a_1$
- Po vydělení číslem a_1 získáme složku rychlosti ve směru osy x , tj. V_1

Příklad

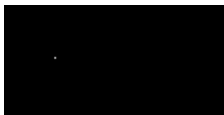
V příkladu dále se bod pohybuje rychlostí 4 px na snímek, odhadnutá rychlost při volbě $a_1 = 5$ je 4.26 px/snímek.

Využití pohybu při segmentaci

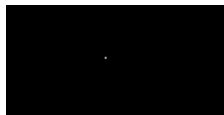
Frekvenční doména



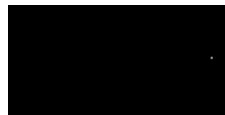
Snímek 0



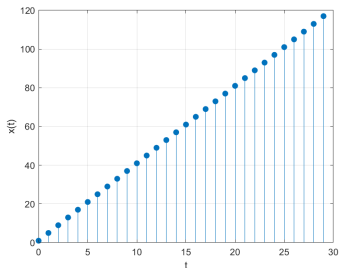
Snímek 7



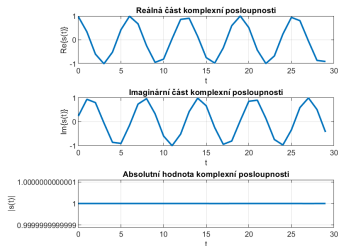
Snímek 14



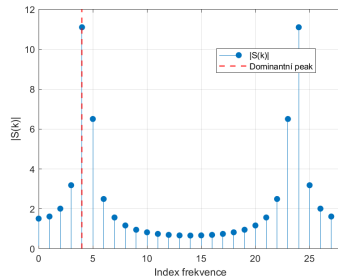
Snímek 29



Trajektorie bodu



Frekvenční doména



Magnituda

Využití pohybu při segmentaci

Frekvenční doména

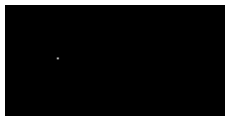
- Analogická analýza poskytne složku rychlosti V_2 ve směru osy y
- Posloupnost snímků bez pohybu produkuje identické exponenciální členy, jejichž Fourierova transformace obsahuje pouze jediný vrchol na frekvenci 0 (tzv. DC složku)

Příklad

V příkladu dále se bod pohybuje rychlostí 0 px na snímek (je statický), odhadnutá rychlost při volbě $a_1 = 5$ je 0 px/snímek .

Využití pohybu při segmentaci

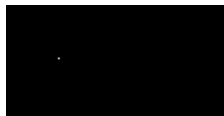
Frekvenční doména



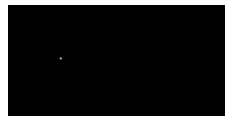
Snímek 0



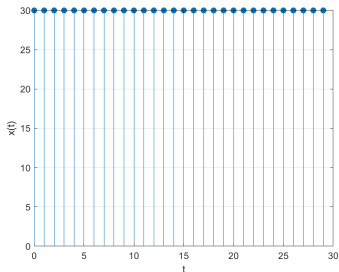
Snímek 7



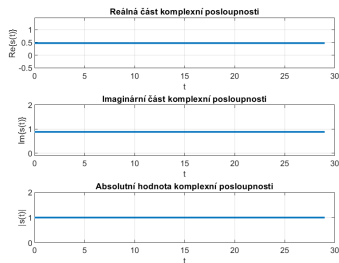
Snímek 14



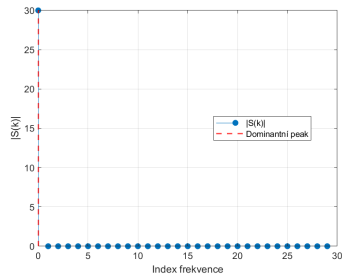
Snímek 29



Trajektorie bodu



Frekvenční doména



Magnituda

Využití pohybu při segmentaci

Frekvenční doména

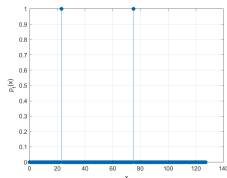
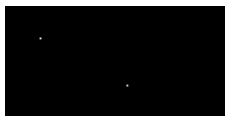
- Protože pohyb pixelu je lineární, bude obecný případ s jedním nebo více pohybujícími se objekty a libovolným statickým pozadím mít Fourierovu transformaci která obsahuje:
 - vrchol v DC složce odpovídající statickým částem obrazu,
 - vrcholy na frekvencích úměrných rychlostem pohybujících se objektů

Příklad

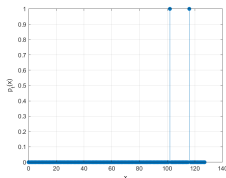
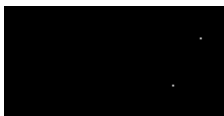
V příkladu dále máme dva body. Bod 1 se pohybuje rychlostí 3 px a bod 2 5 px na snímek. Odhadnuté rychlosti při volbě $a_1 = 5$ jsou 3 px/snímek a 5 px/snímek.

Využití pohybu při segmentaci

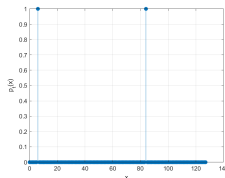
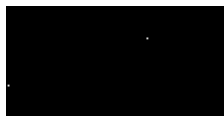
Frekvenční doména



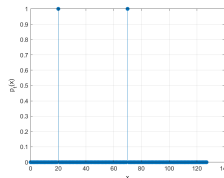
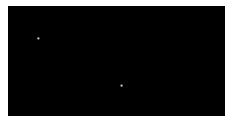
Snímek 0



Snímek 31



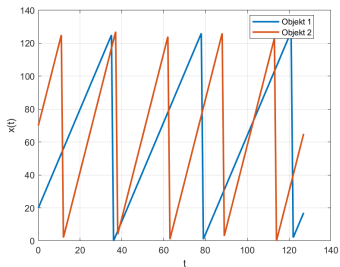
Snímek 63



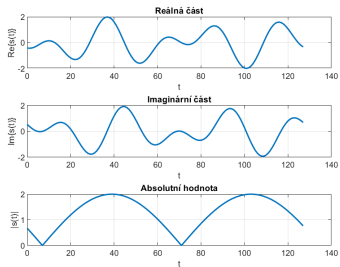
Snímek 127

Využití pohybu při segmentaci

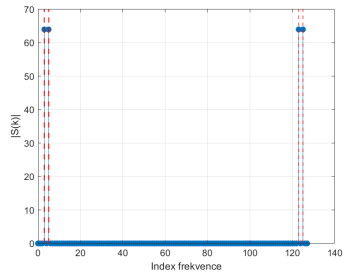
Frekvenční doména



Trajektorie bodů



Frekvenční doména



Magnituda

Využití pohybu při segmentaci

Frekvenční doména

- Pro posloupnost K digitálních obrazů velikosti $M \times N$ je součet vážených projekcí na osu x

$$g_x(t, a_1) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y, t) e^{j2\pi a_1 x \Delta t}$$

- Analogicky projekce na osu y

$$g_y(t, a_2) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y, t) e^{j2\pi a_2 y \Delta t}$$

- 1D Fourierovy transformace

$$G_x(u, a_1) = \sum_{t=0}^{K-1} g_x(t, a_1) e^{-j2\pi ut/K}$$

$$G_y(u, a_2) = \sum_{t=0}^{K-1} g_y(t, a_2) e^{-j2\pi ut/K}$$

- Vztah mezi frekvencí a rychlostí

$$u_1 = a_1 V_1,$$

$$u_2 = a_2 V_2$$

- Jednotkou rychlosti je počet pixelů za dobu trvání celé sekvence snímků

Příklad

Můžeme nějak zjistit skutečnou rychlost?

Využití pohybu při segmentaci

Frekvenční doména

- Skutečná fyzikální rychlost závisí na snímkové frekvenci a vzdálenosti mezi pixely

Příklad

Máme $V_1 = 10$, $K = 30$, snímkovací frekvence je 2 snímky za sekundu a vzdálenost mezi pixely je 0.5 m.

Jaká je skutečná rychlost ve směru osy x ?

Využití pohybu při segmentaci

Frekvenční doména

Příklad

Máme $V_1 = 10$, $K = 30$, snímková frekvence je 2 snímky za sekundu a vzdálenost mezi pixely je 0.5 m.

Jaká je skutečná rychlost ve směru osy x ?

- Skutečná fyzikální rychlost:

$$V_1 = (10 \text{ pixelů})(0.5 \text{ m/pixel})(2 \text{ snímky/s})(30 \text{ snímků})m/s$$

Využití pohybu při segmentaci

Frekvenční doména

- Volba parametrů a_1 a a_2
- Pokud zvolíme nevhodně, dochází k aliasu (špatně zvolené vzorkování)
- Běžná volba

$$a \approx \frac{u_{\max}}{V_{\max}}$$

u_{\max} maximální frekvence bez aliasingu daná počtem snímků K

V_{\max} maximální očekávaná rychlost objektu

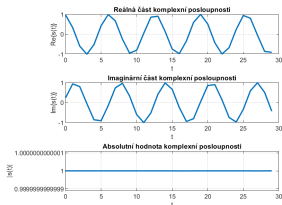
Příklad

V příkladu dříve se bod pohyboval rychlostí 4 px na snímek, odhadnutá rychlost při volbě $a_1 = 5$ byla 4.26 px/snímek. Počet snímků byl 32. Jaká by měla být vhodnější volba a_1 ?

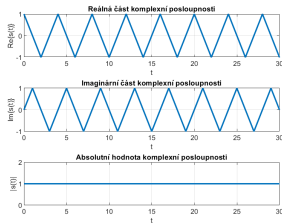
Využití pohybu při segmentaci

Frekvenční doména

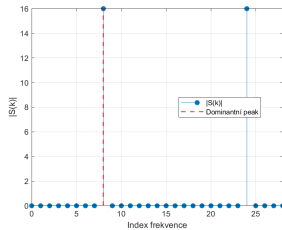
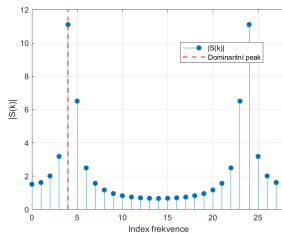
$$a_1 = 5$$



$$a_1 = 8$$



Frekvenční doména



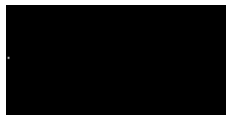
Magnituda

Odhadovaná rychlost je 4 px/snímek.

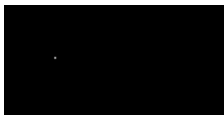
Využití pohybu při segmentaci

Frekvenční doména

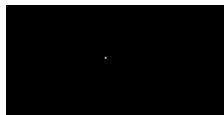
$$a_1 = 8$$



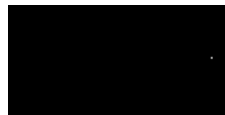
Snímek 0



Snímek 7



Snímek 14



Snímek 29

Využití pohybu při segmentaci

Frekvenční doména

- Znaménko složky rychlosti ve směru osy x lze určit pomocí

$$S_{1x} = \frac{d^2 \operatorname{Re}(g_x(t, a_1))}{dt^2}$$

$$S_{2x} = \frac{d^2 \operatorname{Im}(g_x(t, a_1))}{dt^2}$$

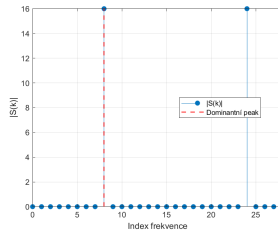
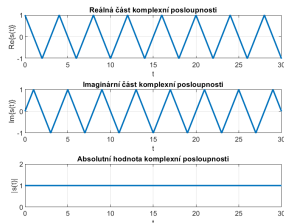
$$t = n$$

- Protože g_x je sinusová funkce, platí, že pokud mají S_{1x} a S_{2x} ve zvoleném čase $t = n$ stejné znaménko, je rychlost V_1 kladná
- Pokud mají znaménka opačná, je rychlost V_1 záporná
- Pokud je jedna z hodnot nulová, vyhodnotí se nejbližší časový okamžik $t = n \pm \Delta t$
- Analogické vztahy platí pro složku rychlosti V_2

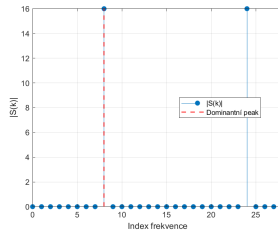
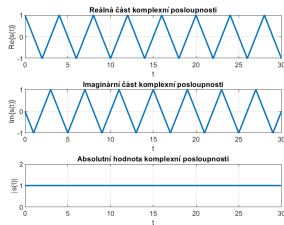
Využití pohybu při segmentaci

Frekvenční doména

Rychlost 4



Rychlost -4



Frekvenční doména

Magnituda

Odhadovaná rychlost v obou případech je 4 px/snímek.

Segmentace barevných obrazů

HSI

- Barva je přímo ve složce H
- S se často používá jako maska, které oblasti nás v odstínu zajímají
- Složka I se při segmentaci barevných obrazů moc nepoužívá



Barevný



H



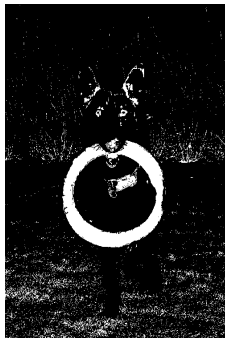
S



I

Segmentace barevných obrazů

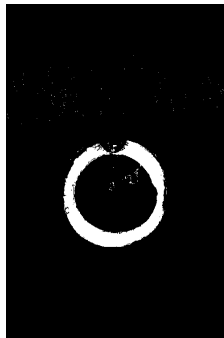
HSI



Maska $S > 0.6$



Maskované H



$0.7 < mH < 0.8$



Výsledek

Segmentace barevných obrazů

RGB

- HSI je intuitivnější, ale lepších výsledků se dosahuje většinou v RGB (při segmentaci)
- Chceme segmentovat objekty v určitém rozsahu barev v RGB obraze
- Z množiny vzorků barev reprezentujících hledanou barvu odhadneme průměrnou barvu **a**
- Průměrná barva definuje rozsah, segmentovaný obraz představuje informaci, zda pixel patří do tohoto rozsahu nebo ne
- K provedení této klasifikace potřebujeme míru podobnosti (např. Eukleidovská vzdálenost)
$$(R_i - R_a)^2 + (G_i - G_a)^2 + (B_i - B_a)^2$$
- Pokud je tato vzdálenost menší, než předepsaný rozsah, pixel *i* bude součástí segmentované oblasti

Segmentace barevných obrazů

RGB



Barevný



Vzdálenost



Maska



Výsledek

Segmentace barevných obrazů

RGB

- Jednoduchá eukleidovská vzdálenost předpokládá, že rozptyl dat je ve všech směrech stejný
- Ve skutečnosti mohou být vzorky barev rozloženy nerovnoměrně
- Proto zavádíme kovarianční matici \mathbf{C} , která popisuje rozptyl a vzájemnou závislost složek R , G , B
- Podobnost bodu i k průměrné barvě \mathbf{a} :
$$D(i, \mathbf{a}) = \sqrt{(i - \mathbf{a})^T \mathbf{C}^{-1} (i - \mathbf{a})}$$
- Tato vzdálenost zohledňuje nejen rozdíl od průměru, ale i tvar rozložení dat
- Jde o zobecnění eukleidovské vzdálenosti
- Množina bodů netvoří v prostoru kouli, ale elipsoid

Segmentace barevných obrazů

RGB

- Výpočet vzdálenosti v RGB prostoru může být výpočetně náročný
- Jednodušším kompromisem je použití **ohraničujícího kvádru**
- Kvádr je umístěn kolem průměrného barevného vektoru **\mathbf{a}**
- Rozměry kvádru v osách R , G , B) jsou voleny úměrně směrodatným odchylkám vzorků barev
- Kvádr tak přibližně odpovídá rozptylu barev, které chceme segmentovat
- Test příslušnosti pixelu ke kvádru je výpočetně velmi jednoduchý
- Je výrazně rychlejší než testování příslušnosti ke kouli nebo elipsoidu v RGB prostoru