

# Detekce geometrických primitiv

Ilona Janáková



---

Rozvrh přednášky:

1. Segmentační metody.
2. Metody orientované na regiony.
3. Znalostní metody.
4. Hybridní metody.
5. Segmentace textur.
6. Segmentace – barevná hloubka obrazu.
7. Vyplňování oblastí, indexace oblastí.

# Detekce geometrických primitiv

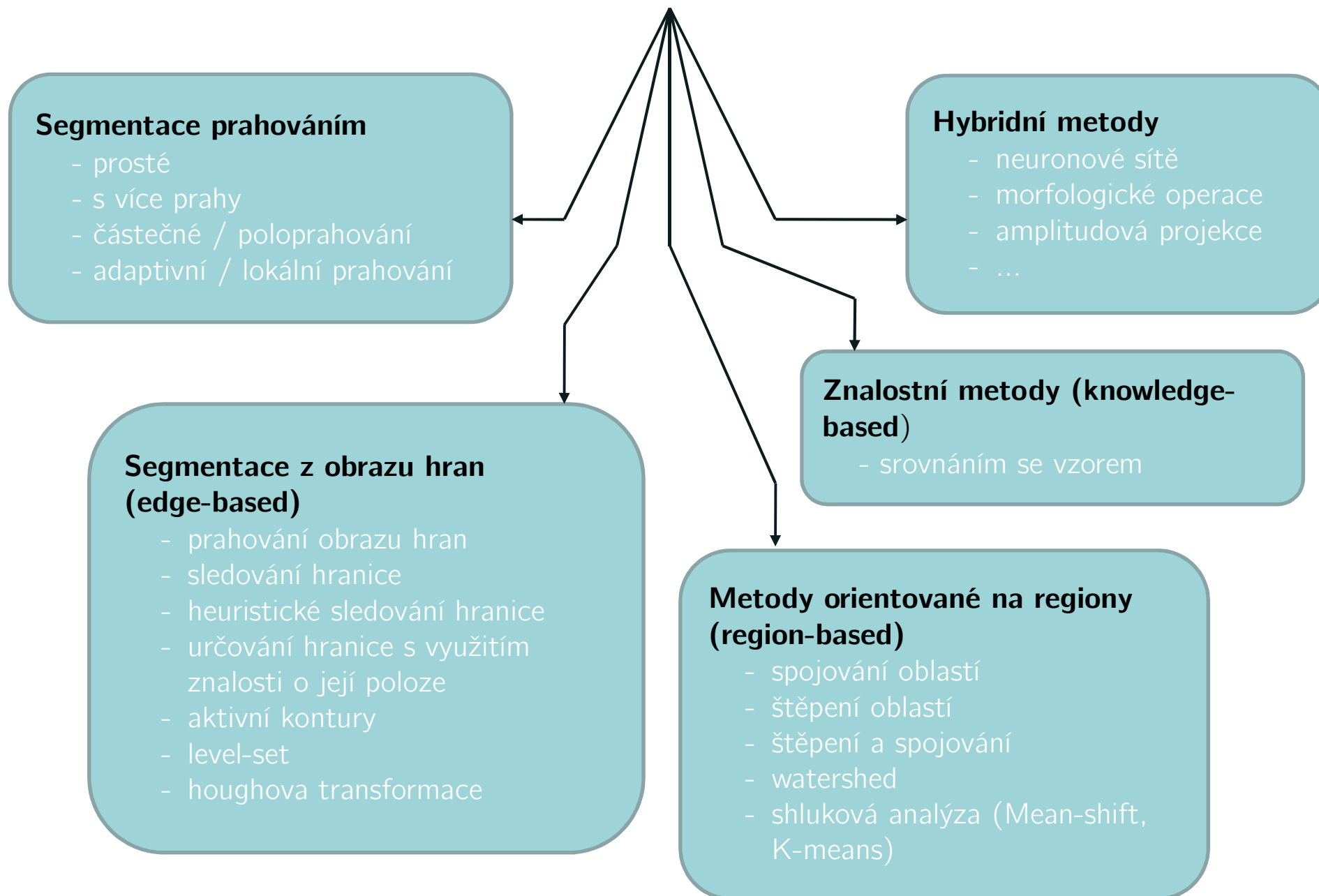
Ilona Janáková



Rozvrh přednášky:

- 1. Segmentační metody.**
2. Metody orientované na regiony.
3. Znalostní metody.
4. Hybridní metody.
5. Segmentace textur.
6. Segmentace – barevná hloubka obrazu.
7. Vyplňování oblastí, indexace oblastí.

# Segmentační metody



# Detekce geometrických primitiv

Ilona Janáková



---

Rozvrh přednášky:

1. Segmentační metody.
- 2. Metody orientované na regiony.**
3. Znalostní metody.
4. Hybridní metody.
5. Segmentace textur.
6. Segmentace – barevná hloubka obrazu.
7. Vyplňování oblastí, indexace oblastí.

# Metody orientované na regiony

- ▶ Snaha o rozčlenění obrázku do maximálních souvislých homogenních oblastí z hlediska zvoleného parametru

Kritérium homogenity:

- jasové vlastnosti,
  - barva,
  - textura,
  - model obrazu,
  - poloha,
  - ...
- ▶ Nemusí dávat stejné výsledky jako metody založené na detekci hran - kombinace metod
  - ▶ Vhodné pro obrazy se šumem, kde se hranice určují obtížně

## **Metody orientované na regiony (region-based)**

- spojování oblastí
- štěpení oblastí
- štěpení a spojování
- watershed
- shluková analýza (Mean-shift, K-means)

# Spojování oblastí

- ▮ Narůstání oblastí spojováním malých homogenních oblastí
- ▮ Pro oblasti požadujeme splnění těchto podmínek:
  1.  $H(R_i) = \text{TRUE}$  pro  $i = 1, 2, \dots, l$
  2.  $H(R_i \cup R_j) = \text{FALSE}$  pro  $i, j \leq 1, 2, \dots, l; i \neq j; R_i$  sousedí s  $R_j$
 kde  $l$  ... počet oblastí,  $R_i$  ... jednotlivé oblasti,  $H(R_i)$  ... dvouhodnotové vyjádření kritéria homogenity
   
 ↙ oblasti musí být (1) homogenní a (2) maximální

## Algoritmus:

1. Definuj:
  - a) počáteční rozdělení obrazu do velkého množství malých oblastí (nejlépe jen bod),
  - b) kritérium spojování dvou sousedních oblastí,
  - c) postup předkládání oblastí.
2. Spojuj sousední oblasti vyhovující kritériu.
3. Pokud již nelze spojit žádné dvě oblasti bez porušení kritéria, skonči.

Různé výsledky pro různé:

- definice počátečních oblastí
- kritéria
- počátky spojování
- pořadí předkládaných oblastí

Příklad:

Kritérium: Pokud rozdíl středních jasů dvou sousedních oblastí je menší než 5, tak tyto oblasti spoj.

			38	41	40	44
		34	39	42	38	
			34	45	47	
				48	57	

- ← 1. krok
- ↓ 2. krok
- ← 3. krok

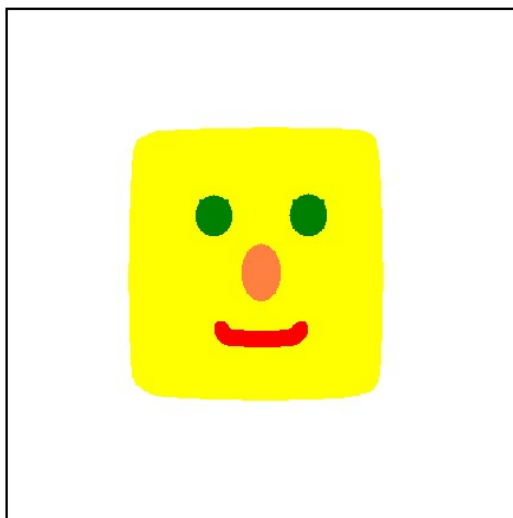
# Štěpení oblastí, Štěpení a spojování

## ► Štěpení oblastí

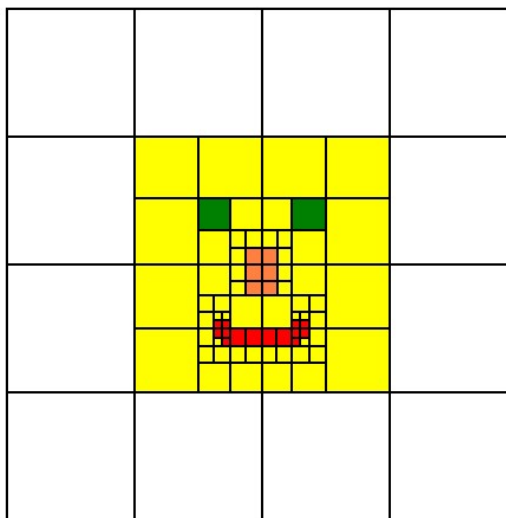
- principiálně opačný přístup než spojování
- na začátku je jediná oblast (celý obraz) a ta se pak dělí dokud podoblasti nesplňují kritérium
- štěpení a spojování mohou dávat různé výsledky
- výsledek lze zapsat pomocí stromové struktury

## ► Štěpení a spojování

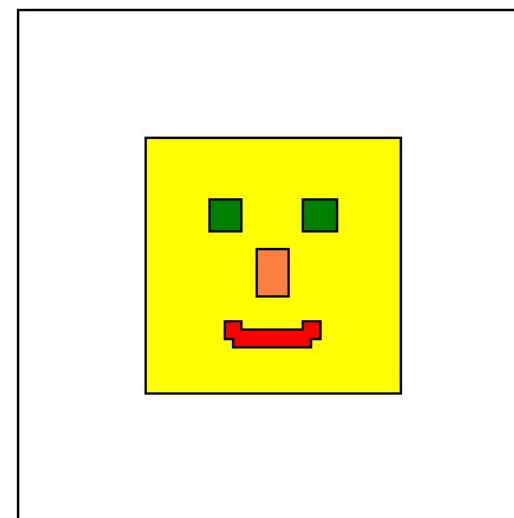
- využívá pyramidové reprezentace obrazu – štěpení a spojování je realizováno v rámci čtvercových oblastí pyramidové datové struktury
- je-li oblast v dané úrovni pyramidy nehomogenní -> oblast je rozdělena na čtyři podoblasti
- jsou-li oblasti navzájem homogenní -> dojde ke spojení do jedné



originál



štěpení



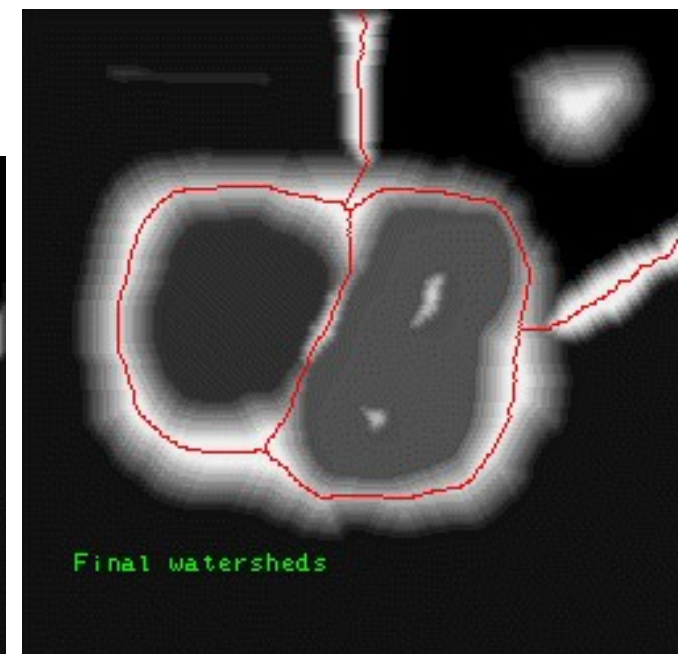
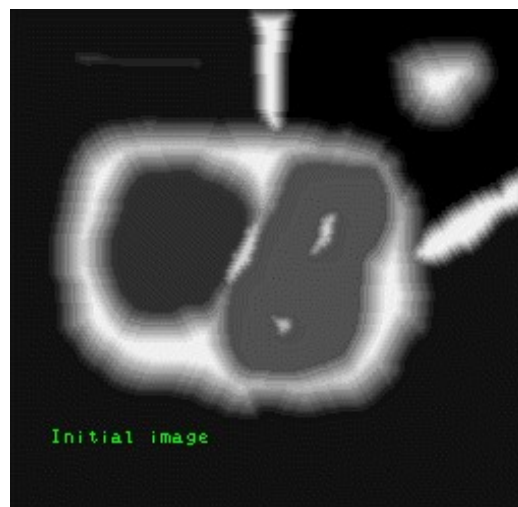
štěpení a spojování

# Watershed

- ▶ *watershed* = rozvodí, povodí či vodní předěl
- segmentační metoda vycházející z geografie - obraz je chápán jako terén nebo topografický reliéf – jas vstupního obrázku určuje výšku terénu (černá nejnižší, bílá nejvyšší)
- princip je založen na postupném zaplňování terénu vodou – stoupání hladiny
- výsledkem je obraz rozdělený do regionů - jednotlivých povodí oddělených hrázemi, všechny body daného povodí jsou označeny stejným unikátním indexem
- varianta s uživatelskou inicializací
- pro obrazy obsahující šum vytváří příliš mnoho oblastí. Lze eliminovat:
  - vhodnou předpřípravou obrázku
  - povolením hráze až určité prahové výšky
  - regiony patřící do stejné oblasti lze později spojit jinými metodami

## Algoritmus:

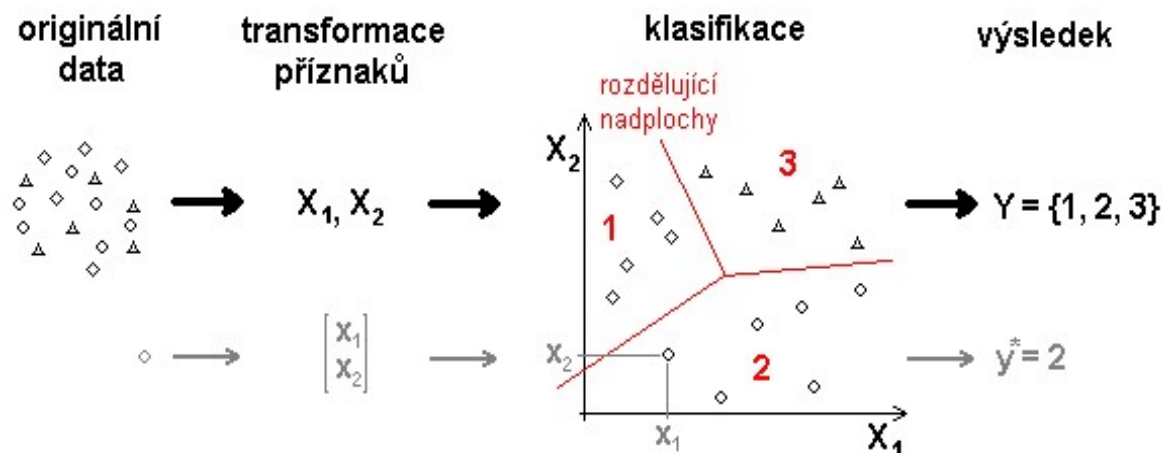
1. Nalezni nejnižší body (lokální minima obrazu).
2. Začni zaplňovat povodí vodou z těchto počátečních bodů – hladina stoupá.
  - v místech, kde by se voda ze dvou různých povodí mohla slít, vytvoř hráze
3. Ve chvíli dosažení nejvyššího bodu terénu (maxima jasů obrazu) skončí.





# Shluková analýza

- Shluková analýza = obecná statistická klasifikační metoda
- Zpracování obrazu -> shlukování pixelů podobných vlastností
  - každý pixel nebo část obrazu je reprezentována vektorem vlastností  $\mathbf{x} = [x_1, x_2, \dots, x_N]$ , což mohou být např. barevné komponenty, vlastnosti okolí, pozice atd. (nutná je normalizace jednotlivých vlastností)
  - je nutné vybrat takové vlastnosti, které mají pixely z jedné oblasti podobné a z různých oblastí rozdílné = měly by v  $N$ -rozměrném prostoru vznikat shluky



- metoda učení bez učitele, iterativní postup
- úlohou segmentace je automaticky určit polohu shluků (někdy i jejich počet) a přiřadit jednotlivým vzorkům nejbližší shluk
- shluková analýza vychází z podobnosti, resp. vzdálenosti vzorků
  - vzdálenost dvou objektů  $r$  a  $s$  (charakterizovanými  $N$  vlastnostmi) lze vyjádřit např.:

$$d(r, s) = \sqrt[k]{\sum_{i=1}^N |x_{ri} - x_{si}|^k}$$

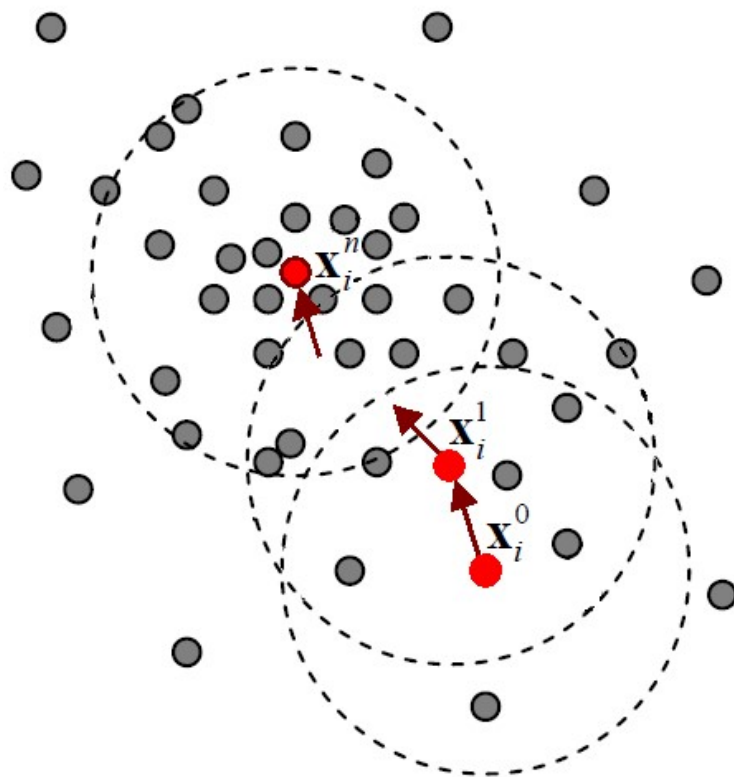
$k = 1$  vzdálenost v městských blocích (Manhattan)  
 $k = 2$  Euklidovská vzdálenost

$$d(r, s) = \max |x_{ri} - x_{si}|$$

Čebyševova vzdálenost

# Mean-shift

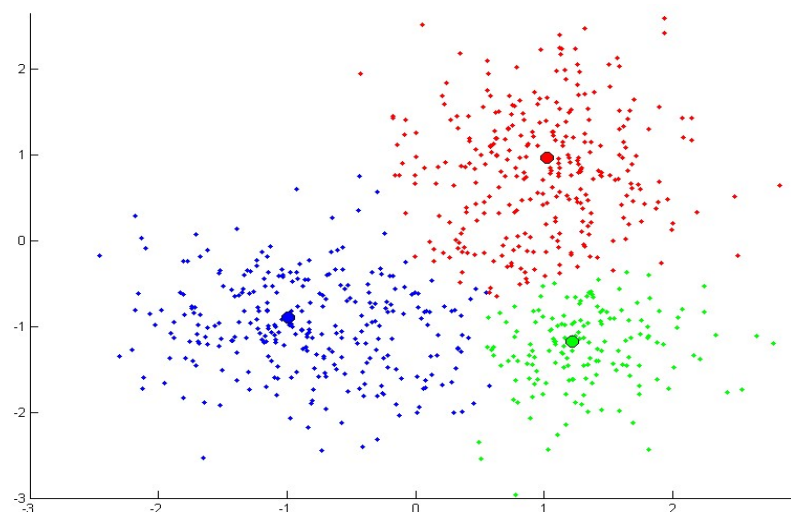
- Mean-shift segmentace shlukuje body obrazu na základě podobnosti jejich vzhledu a blízkosti jejich pozic pomocí konvergence do lokálních maxim spojeného souřadnicového a intenzitního prostoru
- Např.:  $\mathbf{x}_i = [x, y, f(x,y)]$ ,  $\mathbf{x}_i = [x, y, r(x,y), g(x,y), b(x,y)]$



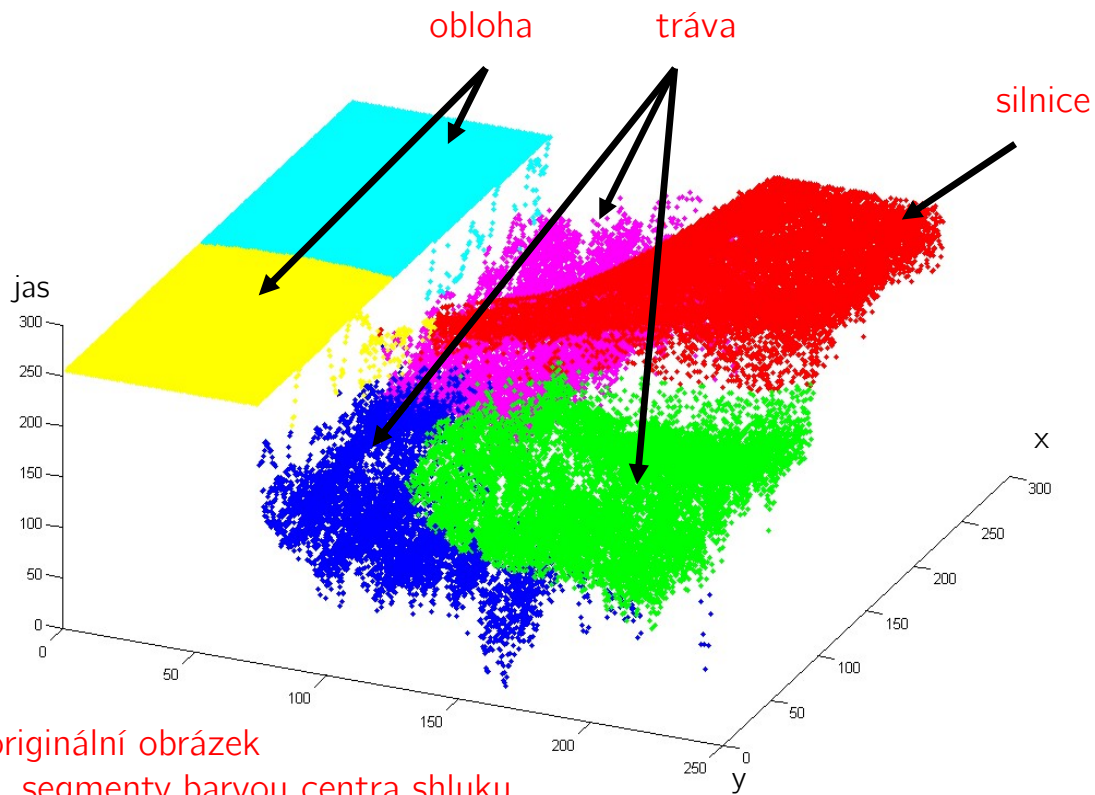
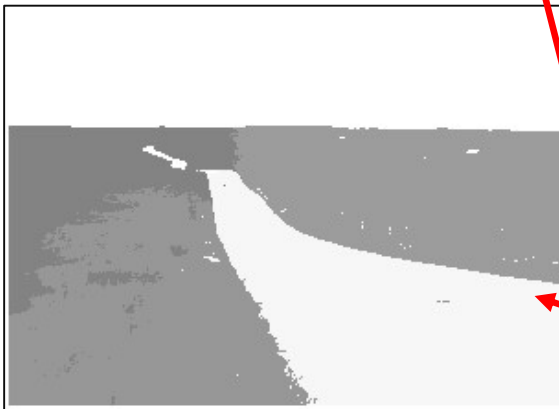
## Algoritmus:

Vstup: množina vzorků (bodů v  $N$ -rozměrném prostoru)

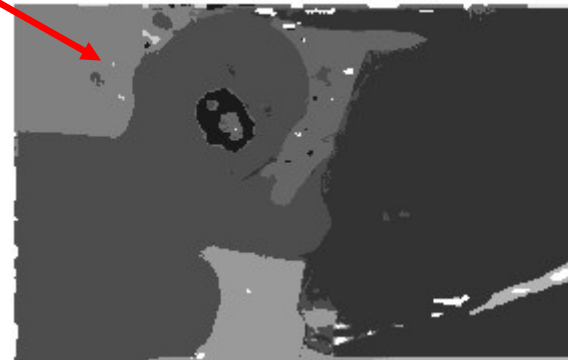
- Definuj velikost a tvar okna
- Pro všechny vzorky:
  - vypočti lokální maximum hustoty vzorků uvnitř okna kolem aktuálního vzorku (*mean* pozice)
  - posuň (*shift*) pozici vzorku do maxima
  - opakuji dokud se mění pozice maxima
  - zapamatuj si poslední pozici
- Shlukujeme ty vzorky, které dokonvergovaly do stejného maxima (s určitou tolerancí).



# Mean-shift



originální obrázek  
segmenty barvou centra shluku  
obrázek indexů



# K-means

- K-means algoritmus iterativně hledá hodnoty vektorů  $\mu_j$  (středů shluků) tak, že minimalizuje střední odchylku mezi zadanou množinou dat a vektory, které mají k těmto datům nejmenší vzdálenost a rozděluje je do předem daného počtu shluků (tříd)  $K$ :  $C_1, C_2, \dots, C_k$ .

## Algoritmus:

Vstup: množina dat  $x_1, x_2, \dots, x_l$  a číslo  $K$  udávající počet shluků (tedy počet vektorů  $\mu_j, j = 1, \dots, K$ )

- Inicializace vektorů  $\mu_j$ : náhodně zvolené hodnoty nebo využitím vhodné heuristiky (např. apriorní informace)
- Iterativní opakování kroků dokud se alespoň jeden vektor  $x_i$  klasifikuje do jiné třídy než byl klasifikován v předcházejícím kroku:

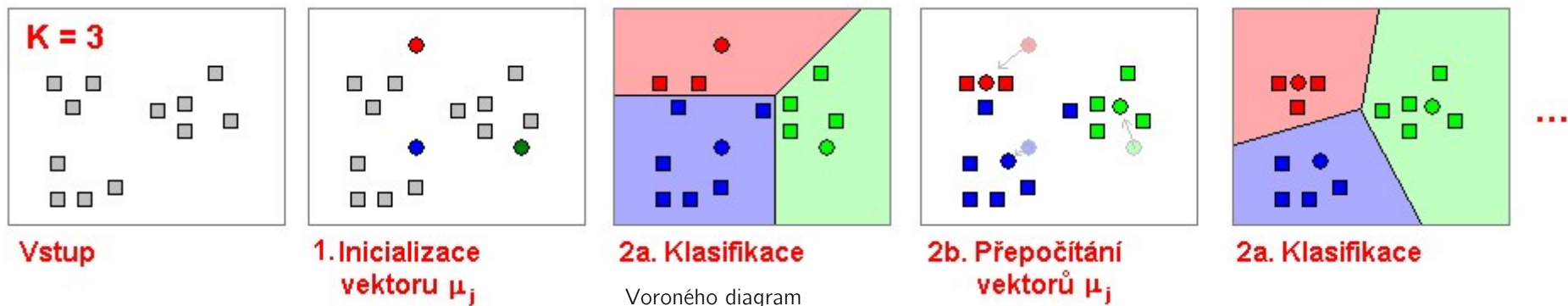
- Klasifikace:** všechna data  $x_i$  se klasifikují do tříd určených vektory  $\mu_j$  podle minima vzdálenosti (většinou euklidovské). Tedy vzor  $x_i$  je přiřazen do třídy  $y_i$ , podle:

$$y_i = \arg \min_j D(x_i, \mu_j)$$

- Přepočítání vektorů  $\mu_j$ :** vypočítají se nové hodnoty vektorů  $\mu_j$  jako střední hodnota dat  $x_i$ , které byly klasifikovány do třídy určené příslušným vektorem  $\mu_j$ . Tedy nová hodnota  $\mu_j$  se určí jako:

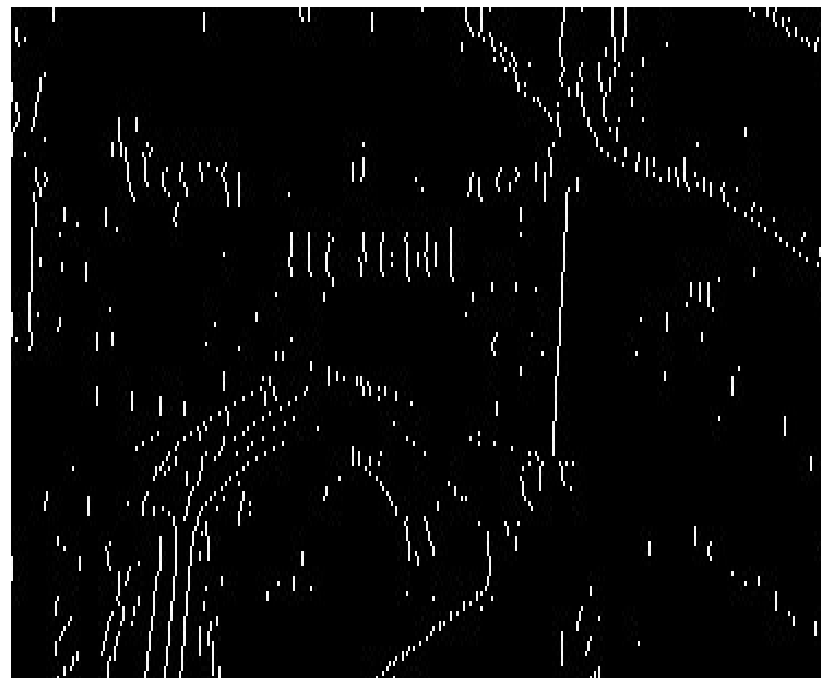
$$\mu_j = \frac{1}{l_j} \sum_{x_i \rightarrow y_j} x_i$$

, kde  $l_j$  je počet vzorů  $x_i$  klasifikovaných v předchozím kroku do třídy určené vektorem  $\mu_j$



# Shluková analýza

- ▶ vstupní vektor příznaků může popisovat vlastnosti ne jen jednotlivých pixelů, ale i většího okolí, vlastnosti oblastí nebo dílčích primitiv obrazu (např. hran)



Př.: Lokalizace SPZ

– ve snímcích po předzpracování (hranové filtry, ztenčování,...)  
se SPZ hledá jako skupina svislých čar (určité délky,  
s podobnými souřadnicemi,...)



# Detekce geometrických primitiv

Ilona Janáková



---

Rozvrh přednášky:

1. Segmentační metody.
2. Metody orientované na regiony.
- 3. Znalostní metody.**
4. Hybridní metody.
5. Segmentace textur.
6. Segmentace – barevná hloubka obrazu.
7. Vyplňování oblastí, indexace oblastí.

# Znalostní metody

- ▶ Patří sem všechny metody využívající dříve získané znalosti o objektech – šablona nebo model
- ▶ Vyhledávání známých objektů nebo jen např. vzorů, textury, kontur, modelů, geometrických útvarů atd. v obrazu pomocí srovnání se vzorem = **template matching**
  - nikdy nemůžeme očekávat absolutní shodu – šum, zkreslení, deformace atd. => hledáme jen maximum vhodného kritéria
  - a to pro všechny uvažované transformace obrazu (natočení, změna měřítka, zkreslení,...)
  - vhodné i pro stereoskopické snímky nebo sledování relativního pohybu – z jednoho snímku se vyjme předloha a v ostatních se pátrá po stejném objektu

## ▶ Metody:

- srovnání regionů, ale možné i srovnávat pouze kontury
- hledání shody částí obrazu jen s jednotlivými částmi vzoru
- srovnávání modelů objektů (modely definovány vlastnostmi jako např. střední jas, velikost, momenty nebo jiné radiometrické či fotometrické příznaky)
- srovnání grafů vytvořených podle vlastností sledovaných objektů

## ▶ Příklady míry souhlasu (hodnotícího kritéria):

$$C_1(u, v) = \frac{1}{\max |f(i+u, j+v) - h(i, j)|}$$

$$C_2(u, v) = \frac{1}{\sum_{(i,j) \in V} |f(i+u, j+v) - h(i, j)|}$$

$$C_3(u, v) = \frac{1}{\sum_{(i,j) \in V} [f(i+u, j+v) - h(i, j)]^2}$$

$f$  ... zpracovávaný obraz

$h$  ... hledaný vzor

$F, H$  ... Fourierovy obrazy

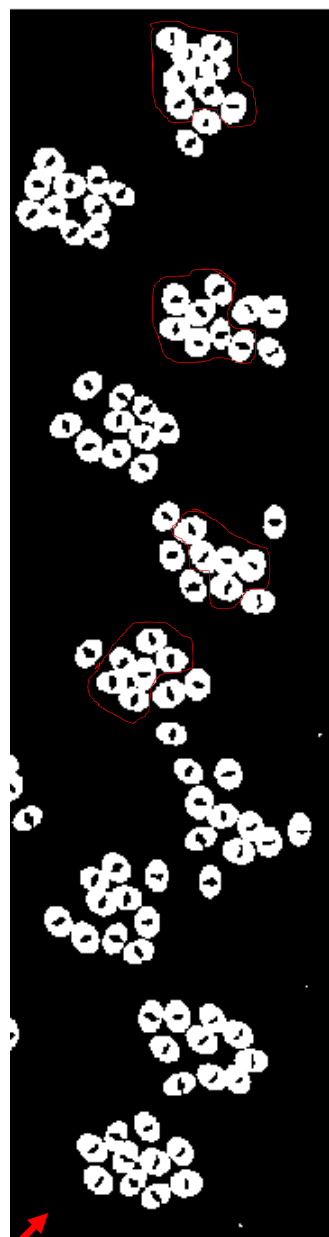
$V$  ... množina všech obrazových elementů vzoru

$C_4$  ...  $F \bullet H$  a následná zpětná transformace

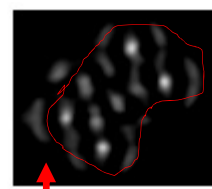
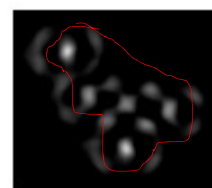
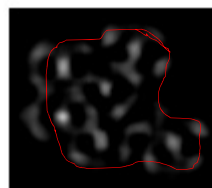
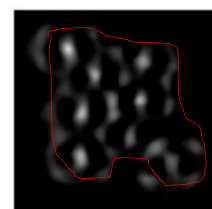
# Srovnání se vzorem – vzájemná korelace



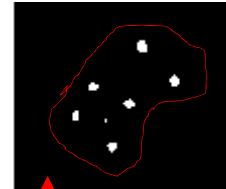
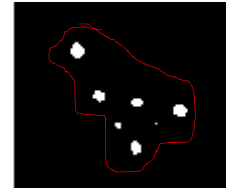
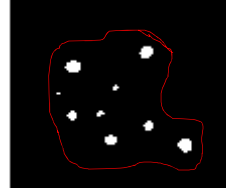
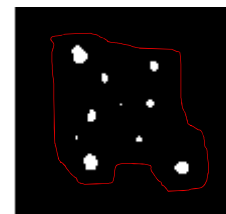
originální snímek



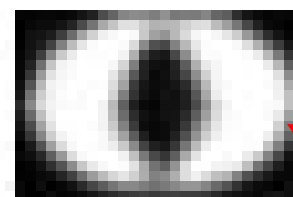
odečtené pozadí + ekvalizace + prahování



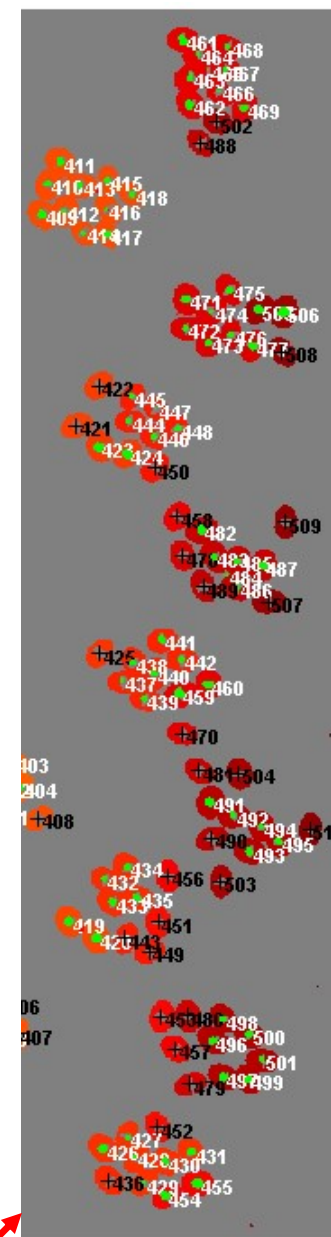
vzájemná korelace se vzorem v základní pozici



součet korelací od čtyř pozic + práh



vzor + rotace



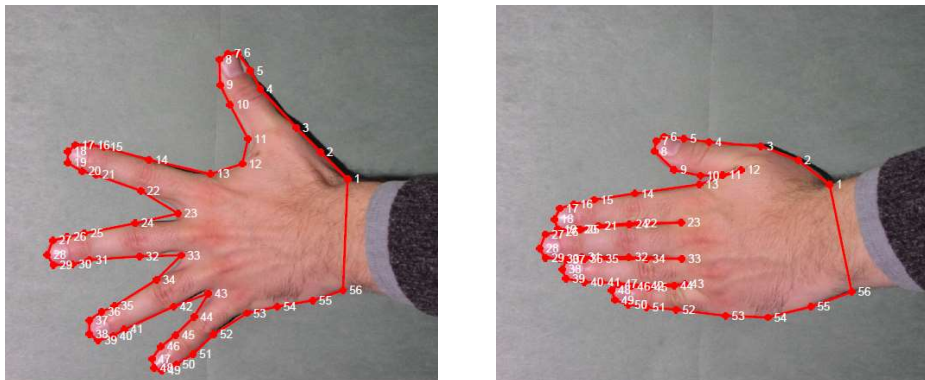
výsledek s označenými indexy



# Active Appearance Models (AAM)

- ▶ AAM využívá statisticky vytvořený model objektů z manuálně segmentovaných trénovacích dat. Parametry získaného modelu je možné přizpůsobit jakémukoliv novému obrazu a ověřit tak přítomnost objektu v obraze
  - model je generován jako kombinace modelu variací tvaru objektu a modelu variací textury (intenzity pixelů)
  - k vytvoření modelu je třeba trénovací sada anotovaných obrazů, kde jsou vždy označeny odpovídající si hraniční body (*landmark points*)
  - v průběhu trénování je zaznamenán vzájemný vztah mezi změnou polohy hraničních bodů a změnou intenzity pixelů v dané množině vzorů
  - vytvoření modelu pomocí statistické metody zpracování dat - PCA analýza (*Principal Component Analysis*)
  - tímto způsobem natrénovaný model umožňuje velice rychlé porovnání modelu s objekty v novém obraze
- ▶ *Výhody*
  - rychlé porovnání / prohledávání obrazu
  - rychlá adaptace modelu na nový obraz
- ▶ *Nevýhody*
  - nutná rozsáhlá a reprezentativní galerie trénovacích vzorů
  - náročná ruční anotace vzorů
  - metoda vyžaduje poměrně přesnou počáteční inicializaci (odhad polohy objektu v testovaném obraze)
  - možnost selhání při porovnávání modelu s novým obrazem

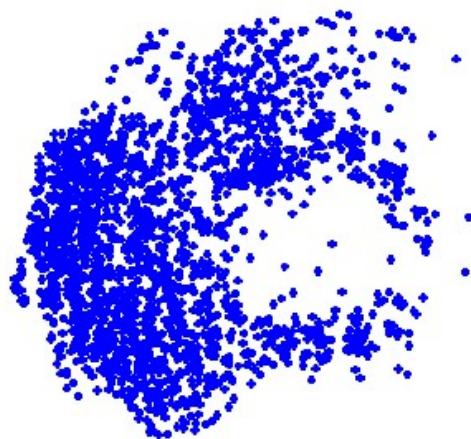
# Active Appearance Models (AAM)



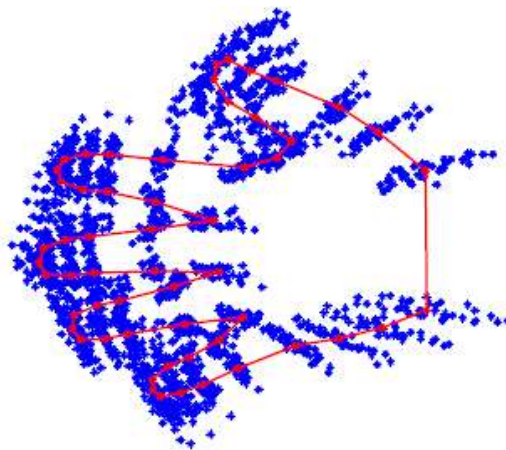
anotované snímky ruky s 56 hraničními body

## Anotace snímků:

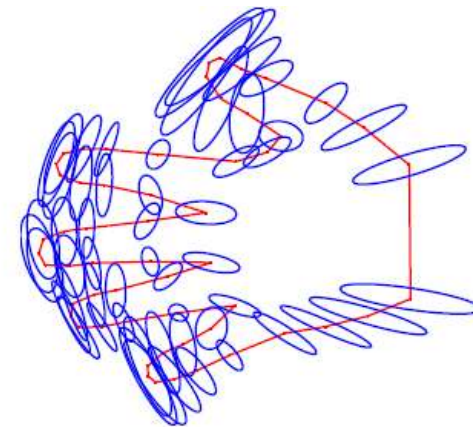
1. Zvolit významné body objektu
2. Najít a označit tyto body na všech snímcích
3. Zarovnat všechny obrázky - translace, rotace, měřítko (např. *Procrustes analysis*, tangenciální projekce)
4. Určit variace polohy všech bodů a variace intenzit
5. Korelační matice bodů – body se pohybují závisle (např. PCA - *principal component analysis*)



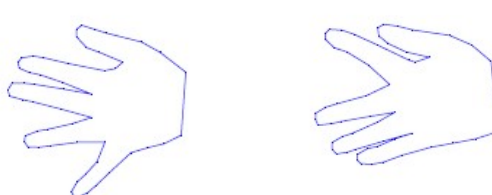
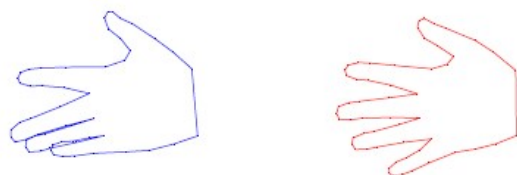
nezarovnané anotace ze 40 snímků



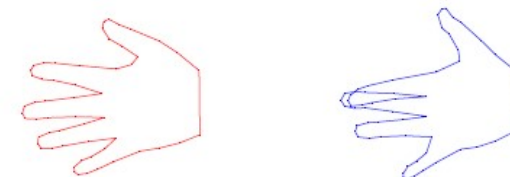
zarovnané anotace se zvýrazněným „středním tvarem“



nezávislá PCA analýza pro každý hraniční bod



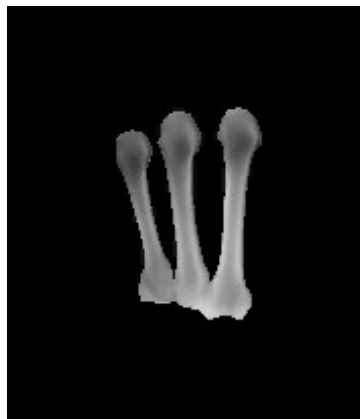
naznačení vzájemného pohybu bodů



# Active Appearance Models (AAM)



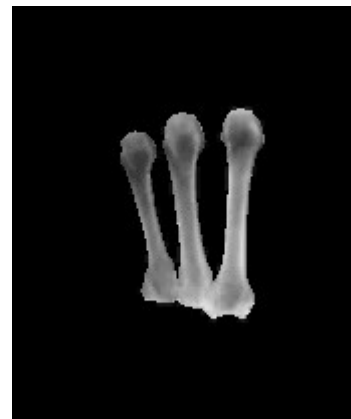
segmentace zápěstních  
kůstek:



model tvaru



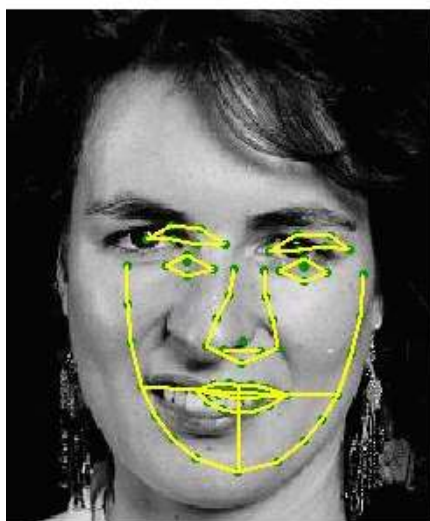
model intenzit



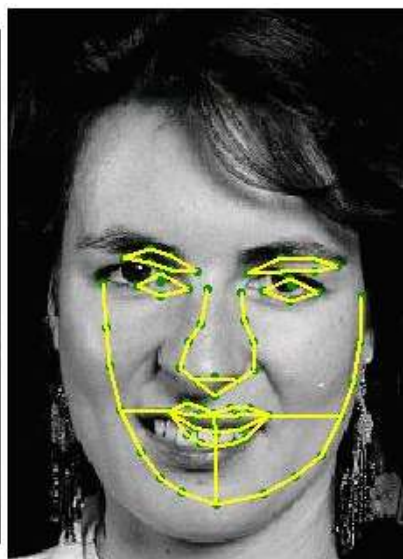
kombinovaný model



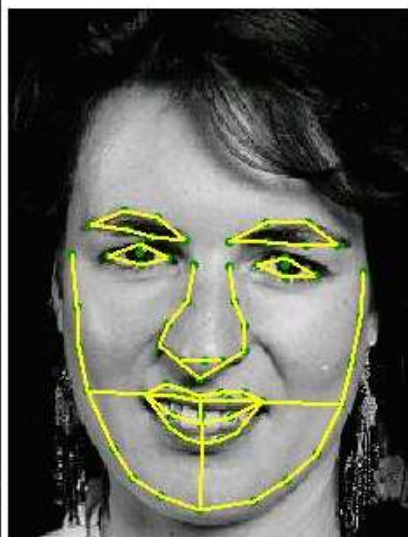
prohledávání



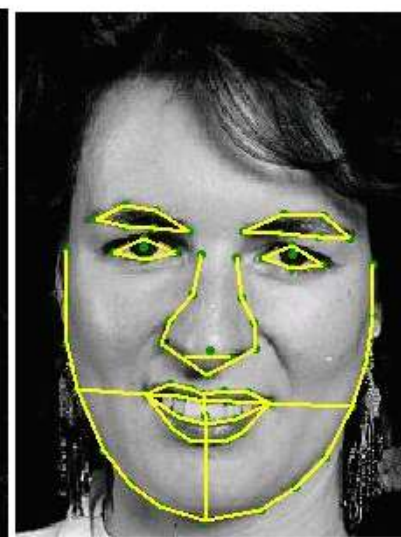
segmentace  
obličeje: inicializace



po první iteraci



po druhé iteraci



konvergence

# Detekce geometrických primitiv

Ilona Janáková



---

Rozvrh přednášky:

1. Segmentační metody.
2. Metody orientované na regiony.
3. Znalostní metody.
- 4. Hybridní metody.**
5. Segmentace textur.
6. Segmentace – barevná hloubka obrazu.
7. Vyplňování oblastí, indexace oblastí.

# Hybridní metody

- ▶ Metody primárně určené k jiným účelům
- ▶ Kombinace různých přístupů
- ▶ Úprava obrazu před nebo po nasazení jiné segmentační techniky
  
- ▶ **Neuronové sítě**
  - obecně nástroj pro klasifikaci, simulaci, zpracování informací atd.
  - paralelní distribuovaný systém jednoduchých výkonných prvků uspořádan účelně tak, aby byl schopen požadovaného zpracování informací
  
- ▶ **Amplitudová projekce, Radonova transformace**
  - technika, kterou je možné použít pro separování obrazových dat
  
- ▶ **Morfologické operace**
  - vhodné především jako úprava/zjednodušení binárního obrazu (např. po prahování), např.:
    - spojení blízkých segmentů, vyplnění děr atd. (dilatace nebo uzavření)
    - rozložení složitého objektu na několik jednodušších (eroze)
    - náhrada složitého objektu jeho skeletem; ztenčování
    - odstranění objektů dle velikosti
  
- ▶ **Segmentace na základě pohybu**

# Neuronové sítě

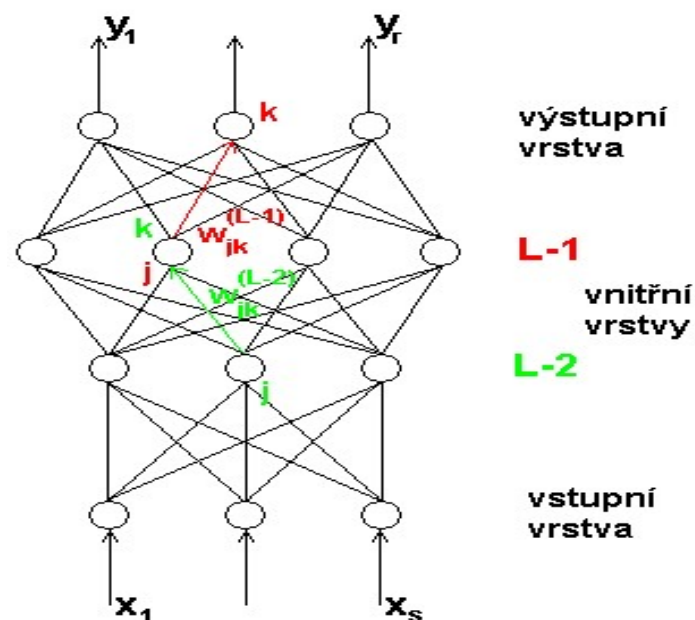
- ▶ NN jsou rozsáhlé paralelní sítě obsahující jednoduché výpočetní elementy (neurony) schopné provádět pouze jednoduché výpočty
- ▶ učení se děje adaptací vah na propojích mezi jednotlivými neurony (znalosti se ukládají do vazeb)
- ▶ simulují schopnost živých organismů učit se
- ▶ Dva přístupy:
  - hledáme charakteristické vlastnosti vstupních dat (obvykle příznakové vektory) a klasifikujeme je do tříd bez jakékoliv další interpretace - *učení bez učitele* (podobné shlukové analýze)
  - *trénování s učitelem* vyžaduje ručně segmentovaná trénovací data. Vstupem učícího algoritmu jsou nejen příznakové vektory, ale i funkce, která každému vstupnímu vektoru přiřazuje určitý segment obrazu.

## ▶ Vlastnosti NN:

- učení se z příkladů a zobecňování znalostí
- potlačení šumu
- odolnost vůči defektům a poruchám ve vstupních datech

## ▶ Použití:

- klasifikace – pomocí trénovacích obrazů jsou nastaveny váhy v síti. Síť je pak schopna segmentovat neznámé obrazy nebo klasifikovat segmentované obrazy do tříd
- shlukování
- deformační modely

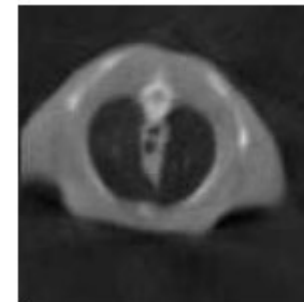
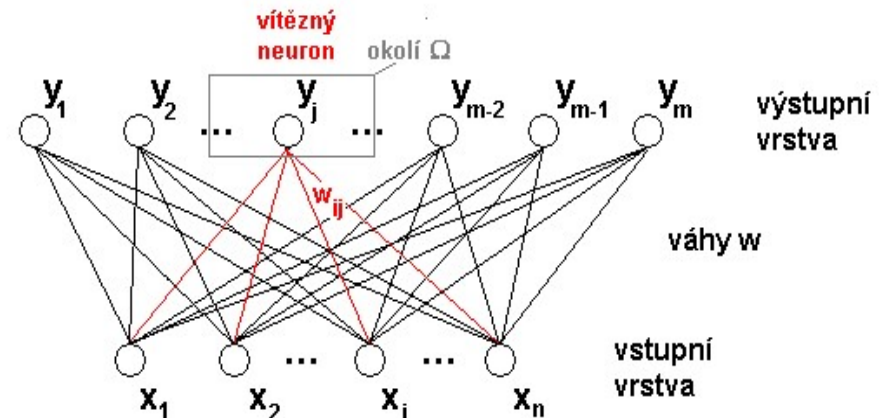


# Neuronové sítě – Kohonenova mapa

- ▶ Samoorganizující se síť = učení bez učitele, učení se z příkladů, podobný shlukování
- využívá se soutěžní strategie učení = neurony ve výstupní vrstvě spolu soutěží o to, který bude aktivní (do které třídy bude vstup zařazen)
- síť je dvouvrstvá, každý neuron výstupní vrstvy je spojen se všemi neurony vstupní vrstvy
  - počet vstupů je dán počtem měřených vlastností pixelu nebo oblastí (jas, pozice, vlastnosti okolí atd.)
  - počet výstupů určuje počet očekávaných shluků – segmentů obrazu (problém s jejich vhodným určením)

## Proces učení:

1. Inicializace vah (např. náhodně nebo rovnoměrně), parametru učení  $\alpha(t)$  a velikosti okolí  $\Omega(t)$
2. Předložení vzoru  $\mathbf{x}(t) = [x_0(t), x_1(t), \dots, x_n(t)]$ 
  - a. výpočet vzdálenosti vzoru pro všechny výstupní neurony, např. euklidovská vzdálenost:
 
$$d_j = \sum_{i=0}^n [x_i(t) - w_{ij}(t)]^2$$
  - b. výběr nejbližšího (nejpodobnějšího) neuronu = vítězný neuron
 
$$d_j^* = \min(d_j)$$
  - c. úprava vah nejbližšího neuronu (případně i okolních), ostatní váhy zůstávají stejné
 
$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) \cdot [x_i(t) - w_{ij}(t)]$$
  - d. je možné upravit i další parametry – parametr učení či velikost okolí
3. Krok 2. opakujeme pro všechny vzory nebo do dosažení potřebné přesnosti



## Proces segmentace:

1. Pro celý obrázek
  - a. předložení nového vzoru  $\mathbf{x}(t)$
  - b. výpočet vzdáleností a určení nejbližšího neuronu = přiřazení indexu daného vítězného segmentu

# Neuronové sítě – Konvoluční sítě

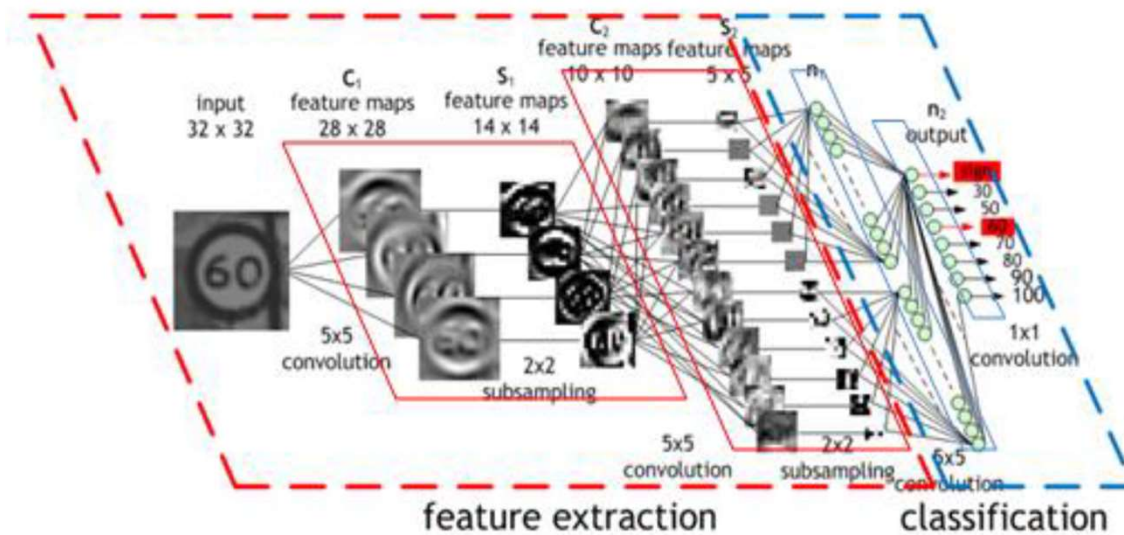
- ▮ Dopředné sítě; často hluboké
- ▮ Základními stavebními kameny CNN jsou konvoluční a redukční vrstvy
- ▮ Vstupem sítě není příznakový vektor obrazu, ale obraz samotný - snaha o napodobení zpracování obrazu v lidském mozku
- ▮ **Konvoluční vrstva** – z několika příznakových map. Výstupy jednotlivých vrstev nejsou získávány pomocí aktivační funkce, ale pomocí konvoluce vstupu a příslušného konvolučního jádra + sdílení vah – jedna a ta samá váha sdílená více neurony
  - příznaková mapa – lokální charakteristiky obrazu – podobrazy, výřezy. Každý podobraz (např. 5x5 pxl) je vstupem jednoho neuronu a tento neuron je díky sdílení vah rozkopírován pro každou podoblast dané velikosti (podoblasti se mohou i překrývat)
- ▮ **Redukční** (subsamplingová, pooling) **vrstva** – napojena na předchozí konvoluční vrstvu a obsahuje stejné množství příznakových map. Představuje proces podvzorkování – urychlení procesu konvergence sítě a nalezení robustnějších obrazových příznaků. Většinou operace max nebo mean.
- ▮ Klasifikační vrstva – nemusí být (plně konvoluční neuronové sítě) nejvyšší vrstva, tolik neuronů, kolik je klasifikačních tříd, plně propojená
- ▮ Učení CNN probíhá obdobným způsobem jako učení klasické neuronové sítě - jen s tím rozdílem, že nedochází k aktualizaci vah, ale jsou upravovány formy jednotlivých konvolučních jader

## Výhody oproti klasickým vrstevnatým sítím:

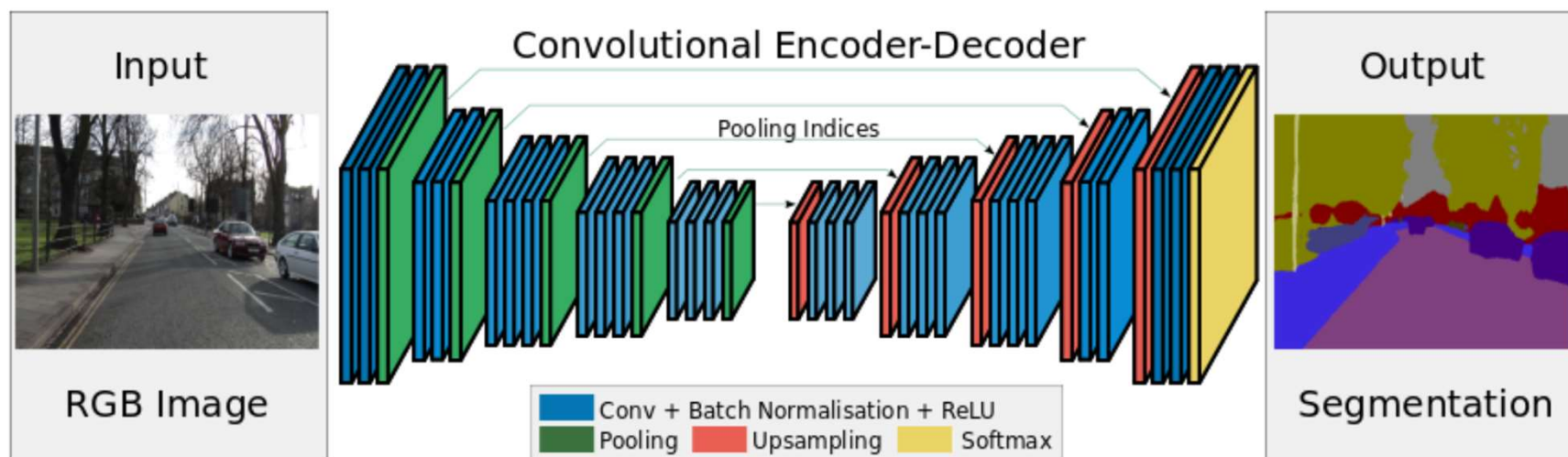
- vhodné pro zpracování obrazu
- posunutý vzor, zdeformovaný vzor
- více vrstev umožňuje zachycení vyšší abstrakce
- rychlá klasifikace



# Neuronové sítě – Konvoluční sítě



Architektura sítě CNN



Architektura sítě SegNet

# Amplitudová projekce

- obrazové segmenty mohou být někdy vhodně separovatelné vytvořením projekce ve směru řádků a sloupců = vertikální a horizontální projekce

$$V(y) = \sum_{x=1}^X f(x, y), \quad H(x) = \sum_{y=1}^Y f(x, y)$$

**Rozpoznávání písma**

Umístění obrázku:  
Testovací vzorky/text.png

THE AVARD WINING WEB BROWSER JUST GOT  
BETTER. ITS FREE, AND EASY TO USE.  
FIREFOX 1.5 HAS INTUITIVE INTERFACE  
ENJOY QUICK PAGE LOADING  
13.12.2005

THEAVARDWININGWLBROWSERJD3TQOT

Výřez Normalizace Vzor Dilatace Eroze

Podobnost: 0.867363

Analyzovaný znak Vzor Součin

Rozpoznat

Rozpoznat řádek THEAVARDWININGWLBROWSERJD3TQOT

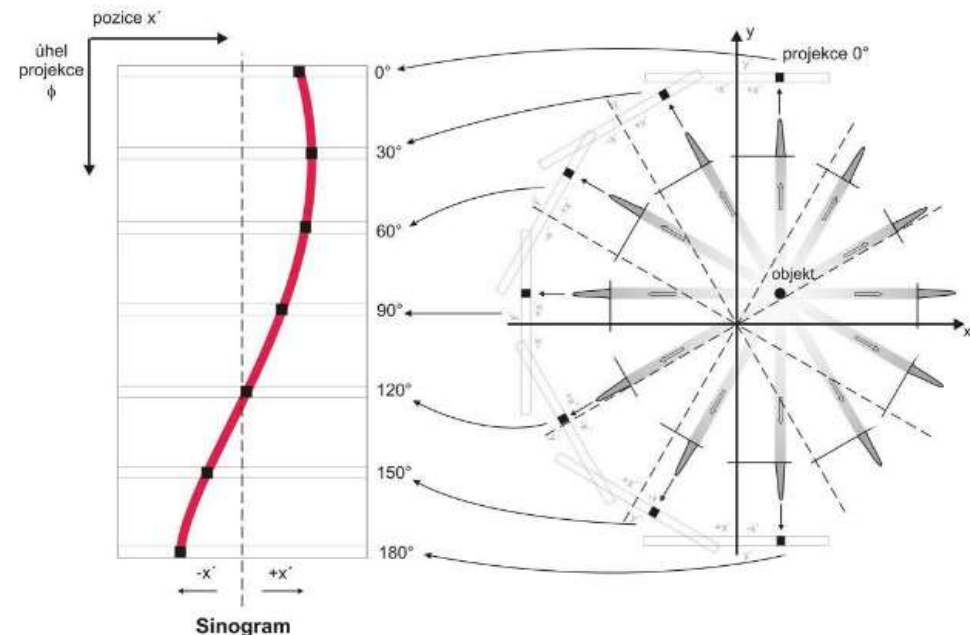
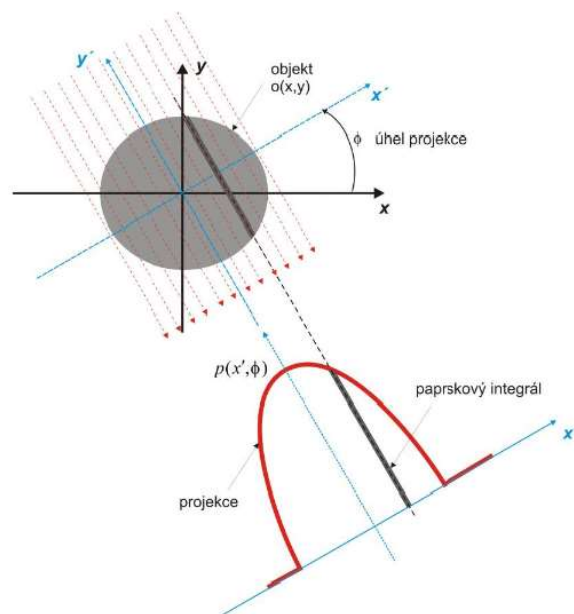
# Radonova transformace

- ▮ integrál funkčních hodnot podél parametricky zadané přímky – amplitudová projekce pod obecným úhlem

$$p_{\phi}(x') \equiv p(x', \phi) \equiv R[o(x, y)] \quad , \quad \text{kde } R[o(x, y)] \text{ je Radonův operátor}$$

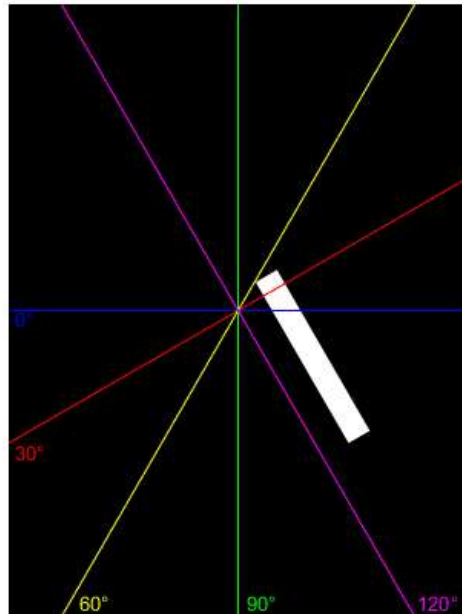
$$R[o(x, y)] = \iint_{-\infty}^{\infty} o(x, y) \delta(x \cdot \cos\phi + y \cdot \sin\phi - x') dx dy = \int_{-\infty}^{\infty} o(x' \cdot \cos\phi - y' \sin\phi, x' \cdot \sin\phi + y' \cdot \cos\phi) dy'$$

- ▮ soubor projekcí pod měnícím se úhlem nazýváme sinogram (Radonův prostor)
  - osa  $y$  představuje úhel pořízené projekce
  - osa  $x'$  představuje pozici měření paprskového integrálu
  - funkční hodnoty jsou hodnoty daných paprskového integrálu

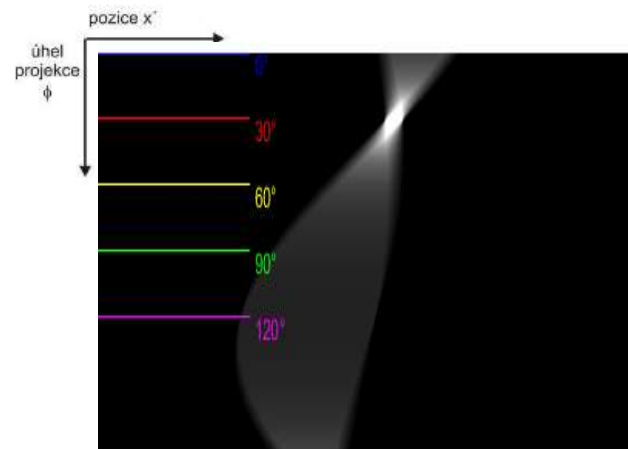


# Radonova transformace – použití

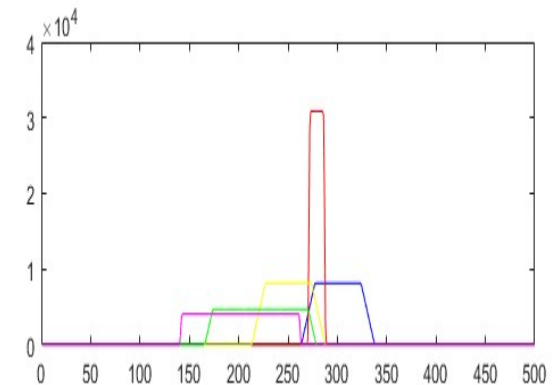
- detekce čárových úseků, určení orientace objektů na obraze



obrázek objektu s vyznačenými několika projekčními přímkami



sinogram



vybrané projekce

- k přímé Radonově transformaci dochází při snímání tomografem
  - paprskový integrál reprezentuje celkové zeslabení toku fotonů rtg záření měřený ve sloupci tkáně  $y'$  v daných pozicích osy  $x'$
- výpočetní tomografie (CT) má za úkol zrekonstruovat ze zpravidla několika stovek projekcí v jedné rovině - plošný řez vyšetřovaným objektem
  - při prosté zpětné Radonově transformaci (součet zpětných promítání jednotlivých projekcí) vzniká hvězdicový artefakt, proto se používá tzv. metoda filtrované projekce (inverzní Radonova transformace)
  - Fourierova rekonstrukce, algebraická rekonstrukce

# Detekce geometrických primitiv

Ilona Janáková



---

Rozvrh přednášky:

1. Segmentační metody.
2. Metody orientované na regiony.
3. Znalostní metody.
4. Hybridní metody.
- 5. Segmentace textur.**
6. Segmentace – barevná hloubka obrazu.
7. Vyplňování oblastí, indexace oblastí.

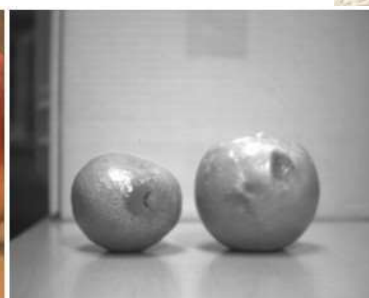
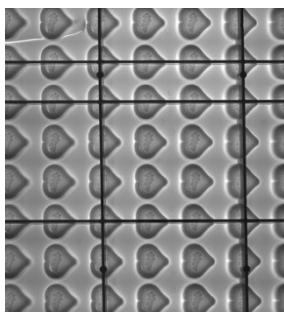
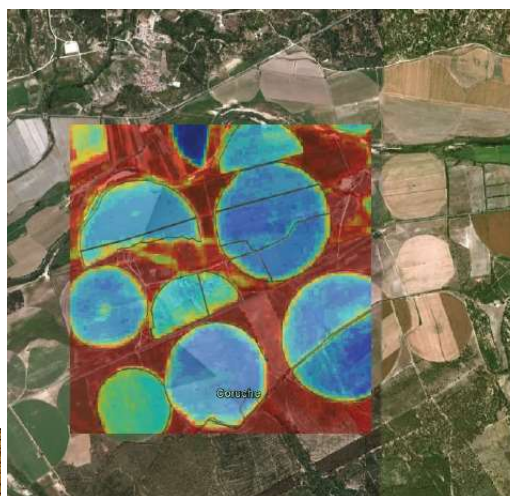
# Segmentace textur

## ▮ Vlastnosti

- opakující se struktura (tonální variace)
- má výrazné statistické vlastnosti
- má vlastní a nevlastní složku (nevlastní – nerovnost povrchu, osvětlení)
- může se vyskytovat ve více úrovních rozlišení

## ▮ Motivace

- kontrola textilií
- kvalita potisku
- dálkový průzkum Země
- povrchové vady dřevěných desek
- kontrola jakosti potravin



# Segmentace textur - metody

## ▮ Prahování

- tón a kontrast textury (z histogramu) – statistika 1. řádu => nepostihuje prostorovou strukturu

## ▮ Segmentace z obrazu hran

- prostorové vlastnosti textur – hustota hran, směr hran
- kombinace s jinými metodami (např. znalostní – porovnání obrazů hran)
- u jednoduchých textur např. typu šrafování může být využita houghova transformace

## ▮ Metody orientované na regiony

- spojování oblastí – definice kriteria homogenity z vlastností textury
- shluková analýza – textura popsána vektorem jejich vlastností

## ▮ Znalostní metody

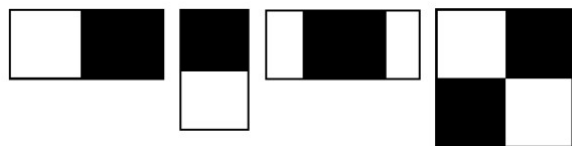
- vzorek textury, se kterým je prohledáván (korelován) obraz

## ▮ Hybridní metody

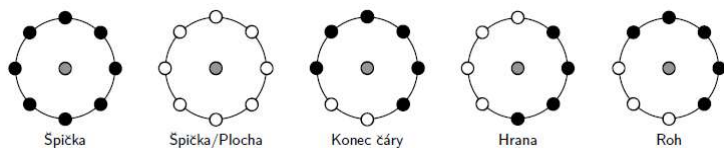
- neuronové sítě – textura popsána vektorem jejich vlastností

# Segmentace textur – popis vlastností textury

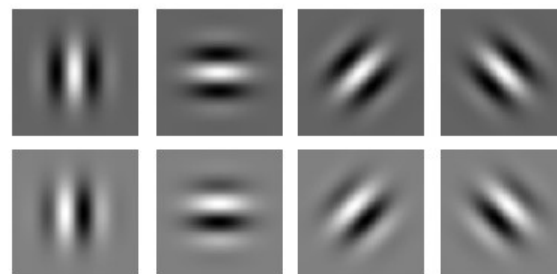
- tón a kontrast (z histogramu) - statistika 1. řádu, nepostihuje prostorovou strukturu
- výkonové spektrum - charakteristické frekvence a směrovost spektra energie ve fourierově rovině
- autokorelační funkce - velikost primitiva a periodicitu
- haralickovy texturní příznaky – energie, entropie, kontrast, homogenita, korelace, ...
- aplikace morfologických operací pro měření zrnitosti textury – funkcí pro různě velká jádra
- histogram kookurence - jak často se v textuře vyskytuje stejná konfigurace pixelů vzdálených o vektor  $(p, q)$  nebo určené poloměrem  $r$  (směrová nebo rotačně invariantní)
- lokální momenty obrazové funkce - konvoluce
- gáborovská analýza – obraz je filtrován bankou filtrů, které jsou popsány Gáborovskou funkcí
- haarovy příznaky – extrahují informace o lokální frekvenci, jednoduchá šablona, mnoho kombinací
- histogram LBP (*local binary pattern*) – pro každý bod obrazu je pomocí prahování a váhování bodů v jeho okolí vytvořen LBP kód, texturním příznakem je pak histogram LBP kódů všech bodů
- ...



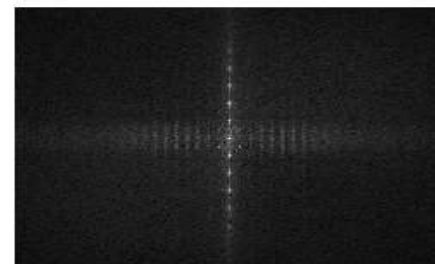
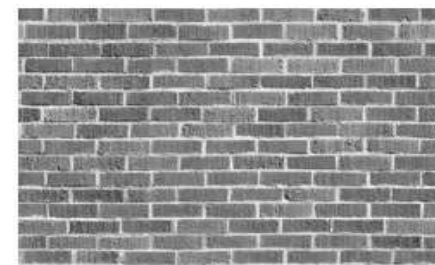
příklady haarových příznaků



primitiva LBP



příklady jader vygenerovaných pomocí gáborovské funkce



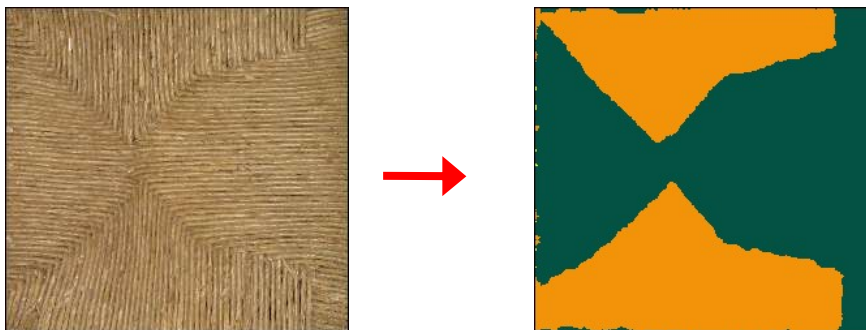
výkonové spektrum



# Segmentace textur - příklady

## Metoda k-means

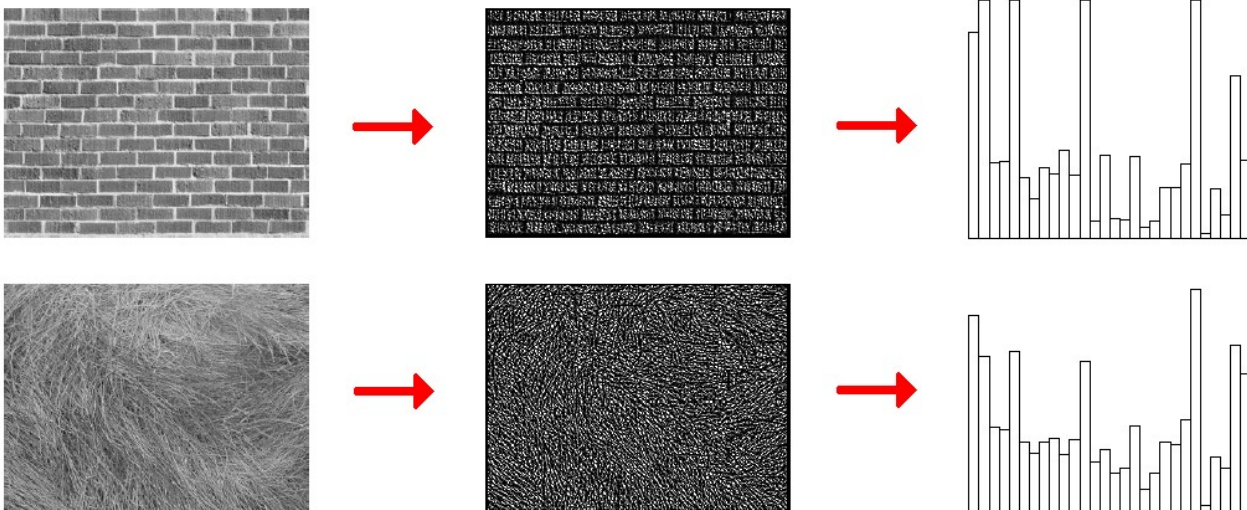
- příznakový vektor zkonstruován z gradientu - vektory jsou dvojdímenzionální, jednotlivé složky jsou obrazové derivace ve směru řádků a sloupců
- dvě textury = dva shluky



převzato z: [people.ciirc.cvut.cz/~hlavac/TeachPresEn/15ImageAnalysis/32-04SegmentationSpatialCoherence.pdf](http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/15ImageAnalysis/32-04SegmentationSpatialCoherence.pdf)

## LBP příznaky

- texturním příznakem není hodnota LBP jednotlivých pixelů, ale histogram hodnot lokálního okolí
- segmentace – porovnáním histogramů textur



převzato z: [http://www.fit.vutbr.cz/study/courses/IKR/public/stare\\_prednasky\\_2012/04\\_obrazove\\_priznaky/ikr-obrazove-priznaky-2012.pdf](http://www.fit.vutbr.cz/study/courses/IKR/public/stare_prednasky_2012/04_obrazove_priznaky/ikr-obrazove-priznaky-2012.pdf)

# Detekce geometrických primitiv

Ilona Janáková



---

Rozvrh přednášky:

1. Segmentační metody.
2. Metody orientované na regiony.
3. Znalostní metody.
4. Hybridní metody.
5. Segmentace textur.
- 6. Segmentace – barevná hloubka obrazu.**
7. Vyplňování oblastí, indexace oblastí.

# Segmentace – barevná hloubka obrazu

## ► Binární

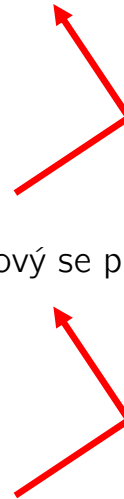
- výhodou je snadnější vyhodnocení
- nevýhodou je nižší přesnost, ztráta podstatných informací

## ► Šedotónový

- nejčastěji 256 úrovní jasu
- náročnější a delší výpočty, ale dosahuje se vyšší přesnosti
- často se pracuje s kombinací binární+šedotónový – šedotónový se převede na binární, kde se „hrubě“ nalezne tvar a v původním obrazu se provede upřesnění

## ► Barevný

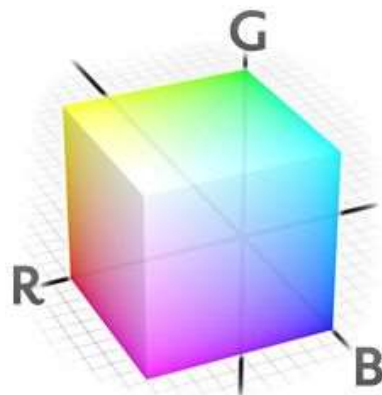
- nejčastěji 3 složky ve 256 úrovních jasu
- řada možností pro další zpřesnění
- kvalitnější segmentace hlavně pokud máme apriorní znalosti o barvě objektu
- nejnáročnější na paměť i výpočty
- Možné postupy:
  - převést na šedotónový obraz
  - vybrat pouze nejvýznamnější barevnou složku a s ní pracovat (šedotónový obraz)
  - rozdělit na jednotlivé složky a každou složku zpracovávat zvlášť
  - komplexní řešení ve všech složkách
  - převést na jiný výhodnější barevný model (RGB, HSV, YCbCr, ...), kde vyniknou požadované informace



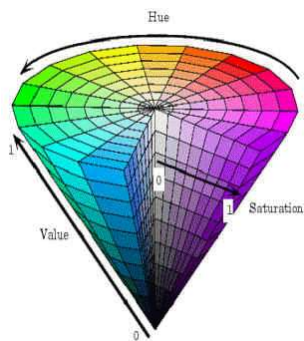
# Barevná segmentace – příklad segmentace barvy kůže



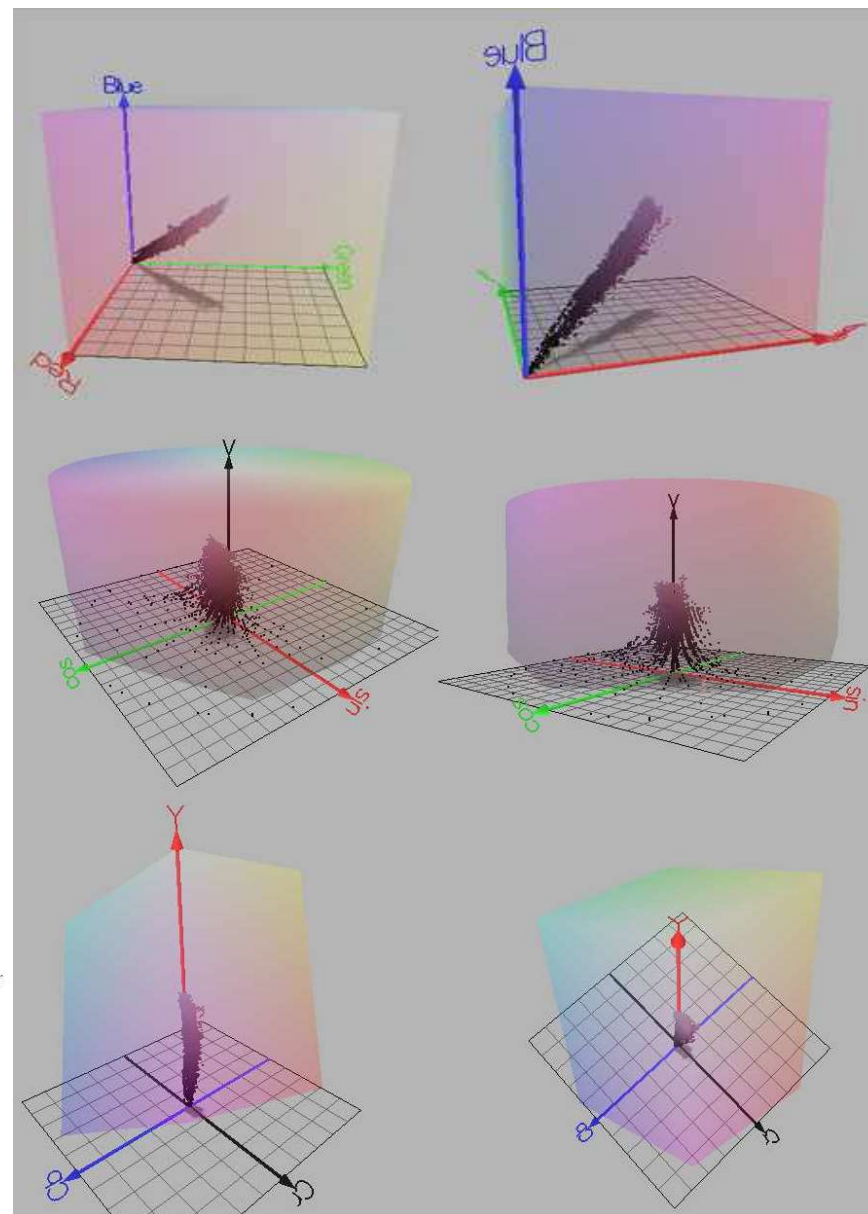
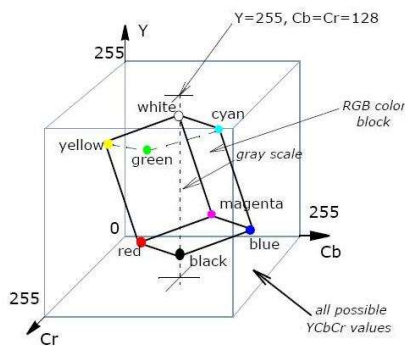
RGB



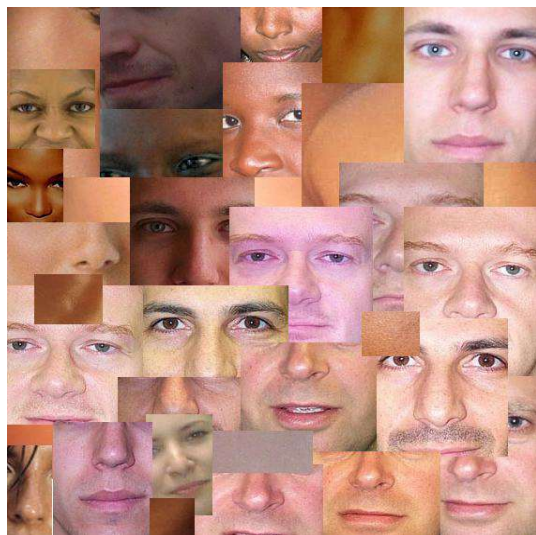
HSV



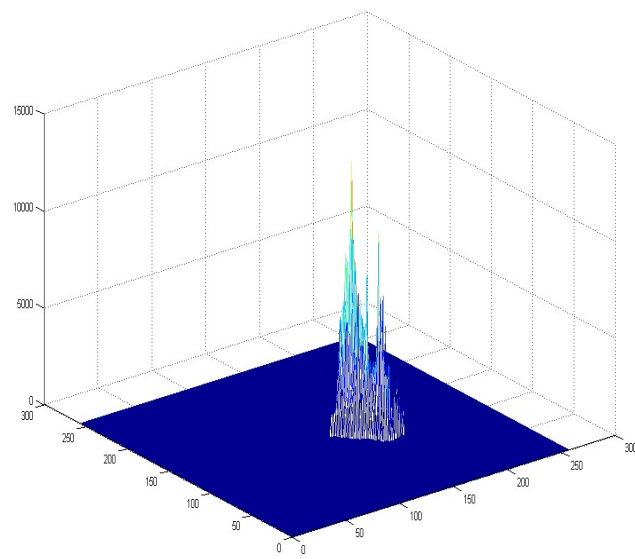
YCbCr



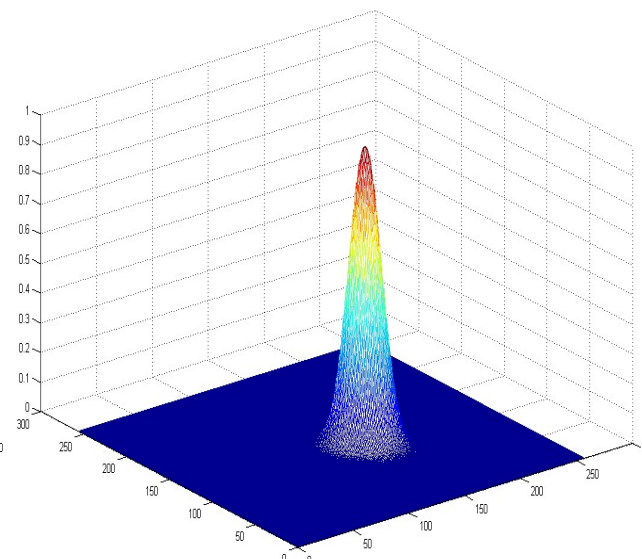
# Barevná segmentace – příklad segmentace barvy kůže



databáze obrazů kůže

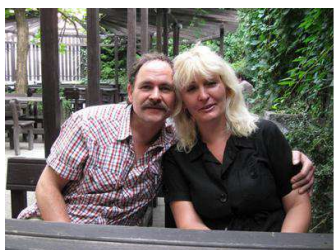
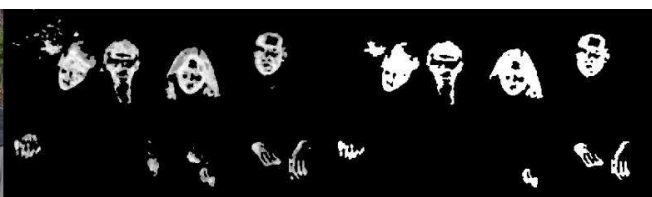


histogram výskytu pixelů  $C_b C_r$



histogram aproximovaný gaussovou křivkou

příklady – originální snímek (vlevo), detekovaná kůže (uprostřed), naprahaný obraz (vpravo)



# Transformace barev – omezení barevného prostoru

## ► Barevný → šedotónový

### ► z hlediska dalšího zpracování obrazu

- můžeme každé složce barevného obrazu přiřadit stejnou váhu

$$I_z = 0,33R + 0,33G + 0,33B$$

- můžeme pracovat pouze s jednou dominantní složkou
- převod do jiného barevného modelu

### ► z hlediska vnímání obrazu člověkem

- využívá se různé citlivosti oka na různé složky

$$I_o = 0,299R + 0,587G + 0,114B$$

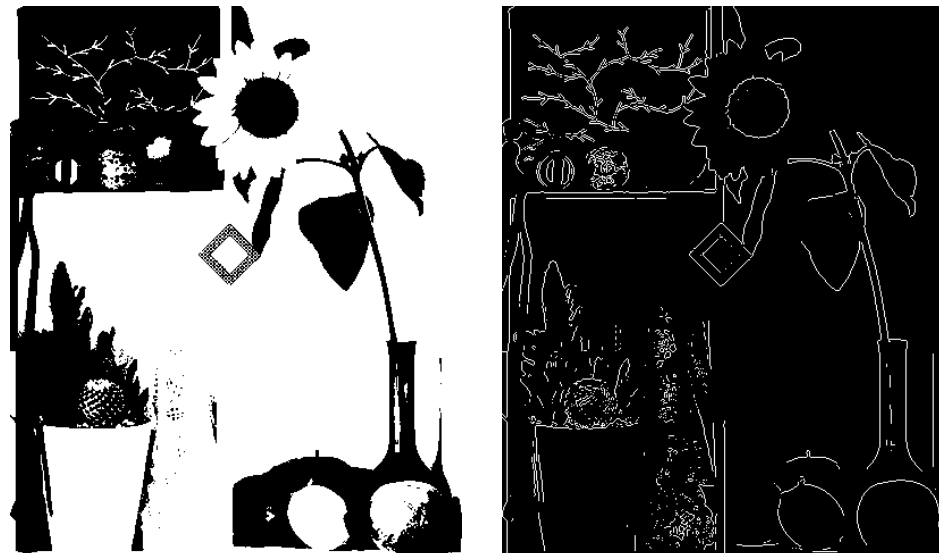


# Transformace barev – omezení barevného prostoru

## Šedotónový $\rightarrow$ binární

### z hlediska dalšího zpracování obrazu

- prahování – zvolení vhodného prahu vzhledem ke snímané scéně
- prahování aplikujeme buď na plochy nebo hrany
- prahování s více prahy



### z hlediska vnímání obrazu člověkem

- využívá se schopnosti oka skládat barvy – průměrovat  $\Rightarrow$  rozptylování

### a) náhodné rozptýlení

Algoritmus:

Cyklus přes všechny souřadnice obrazu  $x, y$ :

Pokud  $C_{in} > \text{random}(0:255)$ , tak  $C_{out} = 1$ , jinak  $C_{out} = 0$



# Transformace barev – omezení barevného prostoru

## Šedotónový → binární

### z hlediska vnímání obrazu člověkem

- využívá se schopnosti oka skládat barvy – průměrovat => rozptylování

### b) pravidelné (maticové) rozptýlení

- předem vytvořené vzory černých a bílých bodů, které budou zastupovat jednotlivé odstíny v originálním obrazu

#### b.1) zvětšení obrazu

- místo původního pixelu vložíme celou vybranou masku

Př.:

$$C_{in} = \begin{matrix} (0-50> & (50-100> & (100-150> & (150-200> & (200-256) \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \end{matrix}$$

### c) distribuce zaokrouhlovací chyby

- algoritmus Floyd-Steinberg, Burkes, Stucki

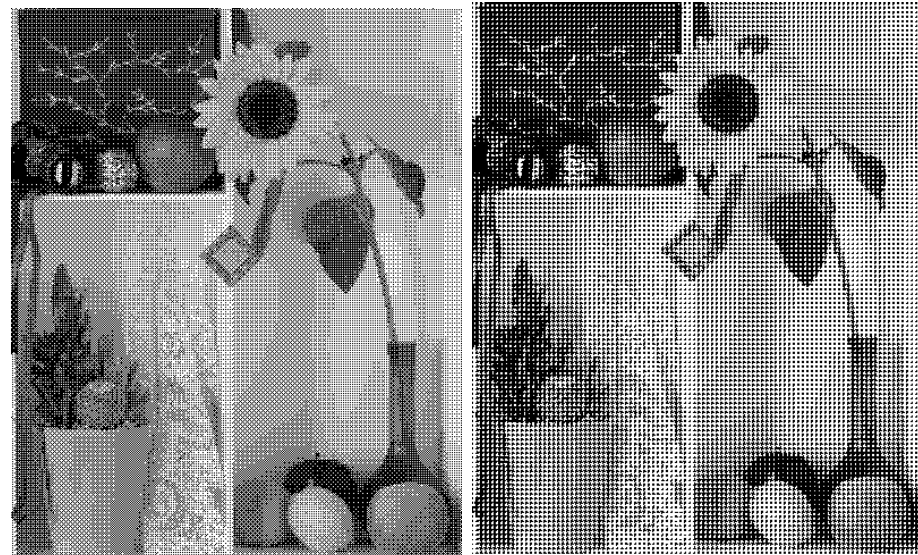
#### b.2) zachování velikosti obrazu

- místo původního pixelu vložíme jen odpovídající hodnotu masky

Algoritmus:

Cyklus přes všechny souřadnice obrazu  $x,y$ :

Pokud  $C_{in} > M[(X \bmod XmaskSize), (Y \bmod YmaskSize)]$ ,  
tak  $C_{out} = 1$ , jinak  $C_{out} = 0$



$$M_{display} = \begin{bmatrix} 0 & 12 & 3 & 15 \\ 8 & 4 & 11 & 7 \\ 2 & 14 & 1 & 13 \\ 10 & 6 & 9 & 5 \end{bmatrix}$$

$$M_{tisk} = \begin{bmatrix} 1 & 5 & 9 & 2 \\ 8 & 12 & 13 & 6 \\ 4 & 15 & 14 & 10 \\ 0 & 11 & 7 & 3 \end{bmatrix}$$



# Detekce geometrických primitiv

Ilona Janáková



---

Rozvrh přednášky:

1. Segmentační metody.
2. Metody orientované na regiony.
3. Znalostní metody.
4. Hybridní metody.
5. Segmentace textur.
6. Segmentace – barevná hloubka obrazu.
- 7. Vyplňování oblastí, indexace oblastí.**

# Vyplňování oblastí

- Definice oblasti souvisí s popisem její hranice

- **a) geometricky určená hranice**

- posloupnost bodů definující mnohoúhelník
- Metody:
  - řádkové vyplňování

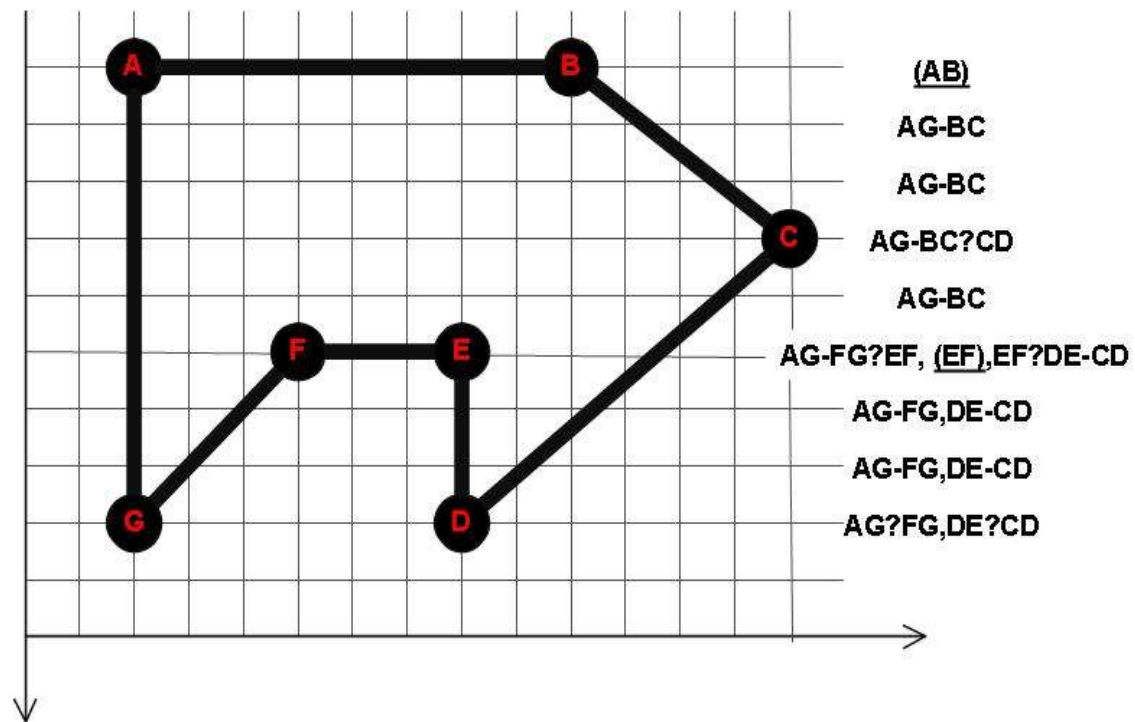
- **b) hranice nakreslená v rastru**

- hranice není jasně definována, všechny informace o oblasti se získávají pouze z obrazu (z obrazové = rastrové paměti)
  - hraniční vyplňování = bod je vnitřní, pokud má jinou barvu než hranice
  - záplavové vyplňování (lavinové, přebarvování) = bod je vnitřní, pokud má stejnou barvu jako semínko
  - měkké vyplňování = bod je vnitřní, pokud má výrazně jinou barvu než hranice (hranice byla vyhlazena a má nestejnou barvu)
- Metody:
  - semínkové vyplňování
    - jednoduché / řádkové
    - rekurzivní / se zásobníkem
  - indexace oblastí - barvení

# Vyplňování geometricky určené hranice

## ► Řádkové vyplňování

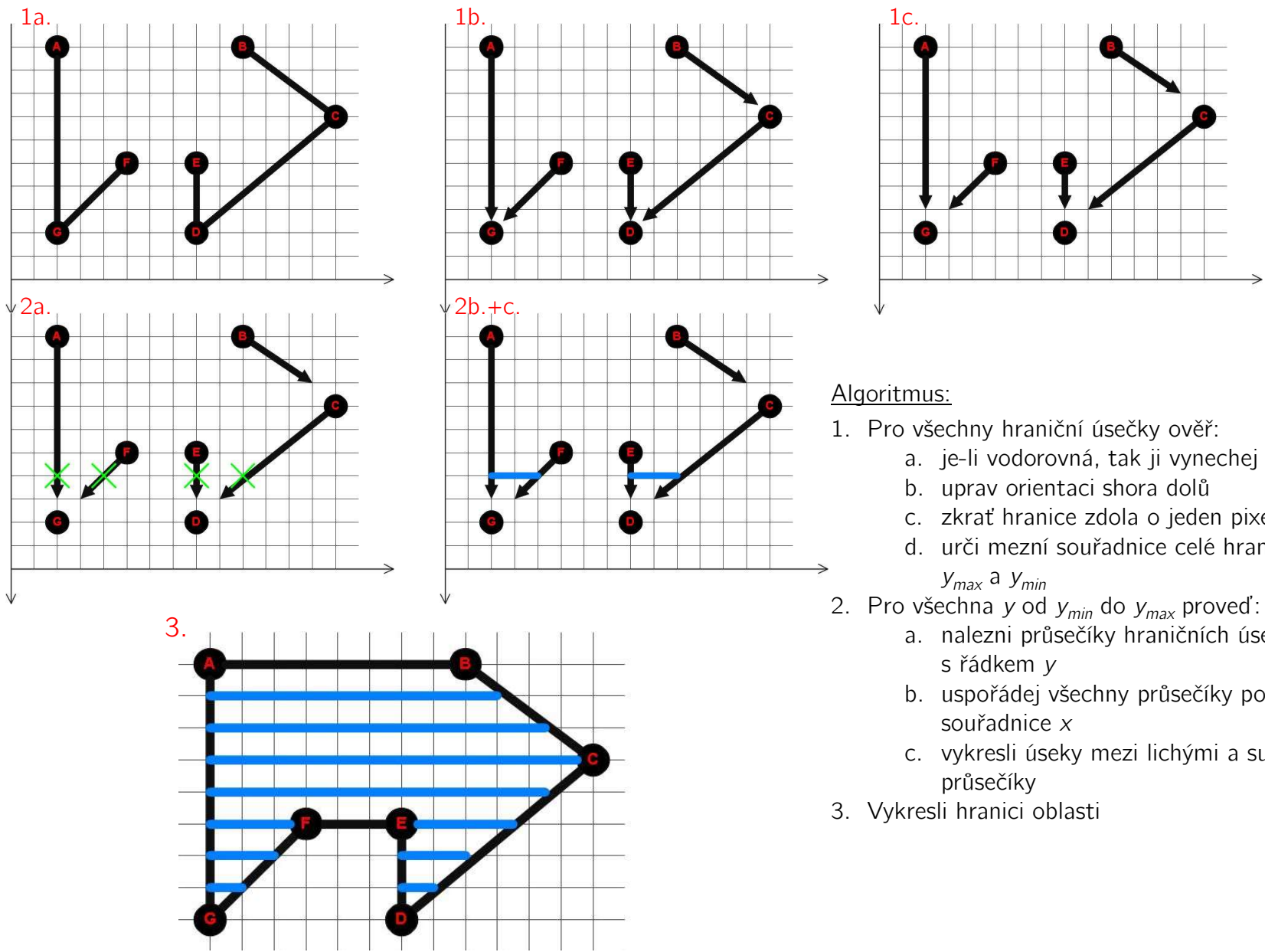
- každým řádkem je vedena pomyslná čára a jsou hledány její průsečíky s hranicí oblasti
- nalezené průsečíky se seřadí podle souřadnic  $x$  – dvojice průsečíků určuje úsečky ležící uvnitř oblasti



- problémy s vrcholy a vodorovnými hranicemi, proto vhodnější modifikace algoritmu



# Vyplňování geometricky určené hranice



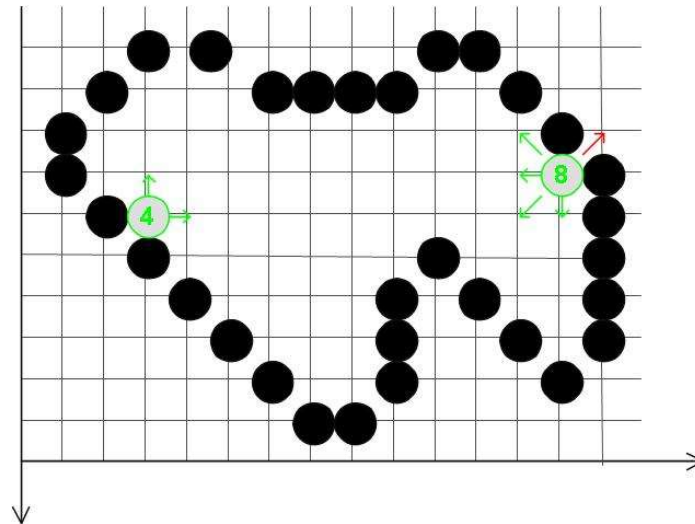
Algoritmus:

1. Pro všechny hraniční úsečky ověř:
  - a. je-li vodorovná, tak ji vynechej
  - b. uprav orientaci shora dolů
  - c. zkrat hranice zdola o jeden pixel
  - d. urči mezní souřadnice celé hranice  $y_{max}$  a  $y_{min}$
2. Pro všechna  $y$  od  $y_{min}$  do  $y_{max}$  proved' :
  - a. nalezni průsečíky hraničních úseček s řádkem  $y$
  - b. uspořádej všechny průsečíky podle souřadnice  $x$
  - c. vykresli úseky mezi lichými a sudými průsečíky
3. Vykresli hranici oblasti

# Vyplňování hranice nakreslené v rastru

## ▮ Semínkové vyplňování

- základem je semínko – libovolný vnitřní bod (určí uživatel nebo dané znalostmi o oblasti)
- od semínka se postupně rozšiřuje prohledávání obrazové paměti a nalezeným vnitřním bodům je přiřazena nová barva
- 4-spojité X 8-spojité oblast
- 4-spojité X 8-spojité hranice



## ▮ a) Jednoduché semínkové vyplňování

- rekurzivní algoritmus – šíření semínka do všech stran
- => prakticky nepoužitelné – každý pixel je testován několikrát, mnohonásobné zanoření rekurze - možné řešit pomocí zásobníku

Algoritmus – rekurze, čtyřokolí:

Seminko(x,y)

Je-li bod [x,y] vnitřní, pak:

1. obarví bod [x,y] požadovanou barvou
2. proved' Seminko(x+1,y)
3. proved' Seminko(x-1,y)
4. proved' Seminko(x,y+1)
5. proved' Seminko(x,y-1)

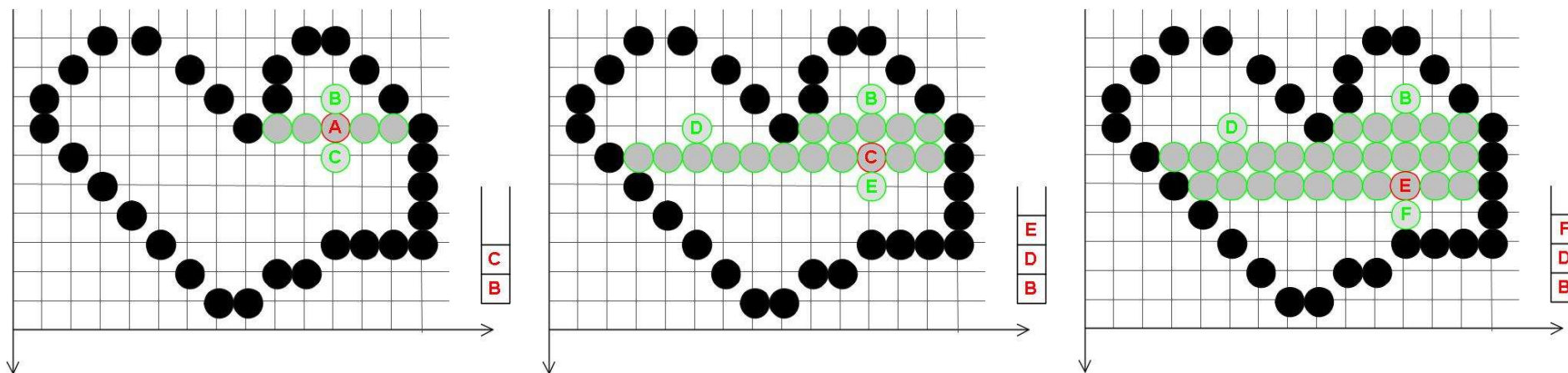
# Vyplňování hranice nakreslené v rastru

## ► b) Řádkové semínkové vyplňování

- vyhledávání sousedních bodů je prováděno v jednom řádku pixelů (úsek), pouze bezprostředně nad ním a pod ním má smysl hledat nová semínka

Algoritmus - zásobník:

- Vlož první semínko do zásobníku
- Dokud není zásobník prázdný, opakuj:
  - vyjmi semínko se souřadnicemi  $[x,y]$  ze zásobníku
  - nalezni hranici  $x_L$  a  $x_R$  na řádku  $y$  v nejbližším okolí bodu  $[x,y]$
  - nakresli úsečku  $[x_L,y], [y,x_R]$
  - na vyšší úsečce  $[x_L,y-1], [x_R,y-1]$  hledej souvislé nevyplněné vnitřní úseky a pro každý z nich vlož do zásobníku souřadnici jednoho vnitřního bodu
  - na nižší úsečce  $[x_L,y+1], [x_R,y+1]$  hledej souvislé nevyplněné vnitřní úseky a pro každý z nich vlož do zásobníku souřadnici jednoho vnitřního bodu



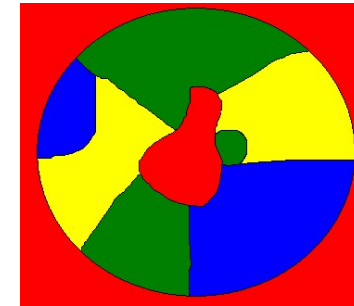
# Indexace oblastí - barvení

## ► Obvyklá metoda - 1 až N barev

- každou samostatnou souvislou oblast opatříme neopakujícím se přirozeným číslem – pozadí má číslo 0, oblastem jsou přiřazena čísla postupně od jedničky. Největší identifikační číslo oblasti tedy udává počet oblastí v obraze.

## ► Jiná možnost - 4 barvy

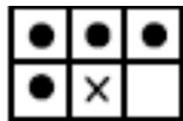
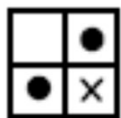
- použijeme menší počet identifikačních čísel - pouze zajistíme, aby žádné dvě sousední oblasti neměly stejné identifikační číslo. Teoreticky stačí čtyři barvy (resp. čísla). Pro identifikaci oblastí je pak třeba mít pro každou oblast uloženou informaci o poloze některého jejího bodu.



Algoritmus – 1 až N barev:

**1.** procházíme obraz po řádcích a každému nenulovému obrazovému elementu přiřadíme hodnotu podle hodnoty všech jeho již obarvených sousedů

- jsou-li všechny nulové, přiřadíme bodu dosud nepřidělenou barvu
- pokud je jeden nenulový, nebo je více nenulových, ale se stejnou barvou, přiřadíme bodu tuto jeho/jejich barvu
- pokud je více nenulových s různou barvou, přiřadíme bodu jednu z těchto barev a zaznamenáme barvy do tzv. **tabulky ekvivalence barev** (došlo k tzv. *kolizi barev*)



sledované pixely pro 4-okolí a pro 8-okolí

příklady objektů, kdy dojde ke kolizi



0	0	1	0	2
0	0	1	0	2
0	0	1	0	2
0	3	? <sub>1</sub>	1	?

kolize barev (4-okolí)

ekvivalence barev

1-3

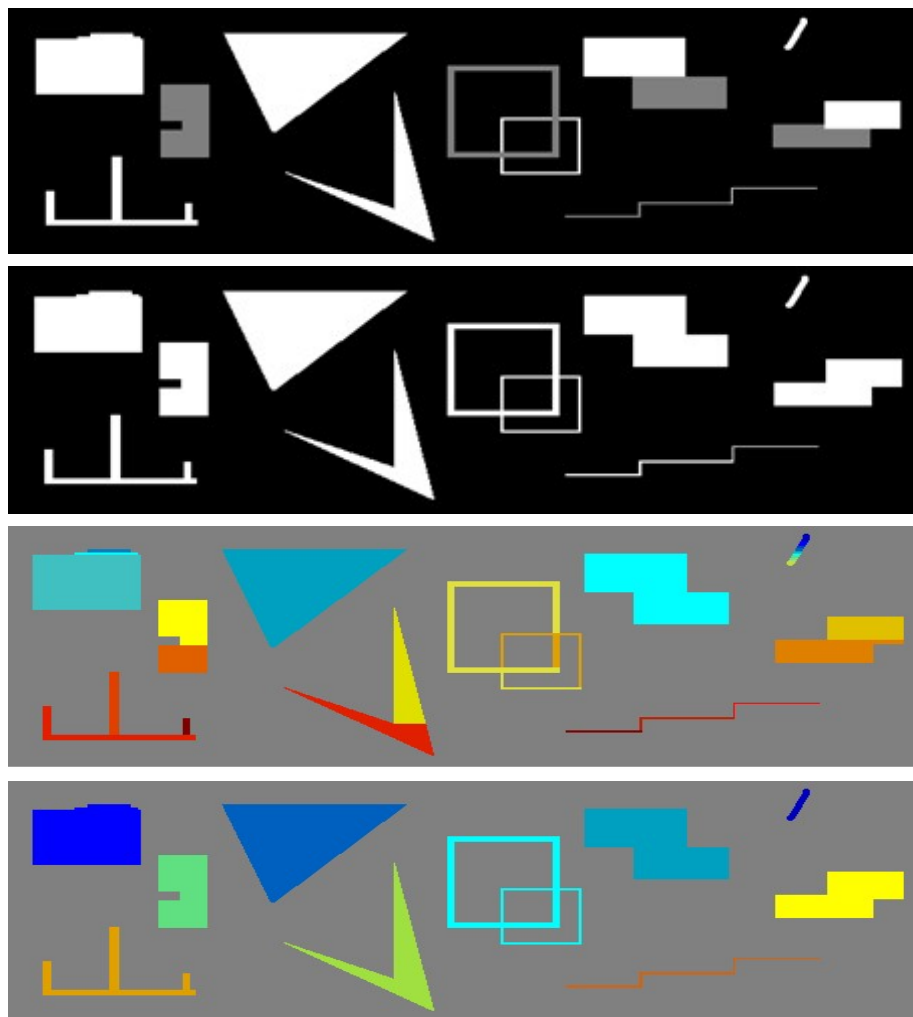
1-2

# Indexace oblastí - barvení

Algoritmus – 1 až N barev:

**2.** projdeme znovu celý obraz po řádcích a přebarvíme obrazové body kolizních barev podle tabulky ekvivalence barev

Př.: kolize barev 7 a 9 = všechny pixely s barvou 9 přebarvíme na 7 (počet oblastí bude od zjištěného čísla po prvním průchodu o jednu nižší). Chceme-li aby žádná barva nebyla vynechána (barva 9), musíme zbytek oblastí přechíslovat (přebarvit).



Barvení:

- původní obraz
- naprahovaný obraz
- první průchod
- druhý průchod



# Zdroje - literatura

## Segmentace:

Hlaváč, V., Šonka, M.: *Počítačové vidění*, Grada Praha, 1992, ISBN 80-85424-67-3.

Španěl, M., Beran, V. Obrazové segmentační techniky <http://www.fit.vutbr.cz/~spanel/segmentace/>

## Watershed:

<http://cmm.enscm.fr/~beucher/wtshed.html>

[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/KIM1/Seminars/Watershed/index.htm](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/KIM1/Seminars/Watershed/index.htm)

Beucher S.: *Image Segmentation and Mathematical Morphology*. Center of Mathematical Morphology home page, 2000.

## Shluková analýza, Mean-shift:

Kelbel, J., Šilhán, D. Shlukova analyza [cmp.felk.cvut.cz/cmp/courses/recognition/zapis\\_prednasky/zapis\\_02/13/shlukovani.pdf](http://cmp.felk.cvut.cz/cmp/courses/recognition/zapis_prednasky/zapis_02/13/shlukovani.pdf)

Doubek, P. Mean-Shift segmentace <http://cmp.felk.cvut.cz/cmp/courses/ZS1/Cviceni/cv4/meanshift.pdf>

## Active Appearance Models:

Cootes, T.F., Edwards, G.J., Taylor, Ch.J. Active Appearance Models, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, june 2001

Stegmann, M.B. Active Appearance Models <http://www2.imm.dtu.dk/~aam/>

## Neuronové sítě:

Rayes-Aldasoro, C.C. Image Segmentation with Kohonen Neural Network Self-Organising Maps.

<http://www.cs.jhu.edu/~cis/cista/446/papers/SegmentationWithSOM.pdf>

Hradiš, M. Konvoluční neuronové sítě <https://openalt.cz/2015/data/Michal%20Hradis%20-%20Konvolucni%20neuronove%20site.pdf>

Deep Learning v prostředí Matlab: systémy strojového vidění; identifikace zboží a osob. AUTOMA [online]. Humusoft s. r. o, 2017, (5), 12-14 [cit. 2018-01-02]. Dostupné z: [http://www.automa.cz/Aton/FileRepository/pdf\\_articles/10482.pdf](http://www.automa.cz/Aton/FileRepository/pdf_articles/10482.pdf)

CHILAMKURTHY, Sasank. A 2017 Guide to Semantic Segmentation with Deep Learning [online]. July 5, 2017 [cit. 2018-05-09]. Dostupné z:

<http://blog.quare.ai/notes/semantic-segmentation-deep-learning-review>

## Textura:

Šára, R. Analýza textury [http://cmp.felk.cvut.cz/cmp/courses/dzo/resources/lecture\\_texture\\_sara.pdf](http://cmp.felk.cvut.cz/cmp/courses/dzo/resources/lecture_texture_sara.pdf)

Láník, A., Zuzanařák, J. Extrakce obrazových příznaků.

[http://www.fit.vutbr.cz/study/courses/IKR/public/stare\\_prednasky\\_2012/04\\_obrazove\\_priznaky/ikr-obrazove-priznaky-2012.pdf](http://www.fit.vutbr.cz/study/courses/IKR/public/stare_prednasky_2012/04_obrazove_priznaky/ikr-obrazove-priznaky-2012.pdf)

## Vyplňování oblastí:

Žára, J., Beneš, B., Felkel, P.: *Moderní počítačová grafika*, Computer Press Praha, 1998, ISBN 80-7226-049-9

Pelikán, J. Vypĺňování souvislé oblasti/Vypĺňování N-úhelníka <https://cgg.mff.cuni.cz/~pepca/lectures/npgr003.slides.cz.php>