

Pokročilé spracovanie obrazu - Histogramy, šum, vyhladzovanie a zostrovanie

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

DAI FMFI UK

23.10.2018

Histogram

imhist

imhist(I) - zobrazí histogram, v prípade že výstup zapíšeme do premennej, tak nič nevykreslí ale vráti nam vektor s histogramom.

Úloha

Preveďte zatisie.jpg na šedotónový obrázok. Na jeho histograme sú tri peaky. Upravte obrázok, tak aby pixely približne patriace len jednému z peakov boli úplne biele, ostatné nechajte tak.

Úprava jasu

Gamma korekcia

Kontrast v obraze môžeme meniť pomocou gamma korekcie:

$i_{out} = A \cdot i^\gamma$, kde i predstavuje jas jednotlivých pixelov obrázka. Pozor jas musí byť medzi 0 a 1!

Lineárne roztiahnutie

Pre lineárne roztiahnutie môžeme použiť nasledujúcu úpravu:

$$i_{out} = \frac{i - \min(I)}{\max(I) - \min(I)},$$

kde i sú hodnoty jasu pre jednotlivé pixely a I predstavuje množinu jasov všetkých pixelov. Tiež chceme hodnoty jasu medzi 0 a 1.

Ekvalizácia histogramu

Ekvalizácia

Ekvalizácia histogramu je metóda, ktorá mení jas v obraze tak, aby výsledný histogram vyzeral čo najrovnomernejšie.

histeq

`histeq(I)` - vráti obrázok po ekvalizácii histogramu.

Úloha

Pre obrázok `krajinka.png` vyskúšajte rôzne metódy úpravy kontrastu. Po úpravách si zobrazte obrázky aj histogramy.

Prahovanie

imbinarize

`imbinarize(l)` - vráti binarizovaný obraz s prahom určeným Otsuho metódou.

imbinarize

`imbinarize(l, t)` - vráti binarizovaný obraz s prahom t .

Úloha

Vyskúšajte prahovanie na obrázkoch `coins.png`, `qr.jpg` a `zatisie.jpg`.

Adaptívne prahovanie

imbinarize

`imbinarize(I, 'adaptive')` - vráti binarizovaný obraz s použitím adaptívneho prahovania.

Úloha

Vyskúšajte adaptívne prahovanie na obrázku `coins.png` a `qr.jpg`.

Šum

Ako vzniká šum?

Šum môže vzniknúť vo viacerých fázach získania obrazu. Napríklad už pri reakcii sezeru na svetlo, alebo pri prenose informácie.

Prečo je nutné modelovať šum?

V reálnych situáciách sa šumu nevyhneme, preto je vhodné mať k dispozícii model šumu, aby sme ho vedeli potlačiť.

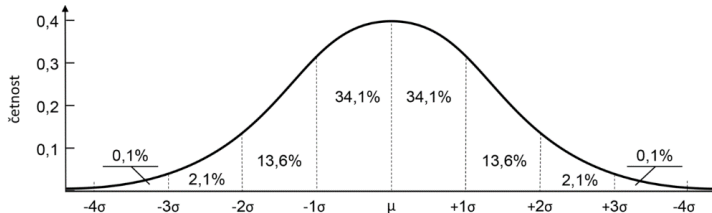
Gaussovský aditívny šum

Aditivita

Šum je aditívny ak pláti $I = I_{orig} + S$.

Gaussovský charakter šumu

$$P(S_{i,j} = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$



Gaussovský aditívny šum - matlab

randn

randn(sz) - vráti maticu veľkosti sz (napr. sz = size(I)) s náhodnými prvkami z gaussovskej distribúcie s $\sigma = 1$ a $\mu = 0$.

Úloha

Vytvorte funkciu zasum(I,sigma), ktorá obraz I (predpokladajte, že je šedotónový a v double) zašumí šumom s $\mu = 0$ a $\sigma = \sigma$. Ošetrte výstup tak, aby bol obraz v rozmedzí medzi 0 a 1. Otestujte funkciu na obrázku. Použite rôzne sigma.

Salt and Pepper

Salt and Pepper

Salt and pepper (sol a korenie) šum nastáva vtedy ak sa jeden pixel zmení buď na úplne tmavý, alebo úplne svetlý.

rand

rand(sz) - vráti maticu veľkosti sz, jej prvky majú náhodné hodnoty z rovnomernej distribúcie medzi 0 a 1.

Úloha

Vytvorte funkciu okoren(l,p1,p2), ktorá šedotónový obraz l (predpokladajte, že je v double) zašumí, tak že s pravdepodobnosťou p1 dostaneme úplne biely pixel a s pravdepodobnosťou p2 dostaneme úplne tmavý pixel. Funkciu otestujte pre rôzne parametre.

Vyhľadzovanie

Prečo vyhľadzujeme

Pri pozorovaní jemne zašumeného obrazu okom je stále jednoduché pozorovať ho a porozmieť jeho obsahu. V spracovaní obrazu a počítačovom videní však niektoré algoritmy môžu jednoducho zlyhať. Preto je nutné potlačiť šum. To dosiahneme vyhľadzovaním.

Konvolúcia

Konvolúcia - integrálna verzia - reálne čísla

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

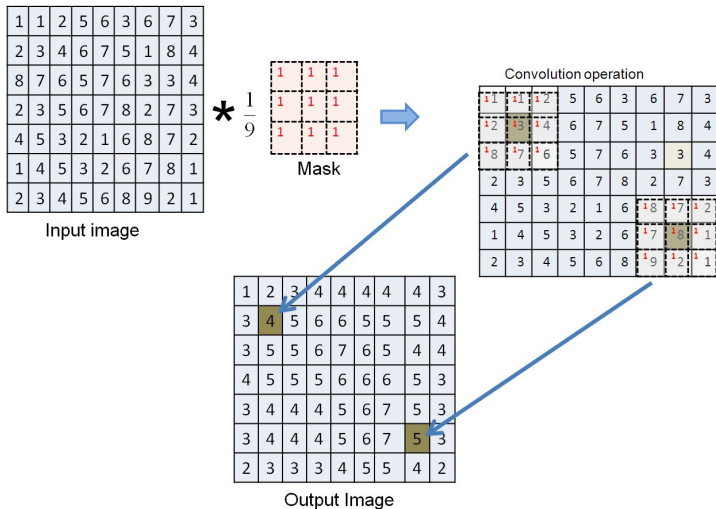
Konvolúcia - diskrétna verzia - reálne čísla

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

Pozor

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

Konvolúcia



Konvolúcia - matlab

conv2

conv2(A,B) - realizuje konvolúciu matice A s maticou B

imfilter

imfilter(I,f) - konvolyčne prefiltruje obraz I filtrom (maticou) f.

imfilter

imfilter(I,f, 'option') - option upravuje veľkosť výsledného obrzu (napr. 'same'), alebo to čo sa bude brať za hodnoty ak filter 'siahne' mimo (napr. 'replicate', 'symmetric'). Môžete použiť aj viac options naraz.

Konvolúcia - filtre

Ručne

Filtre môžeme vyrobiť aj ručne napr. priemerovací filter je `ones(3)/9`.

fspecial

`fspecial('name', params)` - vráti filter podľa mena 'name' a parametrov.

fspecial

`fspecial('average', hsize)` - vráti priemerovací filter veľkosti `hsize`
`fspecial('gaussian', hsize, sigma)` - vráti gaussovský filter veľkosti `hsize` a strednou hodnotou `sigma`

Konvolúcia - filtre

Úloha

Zobrazte si gaussovský filter pre rôzne sigma a veľkosti.

imgaussfilt

`imgaussfilt(I, sigma)` - prefiltruje obraz `I` gaussovským filtrom, je to obdobné ako kombinácia `fspecial` a `imfilter`, ale je efektívnejšia.

Úloha

Zašumte si obrázok aditívnym šumom a skúste ho vyhladiť rôznymi gaussovým a priemerovacím filtrom. To isté spravte pre salt and pepper. Otestujte aj rôzne parametre šumu.

Konvolúcia - filtre

Úloha

Zobrazte si gaussovský filter pre rôzne sigma a veľkosti.

imgaussfilt

`imgaussfilt(I, sigma)` - prefiltruje obraz `I` gaussovským filtrom, je to obdobné ako kombinácia `fspecial` a `imfilter`, ale je efektívnejšia.

Úloha

Zašumte si obrázok aditívnym šumom a skúste ho vyhladiť rôznymi gaussovým a priemerovacím filtrom. To isté spravte pre salt and pepper. Otestujte aj rôzne parametre šumu.

Mediánová filtrácia

Mediánová filtrácia

Na šum typu salt and pepper nefunguje priemerovanie. Preto je vhodnejšie využiť iný postup. Namiesto priemerovania budeme pre každé okno používať medián.

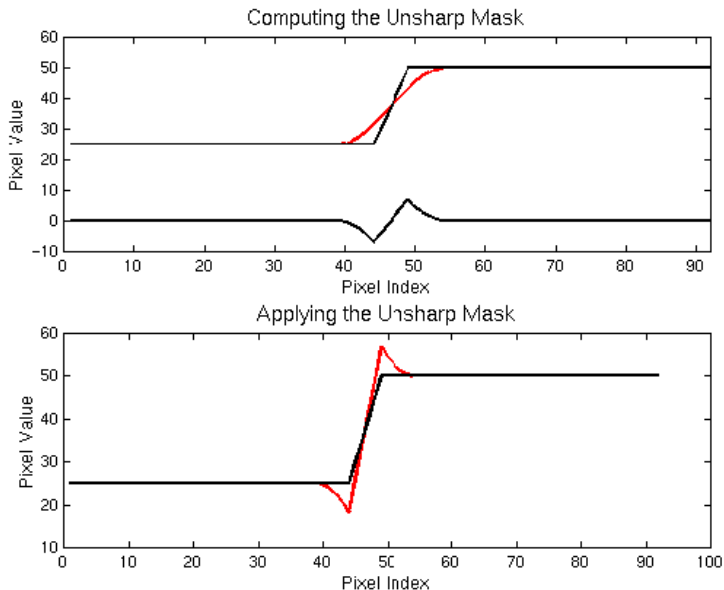
medfilt2

`medfilt2(I, [m n])` - vráti obraz po mediánovej filtrácii oknom veľkosti $m \times n$.

Úloha

Zašumte si obrázok aditívnym šumom a skúste ho vyhladiť mediánovým filtrom. To isté spravte pre salt and pepper.

Unsharp masking - Princíp



Unsharp masking

Ostrenie

Máme obrázok, ktorý je rozostrený. Chceme ho vyostriť. Táto úloha sa dá pochopiť aj ako zvýrazňovanie hrán.

Unsharp masking - princíp

$$I_{ostrý} = I_{originál} + p \cdot (I_{originál} - I_{vyhladený})$$

Úloha

Vytvorte funkciu `unsharp_mask(I,p,sigma)`, ktorý aplikuje unsharp masking s parametrom `p` na obrázok `I` pomocou gaussovho vyhladenia s hodnotou `sigma`. Aplikujte na `blurred.pgm`.