

Počítačové videnie - Hľadanie objektov, Pyramídy

Ing. Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

24.10.2018

Korelácia

Korelácia definícia - spojitá pre reálne hodnoty

$$J = I * M \iff J(\chi, \psi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) M(x + \chi, y + \psi) dx dy$$

Korelácia

Korelácia definícia - diskretná pre reálne hodnoty

$$J = I * M \iff J(r, c) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} I(u, v) M(r + u, c + v)$$

Korelácia definícia - diskretná pre reálne hodnoty

Pre príklad obrazov predpokladáme, že I a M majú nulové hodnoty všade kde je index mimo rozmerov obrazu.

Korelácia - matlab

xcorr2

`xcorr2(I,M)` - vráti $I * M$, keďže počítače neoperujú s nekonečnými poliami, tak matlab vráti 2D pole so šírkou $= \text{šírka}(I) + \text{šírka}(M) - 1$. Obdobne je to s výškou.

Úloha

Teraz budeme hľadať kde na obrázku Einsteina sa nachádzajú oči. Otestujte koreláciu obrázku Einsteina s jeho okom. Zobrazte si výsledný obraz.

Korelácia - matlab

Korelácia s nulovým priemerom definícia

$$J_0(r, c) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} (I(u, v) - \bar{I}) (M(r + u, c + v) - \bar{M})$$

Úloha

Otestujte koreláciu s nulovým priemerom na obrázku einstein.jpg a potom na einstein2.jpg.

Korelácia - matlab

Korelácia s nulovým priemerom definícia

$$J_0(r, c) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} (I(u, v) - \bar{I}) (M(r + u, c + v) - \bar{M})$$

Úloha

Otestujte koreláciu s nulovým priemerom na obrázku einstein.jpg a potom na einstein2.jpg.

Hint

V matlabe máme broadcasting, t.j: $J_0 = (I - \bar{I}) * (M - \bar{M})$

Normalizovaná korelácia

Normalizovaná korelácia definícia

$$J(r, c) = \frac{\sum_u \sum_v (I(u, v) - \bar{I}) (M(r + u, c + v) - \bar{M})}{\sqrt{\sum_u \sum_v (I(u, v) - \bar{I})^2 \sum_u \sum_v (M(r + u, c + v) - \bar{M})^2}}$$

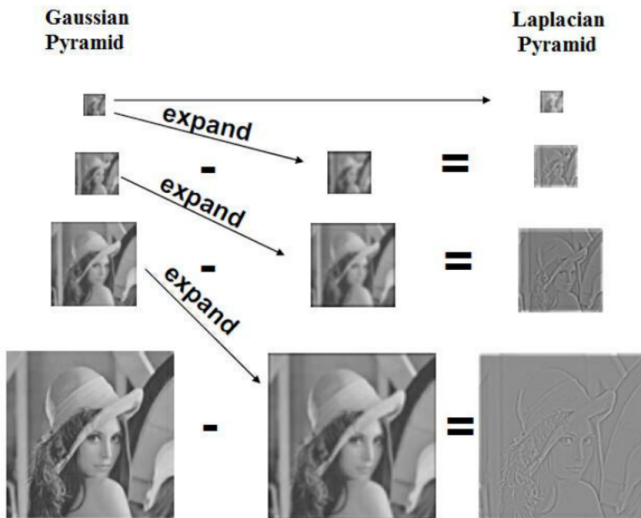
normxcorr2

normxcorr2(template, I) - vráti normalizovanú koreláciu obrazu I pomocou vzoru template (to sme doteraz mali ako M)

Úloha

Otestujte normalizovanú koreláciu obrázku einstein.jpg a potom na einstein2.jpg.

Pyramídy



Nástroje

Toolkit

Prepnite sa do zložky pyramidy zo zipu k dnešnému cviku. V nej sú funkcie z pyramídového toolkitu, ktoré budeme používať.

genPyr

$P = \text{genPyr}(I, \text{typ}, \text{level})$ - vygeneruje cell štruktúru v ktorej jednotlivé bunky ($P_1, P_2 \dots P_{\text{level}}$) su pyramídy až po zadaný level, typ môže byť buď 'gauss', alebo 'laplace'.

Úloha

Vygenerujte si oba druhy pyramíd pre nejaký obrázok a porovnajte ich. Pozrite sa na rozdiel medzi poslednou úrovňou z Laplacovej pyramídy a ostatnými.

Nástroje

pyrReconstruct

`pyrReconstruct(P)` - z Laplacovskej pyramídy `P` vytvorí pôvodný obrázok, `P` musí mať rovnaký formát ako výstup z `genPyr`.

Úloha

Vytvorte si Laplacovskú pyramídu pre `carpet.jpg`, alebo iný obrázok. Zmeňte niektoré úrovne na nulu a zrekonštruujte si obraz.

Úloha blending

Blending

- Načítajte obrázky eye.jpg, carpet.jpg a mask.png
- Vytvorte si Laplacovské pyramídy LP_e a LP_c z obrázkov.
- Vytvorte Gaussovskú pyramídu masky G_m .
- Vytvorte Laplacovskú pyramídu t.ž. pre každý level
$$L^l = G_m \cdot LP_e + (1 - G_m) \cdot LP_c$$
- Rekonštruujte obraz z L.
- Porovnajte rekonštruovaný obraz s obrazom
$$S = M \cdot I_e + (1 - M) \cdot I_c$$

Úloha blending

Blending

- Načítajte obrázky eye.jpg, carpet.jpg a mask.png
- Vytvorte si Laplacovské pyramídy LP_e a LP_c z obrázkov.
- Vytvorte Gaussovskú pyramídu masky G_m .
- Vytvorte Laplacovskú pyramídu t.ž. pre každý level
$$L^l = G_m \cdot LP_e + (1 - G_m) \cdot LP_c$$
- Rekonštruujte obraz z L.
- Porovnajte rekonštruovaný obraz s obrazom
$$S = M \cdot I_e + (1 - M) \cdot I_c$$

Veľkosti obrázkov

Pre tento príklad som dopredu upravil obrázky do vhodnej veľkosti. Bežne je nutné meniť ich, pre inšpiráciu sa pozrite do pyrBlend.m.

Laplacian of Gaussian

LoG definícia

$$LoG = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

fspecial

`LoG = fspecial('log', fsize, sigma)` - vygeneruje LoG filter veľkosti `fsize × fsize` s parametrom $\sigma = \text{sigma}$.

Kód

```
fsize = 2*ceil(3*sigma) + 1;  
LoG = fspecial('log', fsize, sigma);
```

Úloha

Zobrazte si LoG filter pre rôzne sigma.

Laplacian of Gaussian - aplikácia

LoG definícia

`imfilter(I, LoG, 'same', 'replicate')` - vráti obraz `I` po prefiltrovaní LoG filtrom

Úloha

Stiahnite si obrázok slnečníc a prefiltrujte LoG filtrom s rôznymi σ . Detekujte škvrny pomocou funkcie v `FastPeakFind.m` a vykreslite ich do obrázku.