

# Mean-Shift segmentace

Petr Dousek  
CMP FEL ČVUT Praha

29. října 2007

Mean-shift segmentace [1] shlukuje body obrazu (pixely) na základě podobnosti jejich vzhledu a blízkosti jejich pozice pomocí konvergence do lokálních maxim spojeného souřadnicového a intenzitního prostoru.

## 1 Formulace problému

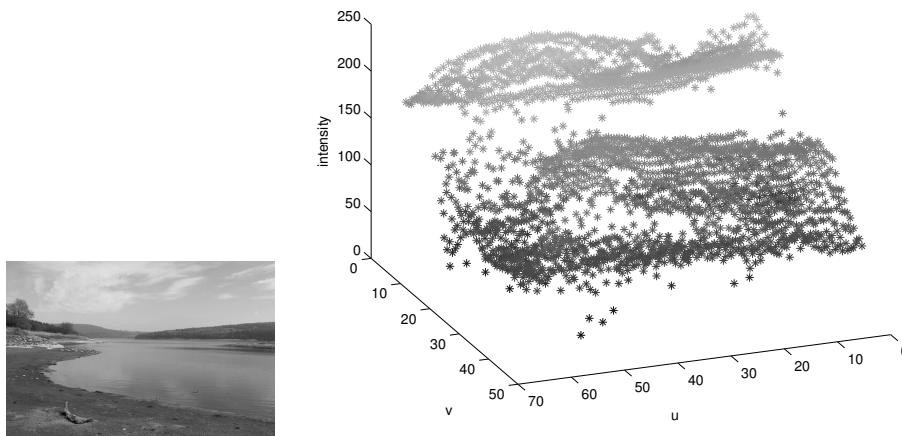
Pro jednoduchost uvažujme šedotónový obraz: každý pixel  $[u, v]$  má danou intenzitu  $i$ . Obraz můžeme reprezentovat nejen jako dvourozměrnou matici intenzit, ale také jako body  $[u, v, i]$  v třírozměrném prostoru, kde první dvě souřadnice udávají pozici a třetí intenzitu, viz obr. 1. Blízké pixely o podobné intenzitě, o nichž předpokládáme, že náležejí stejnému objektu, budou v prostoru  $[u, v, i]$  vytvářet shluky. Naopak pixely výrazně rozdílné intenzity, které spolu v obraze sousedí a přesto pravděpodobně náležejí různým objektům, budou v prostoru  $[u, v, i]$  vzdálené. Tímto převádíme úlohu segmentace na *shlukování* bodů v  $d$ -rozměrném prostoru, pro šedotónový obraz  $d = 3$ , pro barevný  $d = 5$  (přidáme další dva barevné kanály jako souřadnice).

## 2 Použití mean-shift algoritmu

Mean-shift je obecný algoritmus pro iterativní nalezení lokálního maxima hustoty *vzorků* (v našem případě budou vzorky pixely  $\mathbf{x} = [u, v, i]$ ). Jméno vystihuje princip: v každém kroku se posuneme ve směru váženého průměru vzorků (těžiště) z oblasti kolem současné pozice<sup>1</sup>. Při použití pro shlukování postupujeme ve dvou krocích:

---

<sup>1</sup>pro lepší představu si prohlédněte grafické znázornění konvergence např. v textu [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/TUZEL1/MeanShift.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TUZEL1/MeanShift.pdf)



Obrázek 1: Spojený souřadnicový a intenzitní prostor, všimněte si např. jasně odděleného shluku bodů odpovídajícího obloze.

1. Z každého vzorku spustíme mean-shift a zapamatujeme si lokální maximum, do něhož dokonvergoval.
2. Shlukujeme ty vzorky, které dokonvergovaly do stejného maxima (s určitou tolerancí) – oblasti, z které vzorky konvergují do stejného bodu, říkáme *basin of attraction*.

### 3 Mean-shift

Mean-shift počítá v každém kroku počítá gradient hustoty vzorků v současné pozici  $\mathbf{x}$  ze vzorků v okolí, které je definované *kernelem*  $K$  – radiálně symetrickou funkcí. Často je výhodnější ve vzorcích použít *profil kernelu*  $k$

$$K(\mathbf{x}) = c_k k(\|\mathbf{x}\|^2) . \quad (1)$$

My použijeme Epanečnikův kernel:

$$K_E(\mathbf{x}) = \begin{cases} c(1 - \|\mathbf{x}\|^2) & \text{pro } \|\mathbf{x}\| \leq 1 , \\ 0 & \text{jinak .} \end{cases} \quad (2)$$

Odvoďte si profil kernelu  $k_E$  a jeho derivaci  $k'_E$ . Pro obecný kernel je odhad hustoty v bodě  $\mathbf{x}$

$$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n K \left( \frac{\mathbf{x} - \mathbf{x}_i}{h} \right), \quad (3)$$

kde  $h$  je šířka (*bandwidth*) kernelu a  $\mathbf{x}_1 \dots \mathbf{x}_n$  jsou vzorky. My ale nepotřebujeme hustotu, pouze její gradient, ten získáme dosazením (1) do (3) a derivací:

$$\nabla f(\mathbf{x}) = \frac{2c_k}{nh^d} \sum_{i=1}^n (\mathbf{x} - \mathbf{x}_i) k' \left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right). \quad (4)$$

Spočítejte si gradient pro Epanečnikův kernel  $f_E$  dosazením  $k'_E$ , zvolte  $c_k = \frac{1}{2}$ . Mean-shift algoritmus inicializovaný v pozici  $\mathbf{y}_0$  proběhne následovně:

```
do
   $\mathbf{y}_{i+1} \leftarrow \mathbf{y}_i + \nabla f(\mathbf{y}_i)$ 
   $i \leftarrow i + 1$ 
until  $\|\mathbf{y}_{i+1} - \mathbf{y}_i\| < \frac{h}{1000}$ 
```

Pro aplikaci v prostoru s různými veličinami na různých osách je vhodné použít kernel s různou šířkou pro každou veličinu. Konkrétně pro segmentaci obrazu je výhodné mít možnost měnit nezávisle šířku kernelu  $h_s$  ve směru pozice pixelu a šířku  $h_r$  ve směru intenzity, tj. místo vektoru  $[u, v, i]$  použít normalizovaný vektor  $[\frac{u}{h_s}, \frac{v}{h_s}, \frac{i}{h_r}]$ . Při použití tohoto vektoru pak bude v (4)  $h = 1$ . Hodnoty  $h_s$  a  $h_r$  jsou jediné dva parametry této metody.

## 4 Shlukování pixelů

Spuštěním mean-shift v každém pixelu dostaneme pro každý pixel lokální maximum  $\mathbf{z} = [u, v, i]$  a chceme spojovat takové sousední pixely  $a, b$ , pro něž

$$\text{abs}(\mathbf{z}_a - \mathbf{z}_b) \leq \mathbf{h} = [h_s, h_s, h_r] \quad (5)$$

ve všech souřadnicích. Ke spojení použijeme techniku *supergrid* [2] – vytvoříme binární matici o dvojnásobné délce stran, než má obraz, v němž budou reprezentovány jak pixely 1, tak hrany mezi nimi ?, viz příklad pro 4-spojité obraz

```
1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ?
? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0 ? 0
1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ? 1 ?
```

kde hrana  $\tau$  nabývá hodnoty 1, pokud je splněná podmínka (5), jinak má hodnotu 0. V této binární matici lze nalézt spojené komponenty např. funkcí `bwlabel()` v Matlabu.

## 5 Poznámky k implementaci v Matlabu

Mean-shift segmentace je poměrně časově náročná, doporučujeme zejména během vývoje pracovat se zmenšeným obrazem. Pokuste se o implementaci, která nebude zbytečně opakovat stejné operace. Šířku kernelu  $h_s, h_r$  pro čtvrtinový obrázek o rozsahu intenzit 0-255 zkuste volit mezi 10 a 30.

Ve výsledku zůstanou malé několikapixelové komponenty vzniklé typicky z pixelů zatížených velkým šumem. Pro konečnou úpravu můžete k jejich spojení s většími sousedními komponentami využít funkci `remsmall()` [3], pro testování během vývoje ale zbytečně zdržuje.

Pro shlukování pixelů obrazu  $m \times n$ , jejichž souřadnice i intenzity jsou uloženy v matici  $\mathbf{z}_{m \times n \times d}$  technikou supergrid můžete použít následující kód, jehož výsledkem bude označení pixelů číslem komponenty v matici  $\mathbf{l}_{m \times n}$ :

```
s = ones( 2*m+1, 2*n+1, 'int8' );
s(1:2:(2*m+1),:) = zeros( m+1, 2*n+1, 'int8' );
s(:,1:2:(2*n+1)) = zeros( 2*m+1, n+1, 'int8' );
s(2:2:2*m,3:2:(2*n-1)) = all(cat(3, ... % horizontal edges
    abs(z(:,2:end,1:2)-z(:,1:(end-1),1:2)) < hs, ...
    abs(z(:,2:end,3:end)-z(:,1:(end-1),3:end)) < hr ),3);
s(3:2:(2*m-1),2:2:2*n) = all(cat(3, ... % vertical edges
    abs(z(2:end,:,1:2)-z(1:(end-1),:,1:2)) < hs, ...
    abs(z(2:end,:,3:end)-z(1:(end-1),:,3:end)) < hr ),3);
l = bwlabel(s,4); % find connected regions
l = l(2:2:2*m, 2:2:2*n); % extract labeling
```

Pro zobrazení komponent barevně doporučujeme funkci `label2rgb()`.

## Reference

- [1] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, pages 603–619, 2002.
- [2] Milan Šonka, Václav Hlaváč, and Roger Boyle. *Image Processing, Analysis and Machine Vision*. Thomson, 3rd edition, 2007.
- [3] Tomáš Svoboda, Jan Kybic, and Václav Hlaváč. *Image Processing, Analysis and Machine Vision. A MATLAB Companion*. Thomson, 2007.