

Základy počítačovej grafiky a spracovania obrazu

Predspracovanie obrazu
Obrazové transformácie

Doc. RNDr. Milan Ftáčnik, CSc.

RNDr. Júlia Škovierová (Kučerová), PhD.

Jasové transformácie

- Základná funkcia je

```
g = imadjust(f, [low_in high_in], [low_out high_out], gamma)
```

- ktorá mapuje hodnoty jasu medzi low_in a high_in na hodnoty medzi low_out a high_out a gamma je parameter gama korekcie – ak = 1, je lineárna, ak je väčší ako 1, tak je ťahaný viac k tmavým výstupným hodnotám, ak menší, naopak

Jasové transformácie 2

- Keď nevieme hodnoty `low_in` a `high_in`, tak použijeme funkciu

```
Low_High = stretchlim(f)
```

- a skombinujeme ju s predošlou:

```
>> g = imadjust(f, stretchlim(f), [ ]);
```

- alebo tiež

```
>> g = imadjust(f, stretchlim(f), [0 1]);
```

Ekvalizácia histogramu

- Jednoduchý príkaz:

```
g = histeq(f, nlev)
```

- kde *nlev* je počet jasových úrovní, default je 64, ale lepšie je nastaviť 256, vtedy to naozaj robí ekvalizáciu histogramu
- Existuje ešte špecifikácia histogramu a adaptívna ekvalizácia histogramu

Úlohy na jasové transformácie

- Načítajte nízko kontrastný obraz a zobrazte si obraz aj jeho histogram
- Do ďalších dvoch okien zobrazte jeho jasovú transformáciu cez **roztiahnutie jasového intervalu** a príslušný histogram takto upraveného obrazu
- Do ďalších dvoch okien zobrazte obraz a histogram po **ekvalizácii histogramu** a porovnajte

Lokálny priestorový filter

- Priestorový filter s maskou (kernelom) w
- Filtrácia na základe lokálneho okolia pixelu
- Priemerovací filter:

 $\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

 $\frac{1}{25}$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Lokálny priestorový filter

- Priestorový filter s maskou w má nasledovnú syntax

```
g = imfilter(f, w, filtering_mode, boundary_options, size_options)
```

- kde mód filtra je *corr* alebo *conv*, hraničné podmienky súvisia s dopĺňaním núl na okrajoch obrazu a veľkostný parameter je buď *same* alebo *full*

Úlohy na priestorové predspracovanie

- Načítajte obraz – zašumte ho Gaussov- ským šumom – cez `imfilter` a príslušnú masku naňho použite obyčajné priemero- vanie a cez ďalšiu masku Gaussovský filter 3x3 alebo 5x5

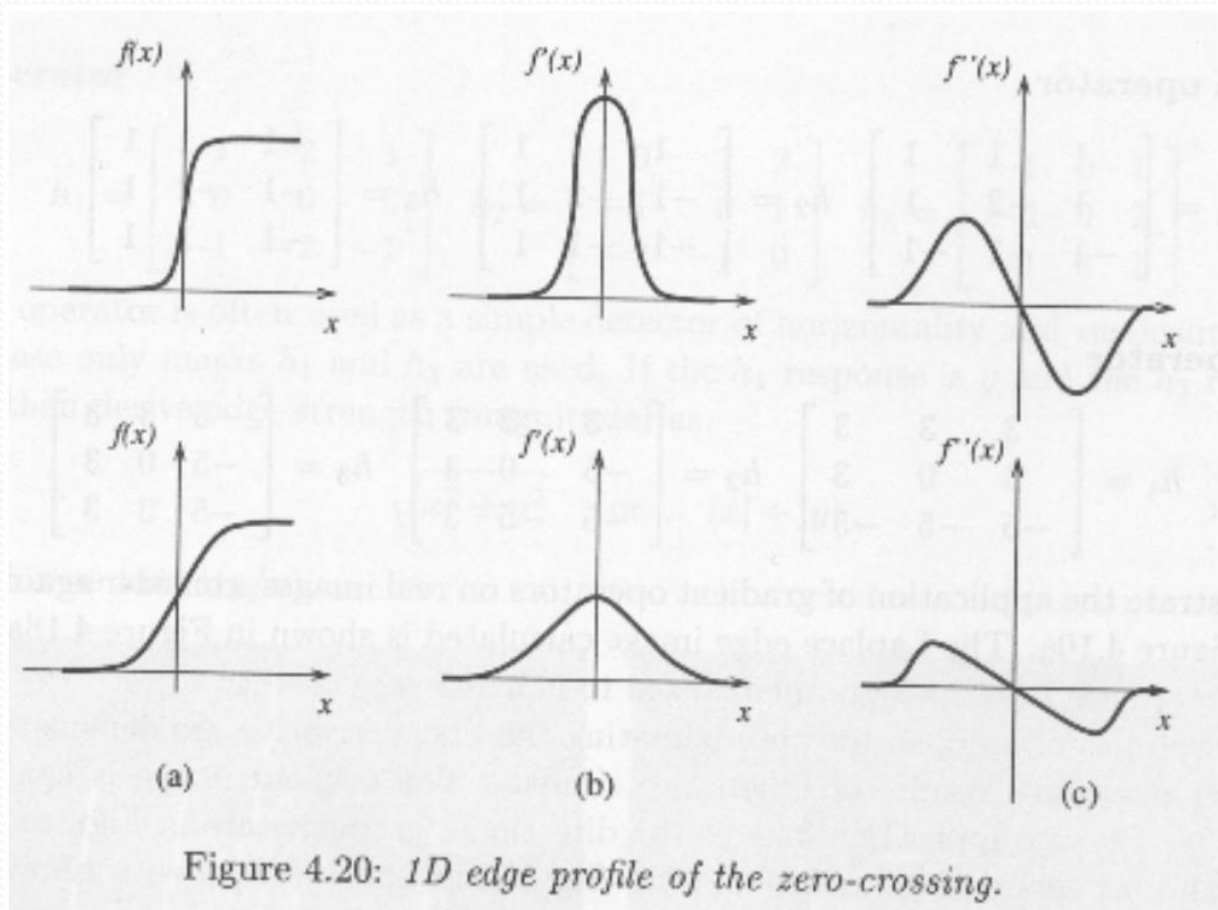
```
gaussian_mask = fspecial('gaussian', [n m], sigma)
```

- Načítajte obraz a zašumte ho impulzným čierno- bielym šumom a použite naňho mediánový filter

```
I_noise = imnoise(I, 'salt & pepper');
```

```
res = medfilt2(I_noise)
```


Detekcia hrán



Detekcia hrán

- Funkciu `imfilter` vieme použiť aj na získanie gradientov v obraze, ktoré využívame pri detekcii hrán
- napr. `fspecial('sobel')` nám vráti masku Sobelovho grad. operátora orientovanú na detekciu horizontálnych hrán

```
[ 1  2  1  
  0  0  0  
-1 -2 -1 ]
```

- Ak chceme detekovať vertikálne, musíme ju transponovať
- Funkcia `imfilter` potom prekonvuluje obraz s touto maskou → získame hrany v obraze

Detekcia hrán

- Výsledok po detekcii horizontálnych a vertikálnych hrán chceme potom skombinovať do jedného – kde budú detekované oba typy hrán
- To spravíme nasledovne:

$$res = \sqrt{H^2 + V^2}$$

- Kde H je výsledok po detekcii horizontálnych hrán, a V po detekcii vertikálnych.
- **Úloha:** Vyskúšajte detekciu hrán na obrázku qr.jpg
- *POZOR* pri týchto výpočtoch chcete pracovať s obrazom, kt. jasové hodnoty sú v intervale (0,1)

Ostrenie obrazu

- Laplacián (druhá derivácia obrazu)

$$\Delta f = \nabla \cdot \nabla f = \sum_{i=1}^n \frac{\partial^2 f}{\partial x_i^2} \stackrel{2D}{=} \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Konvolučné jadro

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ alebo } \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Generujeme ručne alebo `fspecial('laplacian', alpha)`
- `alpha` určuje, ako veľmi berieme do úvahy diagonálnych susedov

Ostrenie obrazu

- Generujeme ručne alebo `fspecial('laplacian', alpha)`
- Pomocou jadra Laplacianu teda ostríme obraz takto:

$$I_{\text{ostry}} = I_{\text{original}} - p(L_{\text{jadro}} * I_{\text{original}})$$

- **Úloha:** skúste si ostrenie pomocou Laplaciánu na obraze `blurred.pgm`
- *POZOR* na dátové typy (`double`)