Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Poznámky k předmětu Zpracování obrazu

Matěj Trakal

Poslední úprava:8. prosince 2010

Obsah

1 Cvičení 01 – 29. září 2010

home_teacher:mifr0494/Studmat/Inzo bazíruje na barvách
sehnat elektronickou verzi matrošů od něj (Základy zpracování obrazu)
u zkoušky bude požadovat rozsah jeho skript
matlab 7.8. R2009a
nějaký porgram + 2 teoretické

Přednášky: Lineární transformace Integrální transformace (furierova, cosinova)
storageuei01.upceucebny.cz

VyukaImPlfa.pdf?

Matlab:

```
I=imread('Cmy343.bmp'); % načtení souboru do matice
imshow(I); % zobrazení obrázku
clear all; % maže všechny proměnné
edit nazev.m % editační okno s vytvořením souboru/rditace souboru
```

Příklad 1

```
$v = \left(2 \times 9.7^{200}\right)/3;$
```

Příklad 2

```
s = sin(pi/6); % 0.5
as = asin(s); % 0.5236
s = sind(30); % 0.5
as = asind(s); % 0.5236
```

Příklad 3

```
x=1;
y = x^2 + (1/(sqrt(x) + x + x^2 + x^3 + 5));
```

Příklad 4

```
x=1;
I = quadl('tan(x)',0,pi/3);
```

Funkce:

```
edit f1.m

function y = f1(x)
t=tan(x);

I = quadl('f1',0,pi/3); %mně nefunguje
```

Příklad 5 $y = sin(x^2)$ rozsah: 0, 2*pi, 100 bodů

```
x = 0:2*pi/99:2*pi;
y = sin(x.^2);
```

Matlab se snaží dělat operaci násobení matic, s teškou provede normálně, jako nematicovou operaci - násobení každého prvku vektoru (prvek na prvek).

```
plot(x,y);
```

Příklad 6

```
z = sin(y+sin(3*x)) <-3, 3, 50 bodů>

x = -3:6/49:3;
y=x;
[X,Y] = meshgrid(x,y);
z = sin(Y+sin(3.*X));
surf(x,y,z);
plot3(X,Y,z);
```

2 Cvičení 02 – 6. října 2010

det(C); trace(C'*C) rank(C); C(3,:)=[4,5,3,2] apostrof transforume matici (řádková - sloupcová) inc(C); eig(F);

2.1 Příklad 1

```
v1 = [1;2;3;4]
v2 = [4;3;2;1]
```

2.1.1 Součin matic:

```
  sM = A*B

%  skalární součin (výsledkem jedna hodnota):
   skal = v1'*v2
%  vektorový součin (výsledkem velikost obou matic):
   vek = v1*v2'
%  vektor * matice (výsledek je vektor)
   vmat = v1'*C
```

2.2 Příklad 2

```
D = [0,1,1,0;  1,1,0,0;  0,1,0,1;  1,0,0,1]

invC = inv(C);
jm = C*inv(C) % zkouška funkčnosti
```

2.3 Příklad 3

Rozepište konjunkční sumu pro obraz 4×4 a odvoďte n-tý řádek matice H (obecně)!

$$f = (f00f01f02f03f10f11f12f13f20f21f22f23f30f31f32f33)$$

$$g(x,y) = \sum(3, \alpha = 0(\sum(3, \beta = 0, (f(\alpha,\beta) * h(x,\alpha,y,\beta)))))$$

$g(x,y) = f(0,0)*h(x,0,y,0)+f(1,0)*h(x,1,y,0)+f(2,0)*h(x,2,y,0)+f(3,0)*h(x,3,y,0)+f(0,1)*$

5.řádek:

$$h(0,0,1,0), h(0,1,1,0), h(0,2,1,0), 16.prvekbudeh(0,3,1,3)$$

Základní transformace, ostatní jsou už jen transformace této rovnice

$$g = H * f;$$

2.4 vyhlazování obrazu

Odvoďte transformační matici H vyhlazováním obrazu 3×3, aplikujte na obraz prvočísel. Proveďte zkoušku pro první 3 body. (demonstrační příklad)!

2.4.1 Obraz prvočísel:

$$f = (1, 2, 35, 7, 1113, 17, 19)$$

$$f(0, 0), f(0, 1), f(0, 2)f(1, 0), f(1, 1), f(1, 2)f(2, 0), f(2, 1), f(2, 2)$$

2.4.2 Konvoluční maska

$$m = 1/10(1, 1, 11, 2, 11, 1, 1)$$

Musíme matici rozšířit (kvůli okrajovým pixelům), na okraje dáme pixel z druhé strany (navazování obrazu jakoby)

$$f(2, 2)f(2, 0), f(2, 1), f(2, 2)f(2, 0); f(0, 2)f(0, 0), f(0, 1), f(0, 2)f(0, 0); f(1, 2)f(1, 0), f(1, 1), f(1, 2)$$

2.4.3 Vektor obrazu

$$f = (f(0, 0)f(1, 0)f(2, 0)f(0, 1)f(1, 1)f(2, 1)f(0, 2)f(1, 2)f(2, 2)$$

2.4.4 Konvoluce obrazu

$$g(0, 0) = 1/10{*}(1{*}f(2, 2)+1{*}f(2, 0)+1{*}f(2, 1)+1{*}f(0, 2)+2{*}f(0, 0)+1{*}f(0, 1)+1{*}f(1, 2)+1{*}f(1, 0)$$

Podle matice F (vektor obrazu, se musí matice seřadit)
$1/10 * (2, 1, 1, 1, 1, 1, 1, 1, 1)$

Prostě po zkrácení vznikne matice $1/10*(9\times9)$, kde na diagonále jsou 2, všude jinde 1.
$g = 1/10 * matice9 \times 9$ s dvojkovou diagonálou $*(1; 5; 13; 2; 7; 11; 13; 17; 19) = (7, 9; 8, 3; 9, 1; ...)$

3 Cvičení 03 – 13. října 2010

3.1 Příklad 1:

Odvoďte matici H pro detekci hran Laplaceovým operátorem na obraz 3 × 3. Aplikujte na obraz F.

3.1.1 Konvoluční maska

$L = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ 0101 – 41010 $f = \begin{pmatrix} 1 & 10 & 10 \\ 1 & 10 & 10 \\ 1 & 10 & 10 \end{pmatrix}$ Laplaceův operátor je druhá derivace funkce, nebo co to žvatlal :)

3.1.2 Obecná matice:

$\begin{pmatrix} f11 & f12 & f13 \\ f21 & f22 & f21 \\ f31 & f32 & f33 \end{pmatrix}$

3.1.3 Rozšíření matice

$\begin{pmatrix} f33 & f31 & f32 & f33 & f31 \\ f13 & f11 & f12 & f13 & f11 \\ f23 & f21 & f22 & f21 & f21 \\ f33 & f31 & f32 & f33 & f31 \\ f13 & f11 & f12 & f13 & f11 \end{pmatrix}$ 1.ř.: $0 \times f33 + 1 \times f31 + 0 \times f32 + 1 \times f13 - 4 \times f11 + 1 \times f12 + 0 \times f23 + 1 \times f21 + 0 \times f22$ (konvoluční matice × rozšířená matice – ale jen 3 × 3 matici)

musíme je seřadit podle matice

```
f = (f11;f21;f31;f12;f22;f32;f13;f23;f33)
```

-411100100

2.ř.: $0 \times f131 \times f110 \times f121 \times f23 - 4 \times f211 \times f220 \times f331 \times f310 \times f32$ 1 -4 1 0 1 0 0 1 0

3.ř.: $0f231f210f221f33 - 4f311f320f131f110f12$ $11 - 4001001$

3.1.4 Výsledná odvozená matice:

```
H_L = [
-4, 1, 1, 1, 0, 0, 1, 0, 0;
1, -4, 1, 0, 1, 0, 0, 1, 0;
1, 1, -4, 0, 0, 1, 0, 0, 1;
1, 0, 0, -4, 1, 1, 1, 0, 0;
```

```
0, 1, 0, 1, −4, 1, 0, 1, 0;
0, 0, 1, 1, 1, −4, 0, 0, 1;
1, 0, 0, 1, 0, 0, −4, 1, 1;
0, 1, 0, 0, 1, 0, 1, −4, 1;
0, 0, 1, 0, 0, 1, 1, 1, −4];

f =[1;1;1;10;10;10;10;10;10]

g = H * f
g =[18;18;18;−9;−9;−9;−9;−9;−9]
```

3.2 Příklad 2:

Odvození matice H u bodových operací. (nejspíš taky u zkoušky, ten nad určitě, možná i stejnej :))

Odvoďte matici H pro k-násobné zvýšení jasu a kontrastu. Aplikujte na obraz f, přesvědčte se, že je obraz separabilní.

3.2.1 Konvoluční suma

$$g(x,y) = \sum_{\alpha=0}^{2}\sum_{\beta=0}^{2} f(\alpha,\beta) * h(x-\alpha, y-\beta) = f(\alpha,\beta) * h(x,y) = K * f(\alpha,\beta)$$

Dodáno další hodinu, jako správně:

$$g(x,y) = \sum_{\alpha=0}^{2}(\sum_{\beta=0}^{2} f(\alpha,\beta) * h(x-\alpha, y-\beta)) = f(x,y) * k(x,y) = K * f(x,y)$$

```
f = (
  f11 f12 f13
  f21 f22 f23
  f31 f32 f33
  )
g11 = K*f11 + 0*f21 + 0*f31 ... + 0*f33
g21 = K*f21 + 0*f11 + ... + 0*f33
g31 = K*f31 + 0*f11 + ... + 0*f33
```

```
g = [
  K 0 0 0 0 0 0 0 0;
  0 K 0 0 0 0 0 0 0;
  0 0 K 0 0 0 0 0 0;
  0 0 0 K 0 0 0 0 0;
  0 0 0 0 K 0 0 0 0;
  0 0 0 0 0 K 0 0 0;
  0 0 0 0 0 0 K 0 0;
  0 0 0 0 0 0 0 K 0;
```

```
    0 0 0 0 0 0 0 0 0 K;
    ];
```

```
hc = hr = [
    sqrt(K), 0, 0;
    0, sqrt(K), 0;
    0, 0, sqrt(K);
    ];
```

3.3 Příklad 3:

Ukažte platnost vztahu $g = Hc * f * Hr'$ pro Haarovu transformaci a obraz f. !!! bude u zkoušky !!!

```
f = [
    0,0,0,0;
    0,50,50,0;
    0,50,50,0;
    0,0,0,0];
```

Haarova transformace bude také $4 * 4$, výsledná bude tedy $16 * 16$.

```
hc = hr = [
    1,1,1,1;
    1,1,-1,-1;
    1.414,-1.414,0,0;
    0,0,1.414,-1.414]
```

$gs = hc * f * hr'$;

3.3.1 Pro kontrolu:

```
    H = kron(hr,hc);
    vf = matnavek(4,4,f); % vektor F
    vg = H*vf; % vektor G
    g = veknamat(4,4,vg);
```

3.4 Příklad 4:

Proveďte rozklad separabilní ... vlož Haarabilní transformace (VH) 4×4 na elementární obrazy.

Elementární obrazy se počítají z matic Hc a Hr, sloupce se berou jako vektory a sloupce dávají nějaký obrazy... WTF?

```
f = [
    0,0,0,0;
    0,50,50,0;
    0,50,50,0;
    0,0,0,0];
```

```
hc = hr = [
   1,1,1,1;
   1,-1,1,-1;
   1,1,-1,-1;
   1,-1,-1,1];

g = hc * f * hr';
```

3.4.1 Kontrola

```
   u1 = [1;1;1;1];
   u2 = hc(:,2); % druhý sloupec, všechny řádky
   u3 = hc(:,3);
   u4 = hc(:,4);
   v1 = u1;
   v2 = u2;
   v3 = u3;
   v4 = u4;
   eo11 = 0*u1*v1';
   eo12 = 0*u1*v2';
   eo13 = 0*u1*v3';
   eo14 = 0*u1*v4';
   eo21 = 0*u2*v1';
   eo22 = 50*u2*v2';
   eo23 = 50*u2*v3';
   eo24 = 0*u2*v4';
   eo31 = 0*u2*v1';
   eo32 = 50*u2*v2';
   eo33 = 50*u2*v3';
   eo34 = 0*u2*v4';
   eo41 = 0*u1*v1';
   eo42 = 0*u1*v2';
   eo43 = 0*u1*v3';
   eo44 = 0*u1*v4';

   % sečtení
deo = eo11 +eo12 +eo13 +eo14 + eo21 +eo22 +eo23 +eo24 +
  eo31 +eo32 +eo33 +eo34+eo41 +eo42 +eo43 +eo44;
```

NĚKDE TAM? JE CHYBA ale!!!

4 Cvičení 04 – 20. října 2010

Příklad na nějakou transformaci.

Proveďte SVD transformaci obrazu `kvet.bmp` Obraz je rozložený na 3 matice, U, V a λ.

```
kvetsc
imshow(kvet,[0,255],'InitialMagnification','fit');
kvetn = kvet/256;
imwrite(kvetn, 'kvet.bmp');
I = imread('kvet.bmp');
I = im2double(I);
[U,S,V] = svd(I);

eo1 = S(1,1)*U(:,1)*V(:,1)';
imshow(eo1,'InitialMagnification','fit');
eo2=S(2,2)*U(:,2)*V(:,2)';
eo3=S(3,3)*U(:,3)*V(:,3)';
eo4=S(4,4)*U(:,4)*V(:,4)';
eo5=S(5,5)*U(:,5)*V(:,5)';
kveteo = eo1+eo2+eo3+eo4+eo5;
```

Vypočítejte první 3 hodnoty,

Fourierovy transformace pro digitální signál $f = 1,1,1,1,0,0,0,0$; $F(n) = \frac{1}{\sqrt{N}} \sum_{x=0}^{7} f(x) \times e^{-j\frac{2\pi}{N} \times n \times x}$

$F(0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{7} f(x) = \frac{1}{\sqrt{8}(1+1+1+1)} = \frac{4}{\sqrt{8}} = 1.414$

$F(1) = \frac{1}{\sqrt{N}} \sum_{x=0}^{7} f(x) \times e^{-j\frac{\pi}{4} \times x} = \frac{1}{\sqrt{N}} \left(1 + 1e^{-j\frac{\pi}{4}} + 1e^{-j\frac{\pi}{2}} + 1e^{-j\frac{3\pi}{4}}\right) = \frac{1}{\sqrt{8}} \left(1 + \left(\cos\frac{\pi}{4} - j\sin\frac{\pi}{4}\right) + (c$
$0.3535 - j0.8535$

```
X = [1;1;1;1;0;0;0;0];
N = 8; % možná 0
Fx = 1 / sqrt(N) * fft(X)
plot(abs(Fx));
x=[1;1;1;1;1;1;1;1]
Fx = 1 / sqrt(N) * fft(x)
```

Př: Proveďte FFT transformaci obrazu 30×30

```
f = ones(30,30);
f(:,13:19) = 0;
imshow(f);

F = 1/sqrt(30) * fft(f);
Fs = fftshift(F);
imshow(abs(Fs),[0 10],'InitialMagnification', 'fit');
```

```
F = 1/sqrt(30) * fft2(f);
Fs = fftshift(F);
imshow(abs(Fs),[0 10],'InitialMagnification', 'fit');
```

Př 4.4: Odvod'te matici U furierovi transformace pro obraz 4×4 aplikujte na obraz f. !!! Bude u zkoušky !!! $f = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

$U_{x,\alpha} = \frac{1}{\sqrt{N}} e^{-j\frac{2\pi}{N}xd}$
$U_{0,0} = \frac{1}{2}$
$U_{0,1} = \frac{1}{2}$
$U_{0,2} = \frac{1}{2}$
$U_{1,0} = \frac{1}{2}$
$U_{1,1} = \frac{1}{2}e^{-j\frac{\pi}{2}}$
$U_{1,2} = \frac{1}{2}e^{-j\pi}$
$U_{1,3} = \frac{1}{2}e^{-j\frac{3\pi}{2}}$
$U_{2,0} = \frac{1}{2}$
$U_{2,1} = \frac{1}{2}e^{-j\pi}$
$U_{2,2} = \frac{1}{2}e^{-j2\pi}$
$U_{2,3} = \frac{1}{2}e^{-j3\pi}$
$U_{3,0} = \frac{1}{2}$
$U_{3,1} = \frac{1}{2}e^{-j\frac{3\pi}{2}}$
$U_{3,2} = \frac{1}{2}e^{-j3\pi}$
$U_{3,3} = \frac{1}{2}e^{-j\frac{9\pi}{2}}$

$$U = \begin{pmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & -0.5j & -0.5 & 0.5j \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & 0.5j & -0.5 & -0.5j \end{pmatrix}$$

```
U = [
   0.5, 0.5, 0.5, 0.5;
   0.5, -0.5i, -0.5, 0.5i;
   0.5, -0.5, 0.5, -0.5;
   0.5, 0.5i, -0.5, -0.5i
];
f=[0,0,0,0;0,1,1,0;0,1,1,0;0,0,0,0];
Ff = U*f*U
iFf = conj(U)*Ff*conj(U)
```

5 Cvičení 05 – 3. listopadu 2010

```
kvetsc
kvetn = kvet/256
imwrite(kvetn,'kvet.bmp');

I=imread('kvet.bmp');

I = im2double(I);

imshow(I,'InitialMagnification','fit');

dctI = dct2(I);
imshow(dctI,'InitialMagnification','fit');

dctI(6:8,6:8) = 0; % omezení frekvencí (filtrace)

% inverzní transformace
invDctI = idct2(dctI);
imshow(invDctI,'InitialMagnification','fit');

% Proveďte diskrétní kosinovou transformaci
% pomocí blokových procedur.

T = dctmtx(8)

B = blkproc(I,[8 8],'P1*x*P2',T,T');
mask = zeros(8,8);
mask(1:5,1:5)=1
B2 = blkproc(B,[8,8],'P1.*x',mask);

% inverzní transformace
invB2 = blkproc(B2,[8 8],'P1*x*P2',T',T);
imshow(invB2,'InitialMagnification','fit');
```

5.1 Příklad 3

Proveďte DCT a iDCT transformaci obrazu eight.tif.

```
clear all;

I = imread('eight.tif'); % obrázek přímo v matlabu...
I = im2double(I);

imshow(I);
T =dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T');
imshow(50*B);

mask = zeros(8,8);
mask(1:4,1:4)=1

% tečka tam je prej něco jako pixelový násobení/operace
Bm = blkproc(B,[8 8], 'P1.*x',mask);
invBm = blkproc(B,[8 8], 'P1*x*P2',T',T);

imshow(I), figure, imshow(invBm);
```

5.2 Příklad 4

Proveďte VH a iVH transformaci obrazu f a vypočítejte elementární obraz eo 3,4.

$$f = \begin{pmatrix} 10 & 10 & 10 & 10 \\ 10 & 60 & 60 & 10 \\ 10 & 60 & 60 & 10 \\ 10 & 10 & 10 & 10 \end{pmatrix}$$

$$f = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

```
clear all
f = [10,10 ,10 ,10;10 ,60 ,60 ,10;10 ,60 ,60 ,10;10 ,10 ,10 ,10;]
WH = [1,1,1,1;1,-1,1,-1;1,1,-1,-1;1,-1,-1,1;]
whf = 0.25*WH*f*WH

% kontrola
iwhf = 0.25 * WH * whf*WH
```

Znaménková změna (počet změn na řádcích si napíšeme a pak ji seřadíme)

$$WH = \left|\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{matrix}\right| \begin{matrix} 0 \\ 3 \\ 1 \\ 2 \end{matrix}$$

$$WHU = \left|\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{matrix}\right| \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix}$$

```
WHU =[1  ,1  ,1  ,1;1  ,1  ,-1  ,-1;1  ,-1  ,-1  ,1;1  ,-1  ,1  ,-1;]

R3 = WHU(3,:);
R4 = WHU(4,:);
EO34 = 10*R3'*R4;

I = mat2gray(EO34)
imshow(I,'InitialMagnification',3000);
```

Proveďte H transformaci pro obraz 4×4 a její inverzi

$$WH = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1.414 & -1.414 & 0 & 0 \\ 0 & 0 & 1.414 & -1.414 \end{pmatrix}$$

```
% Haarova transformace
H = [1,1,1,1;1,1,-1,-1;1.414,-1.414,0,0;0,0,1.414,-1.414;]
hf = 0.25*H*f*H'
ihf = 0.25*H'*hf*H

hf(4,4) = 0
ihf = 0.25*H'*hf*H
imshow(ihf,[-10, 70],'InitialMagnification','fit');
```

5.3 cosi...

```
clear all
haarmat

I = imread('kvet.bmp');
I = im2double(I);
```

```
Hkvet =(0.125*haar8*I*haar8');
Hkvet(5:8,5:8) = 0
iHkvet = 0.125 *haar8'*Hkvet*haar8
imshow(iHkvet,'InitialMagnification','fit');
```

6 Cvičení 06 – 10. listopadu 2010

6.1 Příklad (nebude u zkoušky)

Máme náhodnou veličinu

pH	3.41	3.5	3.57	3.58	3.69	3.7	3.85	4.0	4.21	4.35	4.51
T	20	21	22	23	24	25	26	27	28	29	30

```
Ph = [3.41; 3.5  ; 3.57  ;3.58  ; 3.69 ; 3.7  ; 3.85  ;4.0 ; 4.21  ; 4.35
; 4.51 ];
T = [20      ; 21   ; 22   ;  23 ; 24 ; 25; 26; 27; 28; 29;30];
mPh = mean(Ph) % střední hodnota
mT = mean(T) % střední hodnota
vPh = var(Ph)% rozptyl
vT = var(T) % rozptyl
Cpht = cov(Ph,T) % kovariance (zobrazují se jako matice)
Corpht = corrcoef(Ph,T) % vazba mezi kovariancí a korelační koeficient?
```

Korelační koeficient je roven...

$$\tau = \frac{cov(pht)}{\sigma_{ph} * \sigma_t} = 0.966$$

6.2 Příklad 2

Vypočítejte výběrové průměry v kovarianční matici a korelační matici trojrozměrné náhodné veličiny, složek RGB 3×3.

$$R = \begin{pmatrix} 3 & 3 & 5 \\ 3 & 6 & 4 \\ 4 & 5 & 5 \end{pmatrix}$$

$$G = \begin{pmatrix} 3 & 1 & 3 \\ 4 & 6 & 2 \\ 4 & 5 & 3 \end{pmatrix}$$

$$B = \begin{pmatrix} 4 & 2 & 3 \\ 1 & 4 & 6 \\ 4 & 3 & 3 \end{pmatrix}$$

$$C_{rgb} = \begin{pmatrix} Crr & Crg & Crb \\ Cgr & Cgg & Ggb \\ Cbr & Cbg & Cbb \end{pmatrix}$$

```
R =[3 ;3 ;5;3 ;6 ;4;4 ;5 ;5];
G =[3 ;1 ;3;4 ;6 ;2;4 ;5 ;3];
B =[4 ;2 ;3;1 ;4 ;6;4 ;3 ;3];
mR = mean(R) % průměr složky R
mG = mean(G) % průměr složky G
mB = mean(B) % průměr složky B
vR = var(R)
vG = var(G)
vB = var(B)
Crg = cov(R,G) % kovariance mezi náhodnou veličinou R a náhodnou veličinou G
Crb = cov(R,B) % kovariance mezi náhodnou veličinou R a náhodnou veličinou B
Cgb = cov(G,B) % kovariance mezi náhodnou veličinou G a náhodnou veličinou B
```

$$C_{rgb} = \begin{pmatrix} 1.1944 & 1.01319 & 0.4167 \\ 1.01319 & 2.2778 & -0.0417 \\ 0.4167 & -0.0417 & 2 \end{pmatrix}$$

```
Rrg = corrcoef(R,G) % korelace mezi R a G
Rrb = corrcoef(R,B) % korelace mezi R a B
Rgb = corrcoef(G,B) % korelace mezi G a B
```

$$Corr_{rgb} = \begin{pmatrix} 1 & 0.6147 & 0.2696 \\ 0.6147 & 1 & -0.0195 \\ 0.2686 & -0.0195 & 1 \end{pmatrix}$$

6.2.1 Příklad 3 – bude u zkoušky!

Z osmi realizací obrazu 4×4, vypočítejte: střední hodnotu jednotlivých náhodných veličin, prostorové průměry jednotlivých realizací a nakonec 3 hodnoty autokorelační funkce $(E\{f_{11} * f_{11}\}, E\{f_{23} * f_{32}\}, E\{f_{32} * f_{43}\})$.

$$\begin{vmatrix} \times & . & . & . \\ . & . & \times & . \\ . & \times & . & . \\ . & . & \times & . \end{vmatrix}$$

Předem připravené věci v matlabu (NP8.mat):

```
clear all
load NP8.mat
```

Máme 8 obrazů o1 – o8, a musíme je dostat do vektorů. Budeme vytvářet náhodné veličiny – ze všech 8 obrázků potřebujeme vzít hodnoty ze stejné pozice.

```
% vektor z 8 obrázků na pozici 1,1
nv11 = [o1(1,1);o2(1,1);o3(1,1);o4(1,1);o5(1,1);o6(1,1);o7(1,1);o8(1,1)]
```

```
% vektor z 8 obrázků na pozici 2,3
nv23 = [o1(2,3);o2(2,3);o3(2,3);o4(2,3);o5(2,3);o6(2,3);o7(2,3);o8(2,3)]
% vektor z 8 obrázků na pozici 3,2
nv32 = [o1(3,2);o2(3,2);o3(3,2);o4(3,2);o5(3,2);o6(3,2);o7(3,2);o8(3,2)]
% vektor z 8 obrázků na pozici 4,3
nv43 = [o1(4,3);o2(4,3);o3(4,3);o4(4,3);o5(4,3);o6(4,3);o7(4,3);o8(4,3)]
```

Výběrové poměry náhodných veličin

Aritmetický vektor, který obsahuje soubor všech průměrů náhodných veličin

```
mnv11 = mean(nv11)
mnv23 = mean(nv23)
mnv32 = mean(nv32)
mnv43 = mean(nv43)
```

Prostorové průměry jednotlivých obrázků (sečtou se jasy jednotlivých pixelů obrázků)

```
mo1 = mean(mean(o1))
mo2 = mean(mean(o2))
mo3 = mean(mean(o3))
mo4 = mean(mean(o4))
mo5 = mean(mean(o5))
mo6 = mean(mean(o6))
mo7 = mean(mean(o7))
mo8 = mean(mean(o8))
```

Předpoklad ergodického (ergodicity) procesu – že jsou hodnoty průměrů stejné, jak náhodné veličiny v rámci náhodného procesu, tak prostorové průměry jednotlivých realizací.

Autokorelace// střední hodnota, kvůli nematicovému násobení je tam tečka (vynásobení dvojic – pixel na pixel)

```
% autokorelace
% střední hodnota, kvůli nematicovému násobení je tam tečka (vynásobení dvojic – pix
Rf1111 = mean(nv11.*nv11)
Rf2332 = mean(nv23.*nv32)
Rf3243 = mean(nv32.*nv43)
```

Tento proces je s určitou přesností (odchylkou) homogenní. Striktní proces není homogenní.

7 Cvičení 07 – 24. listopadu 2010

7.1 Karhunen-Leove transformace

Dnes se používá pro kompresi obrazu. V obrazu zůstávají data, která na sobě nezávisí. Pokouší se převést obraz na obraz

$$\widetilde{g} = A\left(g - \mu_g\right)$$

, kde A je transformační matice $N^2 \times N^2$ a μ_g konstantní vektor $N^2 \times l$ prostorového průměru obrazu.

7.2 Příklad 1

```
load NP8;
o3
% výpočet autokorelační matice
N=4;
Ro3 = autocorr(N,o3);
Co3 = autocov(N,o3);
% průměr obrazu?
mv = mean(mean(o3));
% o3 centrované
o3c = o3-mv
% autokorelační matice obrázku
% tato matice s tou výš by měli být shodné
Co3c = autocorr(N,o3c);
```

7.3 Příklad 2 – Vylepšování parametru obrazu – kontrastu

Pomocí transformace globálního zesílení obrazu proveďte zesílení kontrastu obrazu f

$$o = \begin{pmatrix} 70 & 100 & 80 & 150 \\ 80 & 130 & 70 & 100 \\ 140 & 120 & 170 & 130 \\ 90 & 180 & 130 & 110 \end{pmatrix}$$

$$g(x,y) = A[f(x,y) - m] + m$$

$$m = \frac{1800}{16} = 112,5$$

Zvolíme $A = 2$, dle hodnoty 180 z matice (aby při zvolení A=2 nám nepřesáhla hodnota hodnotu 255 (8 bitová hloubka barev)).

$g_{11} = 2(70-112.5)+112.5 = 27.5 g_{12} = 2(100-112.5)+112.5 = 87.5 g_{42} = 2(180-112.5)+112.5 = 247.$

7.4 Příklad 3 – bude u zkoušky!

Pomocí transformace lokálního zesílení proveďte zvýšení kontrastu obrazu f. Obraz je shodný, obraz rozdělte na 4 části, vypočítejte jednotlivá zesílení v jednotlivých částech v závislosti na směrodatné odchylce, podle vztahu $A = k * \frac{m}{\sigma}$

$$o = \left(\begin{array}{cc|cc} 70 & 100 & 80 & 150 \\ 80 & 130 & 70 & 100 \\ \hline 140 & 120 & 170 & 130 \\ 90 & 180 & 130 & 110 \end{array}\right)$$

7.5 Příklad 4 – může být u zkoušky!

V programu Matlab proveďte manipulace s histogramy obrazů `pout.tif`, `football.jpg`

```
clear all
% načtení obrázků
I = imread('pout.tif');
% zobrazení
imshow(I)
% zobrazení histogramu(obrázek, počet hodnot v histogramu)
imhist(I,128);
% equalization - vyrovnání histogramu
J = histeq(I);
imhist(J,128);
% zobraznеí 2 obrázků vedle sebe
imshow(J), figure, imhist(J);
% přizpůsobení mezí úzkého histogramu
J = imadjust(I,[0.3 0.5],[0.1 0.75]);
imshow(J)
```

7.6 Příklad 5 – změna kontrastu barevného obrázku – též ke zkoušce

```
% načtení barevného obrázku
rgb1 = imread('football.jpg');
imshow(rgb1);
% zobrazení červené složky
R1 = rgb1(:,:,1);
imshow(R1);
imhist(R1);
% zobrazení zelené složky
G1 = rgb1(:,:,2);
imshow(G1);
imhist(G1);
```

```
% ekvalizace červené složky (zvýšení kontrastu)
% roztažení pixelů z hodnot 0.2 - 0.6 do krajů
R1e = imadjust(R1,[0.2 0.6],[0 1]);
imshow(R1e);
% vrácení nových hodnot do původní matice
rgb1(:,:,1) = R1e;
imshow(rgb1);
```

To samé lze provést jediným příkazem viz níže `rgb2`, kde parametry jsou původní obrázek, původní rozmezí hodnot RGB a nové rozmezí (prázdné vázorky znamenají „do krajů“)

```
rgb2 = imadjust(rgb1,[0.2 0 0;0.6 1 1],[]);
imshow(rgb1), figure, imshow(rgb2);
```

7.7 Příklad 6

```
clear all;
I1 = imread('flower.bmp');
I2 = imread('motyl.bmp');
imshow(I1), figure, imshow(I2);
% smíchání obrázků
I = I1 + I2;
imshow(I), figure, imshow(I2);
% smíchání s různými hodnotami obrazu
I = 0.7*I1+0.3*I2;
imshow(I);
```

7.8 Příklad 7

Do obrazu Cmy343.bmp přidejte Gaussovský a Poissonův šum a odfiltrujte pomocí jednoduchých filtrů.

```
clear all;
I = imread('Cmy343.bmp');
imshow(I);
% konvoluční maska
h1=ones(5,5);
% centrální pixely
h1(3,2:4) = 2
% ---
h2 = ones(9,9);
h2(4:6,4:6) = 2
% děleno počtem vah (25 + 3 s hodnotou 2)
h1 = h1/28
h2 = h2/90
% gaussovský šum
Ing = imnoise(I, 'gaussian', 0,0.02);
imshow(Ing);
% Vyfiltrování šumu
```

```
If1 = imfilter(Ing, h1);
If2 = imfilter(Ing, h2);
imshow(Ing), figure, imshow(If1), figure, imshow(If2);

Inp = imnoise(I, 'poisson');
imshow(Inp);
% odfiltrování šumu
If3 = imfilter(Inp, h1);
If4 = imfilter(Inp, h2);
imshow(Inp), figure, imshow(If3), figure, imshow(If4);

% filtrace medianem
Ig = imread('eight.tif');
imshow(Ig);
% zašumění impulsním šumem
Insp = imnoise(Ig, 'salt & pepper',0.02);
imshow(Insp);
If5 = medfilt2(Insp, [5 5]);
imshow(Insp), figure, imshow(If5);

If7 = imfilter(Insp, h1);
imshow(Insp), figure, imshow(If7);
```

8 Cvičení 08 – 1. prosince 2010

8.1 Příklad 1 – bude u zkoušky!

Navrhněte 2D prostorový filtr dolní propusti s frekvenční oblastí? Aplikujte na zašuměný obraz `Cmy343.bmp`;

```
clear all;
H=zeros(21,21);
% střední oblast, která bude propouštět
H(8:14,8:14) = 1;
% zobrazení 3D obrazu filtru (mesh)
% vypočítání mřížky frekvencí
[f1,F2] = freqspace(21,'mashgrid');
mesh(f1,f2,H);
h=fsamp2(H);
% převádí nový filtr do původní oblastia také ho zobrazí
freqz2(h,[32 32]);
% zobrazení, jak nám filtr vyhladil obraz
I = imread('Cmy343.bmp');
% zašumění obrázku
Ing = imnoise(I, 'gaussian',0,0.02);
% zobrazení zašuměného obrázku
imshow(Ing);
% filtrace prostorové oblasti (konvoluce oblasti obrázku s filtrem H)
If1 = imfilter(Ing,h);
imshow(Ing), figure, imshow(If1);
h=fwind1(H,boxcar(11));
% zobrazení, jak se změní náš původní filtr
freqz2(h,[32 32]);
% vyfiltrování novým filtrem 11x11
If2 = imfilter(Ing, h);
imshow(If1), figure, imshow(If2);

% nový filtr (bartlett)
hw = fwind1(H, bartlett(13));
freqz2(hw,[32 32]);
% filtrace obrázku tímto filtrem
If3 = imfilter(Ing, hw);
imshow(If1), figure, imshow(If2), figure, imshow(If3);
```

8.2 Příklad 2 - pokračuje z prvního

Navrhněte 2D prostorový filtr horní propusti z obdélníkového frekvenčního filtru. Aplikujte na obraz `???.png` (blběček si něco začně mumlat pod vousy)!

```
% filter horní propust a srovnáme s laplaceovým filtrem (něco s procházením 0)
H = ones (7,7);
```

```
H(3:5,3:5) = 0;
[f1,f2] = freqspace(7,'meshgrid');
% zobrazení v 3D pohledu
mesh(f1,f2,H);
% načtení obrázku s mincemi
I1 = imread('coins.png');
imshow(I1);
h = fsamp2(H);
freqz2(h,[32 32]);
If4 = imfilter(I1, h);
imshow(I1), figure, imshow(If4, [0 10]);
% hledání hran, nebo co, matice Laplacian
h2 = [0,1,0;1,-4,1;0,1,0];
If5 = imfilter(I1, h2);
imshow(If4, [0 5]), figure, imshow(If5, [0 10]);
```

8.3 Příklad 3 – navazuje na předchozí, kruhový/obdélníkový u zkoušky

Navrhněte 2D prostorový filtr kruhové dolní propusti, aplikujte na zašuměný obraz.

```
H = zeros(19,19);
[f1,f2] = freqspace(19,'meshgrid');
% omezení mřížky na kruh
d = sqrt(f1.^2+f2.^2) < 0.5;
H(d) = 1;
mesh(f1, f2, H);
I = imread('Cmy343.bmp');
Inp = imnoise(I, 'poisson');
Inp = imnoise(I, 'gaussian',0,0.02);
imshow(Inp);
% převedené do prostorové oblasti + je současně ořezán na velikost 19x19
h = fsamp2(H);
freqz2(h,[32 32]);
% filtrace
If1 = imfilter(Inp, h, 'circular');
imshow(If1);
hw = fwind1(H, hamming(13));
freqz2(hw,[32 32]);
If2 = imfilter(Inp, hw, 'circular');
imshow(If1), figure, imshow(If2);
```

8.4 Příklad 4 – rekonstrukce obrazu

Proveďte rekonstrukci obrazu `onion.png` inverzními filtry programu MatLab

```
clear all;
```

```
I = imread('onion.png');
imshow(I);
% velikost rozmazání delta
del = 30;
% úhel rozmazání
uhel = 10;
% impulsní odezva
psf = fspecial('motion',del, uhel);
% degradace filtru
Id = imfilter(I, psf, 'circular', 'conv');
imshow(Id);
% vrácení původního obrazu, rekontrukce
Ire1 = deconvlucy(Id, psf, 20);
imshow(Ire1);
Ire2 = deconvreg(Id, psf);
imshow(Ire1), figure, imshow(Ire2);
```

9 Cvičení 09 – 8. prosince 2010

9.1 Příklad 1 – bude u zkoušky

Rekonstruujte rozmazaný obraz onion.png Wienerovým filtrem.

```
clear all;
I = imread('onion.png');
% velikost rozmazání delta
del = 30;
% úhel rozmazání
uhel = 10;
% impulsivní degradační odezva (rozmazání obrazu - simulace pohybu kamery)
psfdeg = fspecial('motion',del,uhel);
% aplikace na původní obrázek
Id = imfilter(I, psfdeg, 'circular','conv');
imshow(Id);
% rekontruovaný obraz
Ire1 = deconvwnr(Id, psfdeg, 0);
imshow(Id), figure, imshow(Ire1);
% obrázek bez zbytečného šumu -- ale zase je obraz více rozmazaný...
Ire1 = deconvwnr(Id,psfdeg,0.01);
imshow(Id), figure, imshow(Ire1);
```

9.2 Příklad 2 – bude u zkoušky

Proveďte rekonstrukci černobílého zašuměného monochromatického obrazu `cameraman.tif` Wienerovým filtem, srovnejte s filtem Lucy.

```
clear all;
I = im2double(imread('cameraman.tif'));
imshow(I);
% velikost rozmazání delta
del = 21;
% úhel rozmazání
uhel = 10;
% definování rozmazání příkazem fspecial
psf = fspecial('motion', del, uhel);
% aplikace na původní obrázek
Id = imfilter(I, psf, 'circular', 'conv');
% zašumění obrázku (variance of noise, druhá mocnina směrodatné odchylky šumu)
varn = 0.0001;
% degradace obrazu - rozmazáním a šumem
Idn = imnoise(Id, 'gaussian', 0, varn);
imshow(Idn);
% poměr signálu k šumu
NSR = 0;
NSR1 = 0;
```

```
Ire1 = deconvwnr(Idn, psf, NSR1);
imshow(Ire1);
% cosi
NSR2 = varn/var(I(:))
Ire2 = deconvwnr(Idn, psf, NSR2);
imshow(Ire2);

Ire3 = deconvlucy(Idn, psf, 10);
Ire3 = deconvlucy(Idn, psf, 10);
imshow(Ire3), figure, imshow(Ire2);
```

9.3 Příklad 3 – pro zajímavost

Rekonstrukce barevného rozmazaného obrazu Wienerovým filtrem . . .

```
clear all;
I = imread('board.tif');
imshow(I);
% oříznutí a posunutí velkého obrázku v RGB
I = I(50+[1:256], 2+[1:256], :);
imshow(I);
% velikost rozmazání delta
del = 10;
% úhel rozmazání
uhel = 20;
% definování rozmazání příkazem fspecial
psf = fspecial('motion', del, uhel);
% degradovaný obraz, rozmazaný
Id = imfilter(I, psf, 'symmetric', 'conv');
imshow(Id);
% poissonovo rozostření/rozmazání/zašumění?
Idn = imnoise(Id, 'poisson');
imshow(Idn);

Ire1 = deconvwnr(Idn, psf, 0.01);
imshow(Idn), figure, imshow(Ire1);

Ire2 = deconvlucy(Idn, psf, 10);
imshow(Ire2), figure, imshow(Ire1);
```

9.4 Segmentace obrazu

K segmentaci je třeba program `Anatis 2`. Používá se funkce `segprah.m`

```
function Imseg=segprah(M,N,prh, inmat)
% výpočet obrazu segmentace prahováním

Imseg = zeros(M,N);
```

```
for i=1:M
  for j=1:N
    jas = inmat(i,j);
    if(jas < prh)Imseg(i,j)=10;
      else Imseg(i,j) = jas;
    end;
  end;
end;
```

9.5 Příklad 4

V programu MatLab proveďte segmentaci zrnek rýže `rice.png`.

```
clear all;
I = imread('rice.png');
imshow(I);
% zobrazení histogramu
imhist(I);
prah = 145;

Iseg = segprah(256,256,prah, I);
imshow(Iseg,[1 200]);
% rozdělení obrazu na více částí
imhist(I([1:192],:));
% lokální práh nastavíme někde kolem 140
pl1 =140;
imhist(I([193:256],:));
pl2 = 115;
%
Iseg1 = segprah(192,256,pl1, I);
imshow(Iseg1,[1 200]);
Iseg1 = segprah(256-192,256,pl1, I);
```

Dokončit – je chybně funkce, jelikžo potřebujeme segmentovat spodní část obrázku, kdežto ve funkci se segmentují horní řádky. Opravit funkci. Řešením by bylo u matice I.