Pokročilé spracovanie obrazu - Kalibrácia a histogramy

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

DAI FMFI UK

18.10.2018

1. Jas
   - Histogram
   - Úpravy jasu
   - Prahovanie
2. Šum
   - Gaussovský aditívny šum
   - Salt and Pepper
3. Vyhladzovanie
   - Priemerovanie
   - Mediánová filtrácia
4. Ostrenie obrazu
   - Unsharp Masking

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 linárne 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(I) - vráti binarizovaný obraz s prahom určeným Otsuho metódou.

imbinarize

imbinarize(I, 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 sezoru 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.

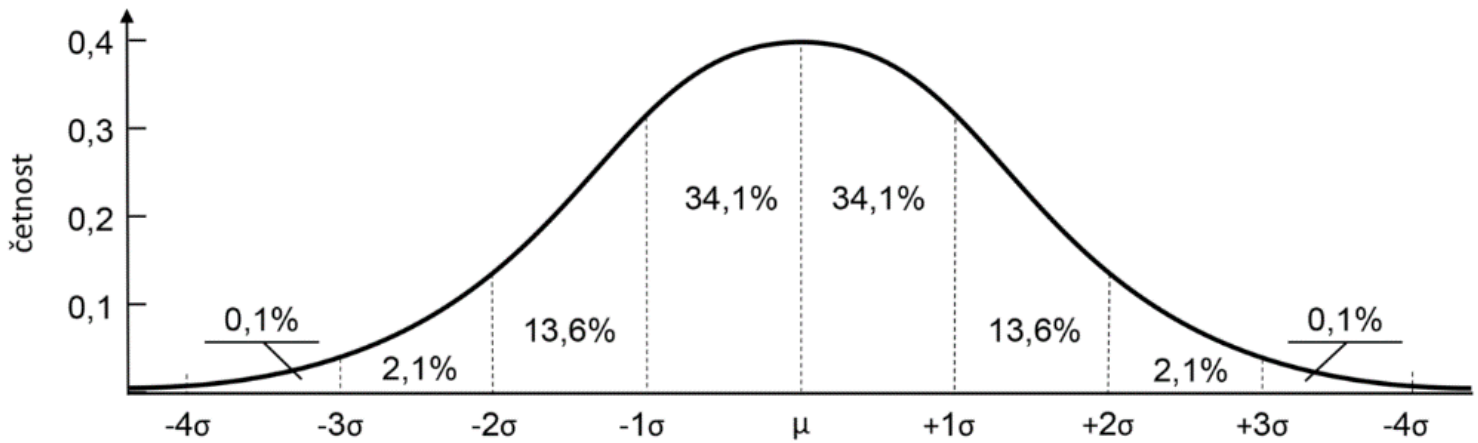
Gaussovský aditívny šum

Aditivita

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

Gaussovovský 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 v double) zašumí šumom s $\mu = 0$ a $\sigma =$sigma. Ošetrite 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(I,p1,p2), ktorá obraz I (predpokladajte, že je v double) zašúmí, tak že s pravdepodobnosťou p1 dostaneme úplne biely pixel a s pravdepodobnosťou p2 dostaneme úplne tmavý pixel. Funkciu otestujte pre rôzne parametre.

Vyhladzovanie

Prečo vyhladzujeme

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

Konvoúcia

Konvolúcia - integrálna verzia - reálne čísla

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

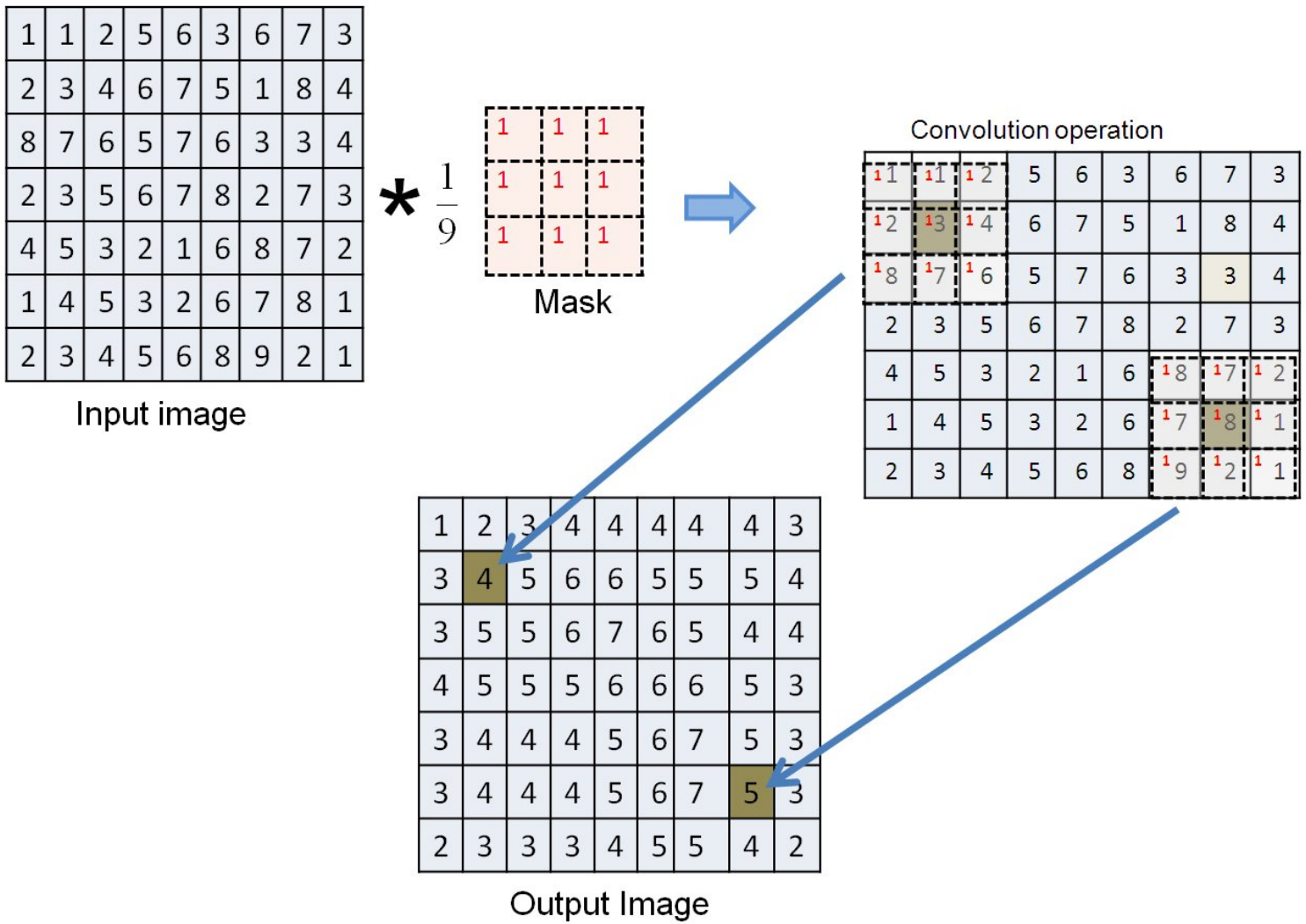
Konvolúcia - diskrétna verzia - reálne čísla

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

Pozor

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

Konvolúcia


1 1 2 5 6 3 6 7 3
2 3 4 6 7 5 1 8 4
8 7 6 5 7 6 3 3 4
2 3 5 6 7 8 2 7 3
4 5 3 2 1 6 8 7 2
1 4 5 3 2 6 7 8 1
2 3 4 5 6 8 9 2 1
Input image
* 1/9
1 1 1
1 1 1
1 1 1
Mask
Convolution operation
1 2 3 4 4 4 4 4 3
3 4 5 6 6 5 5 5 4
3 5 5 6 7 6 5 4 4
4 5 5 5 6 6 6 5 3
3 4 4 4 5 6 7 5 3
3 4 4 4 5 6 7 5 3
2 3 3 3 4 5 5 4 2
Output Image

Konvolúcia - matlab

conv2

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

imfilter

imfilter(I,f) - konvolučne 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.

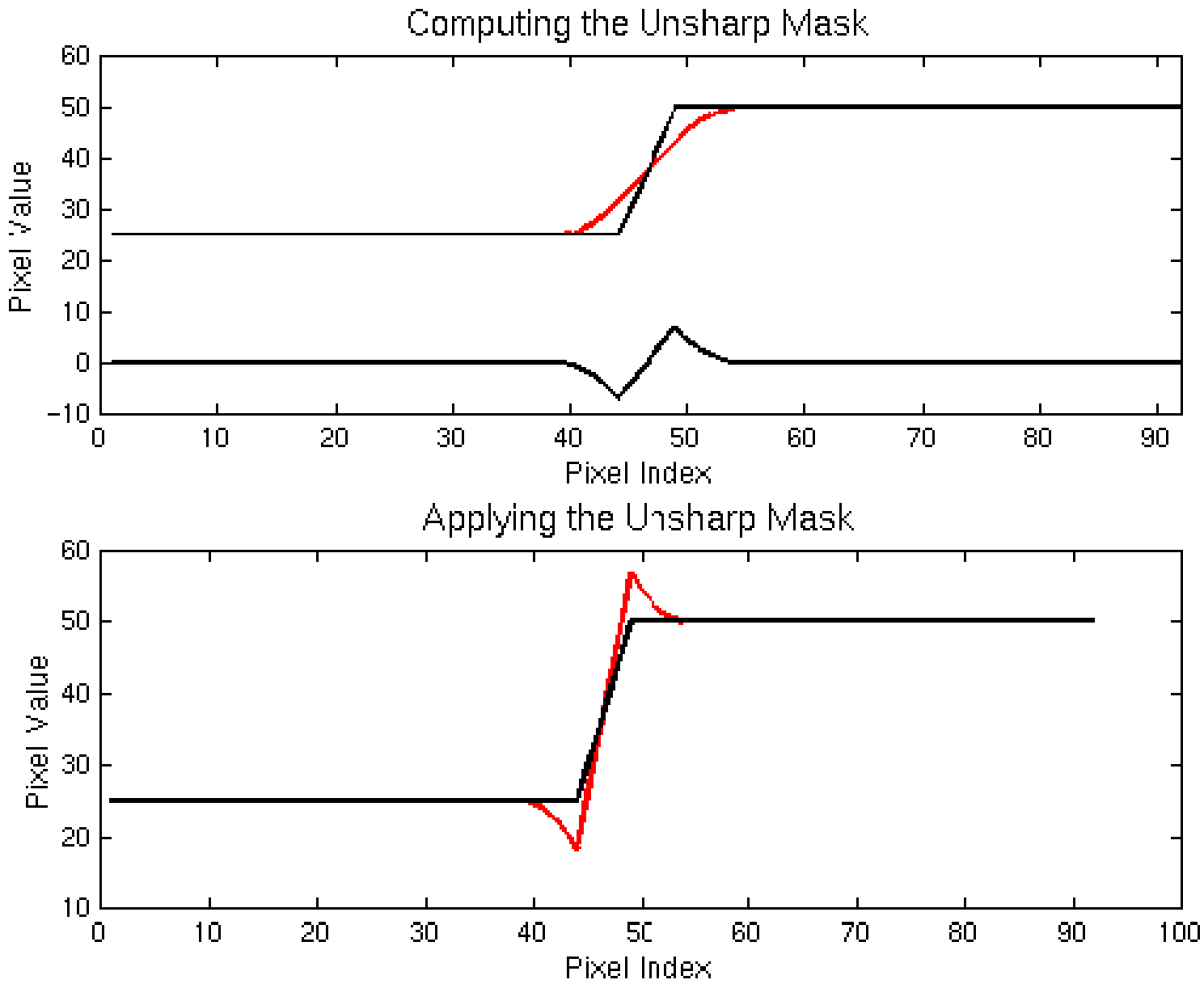
medfilt2

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.