ROZ2 - Cv. 1 - Dekonvoluce - výsledky

Adam Novozámský, novozamsky@utia.cas.cz
Honza Kotera, kotera@utia.cas.cz

27. října 2011

Maska Gaussiánu

1. D-dimenzionální Gaussova funkce:

Maska Gaussiánu

1. D-dimenzionální Gaussova funkce:
   - $f(x,y) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{\left[-\frac{1}{2}(x-\mu)^t\Sigma^{-1}(x-\mu)\right]}$

     d... počet dimenzí
     d... vektor střední hodnoty
     $()^t$... transpozice vektoru
     Σ... kovarianční matice-diagonální $\Sigma_{ii} = \sigma_i^2$
     $||$... determinant

2D Gaussova funkce

1. $|\Sigma|$

2D Gaussova funkce

1. $|\Sigma|$
   - $\det \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} = \sigma_1^2 \sigma_2^2$

2D Gaussova funkce

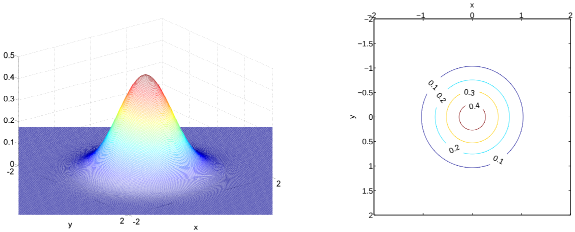
1. $|\Sigma|$
   - $\det \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} = \sigma_1^2 \sigma_2^2$
2. $(x - \mu)^t \Sigma^{-1} (x - \mu)$

2D Gaussova funkce

1. $|\Sigma|$
   - $\det \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} = \sigma_1^2 \sigma_2^2$
2. $(x-\mu)^t \Sigma^{-1} (x-\mu)$
   - $\begin{pmatrix} x_1-\mu_1 & x_2-\mu_2 \end{pmatrix} \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}^{-1} \begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix} =$
     $\begin{pmatrix} x_1-\mu_1 & x_2-\mu_2 \end{pmatrix} \begin{pmatrix} 1/\sigma_1^2 & 0 \\ 0 & 1/\sigma_2^2 \end{pmatrix} \begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix} =$
     $\begin{pmatrix} \frac{x_1-\mu_1}{\sigma_1^2} & \frac{x_2-\mu_2}{\sigma_2^2} \end{pmatrix} \begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix} =$
     $\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}$

2D Gaussova funkce - symetrická (izotropní)

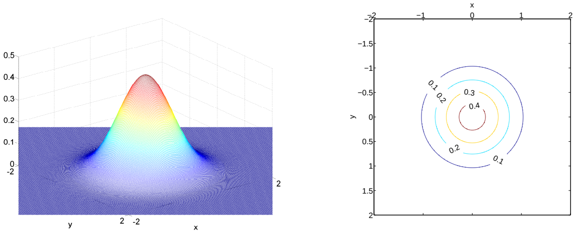
1. $\sigma_x = \sigma_y = \sigma$



Obrázek: $\{\mu_x, \mu_x, \sigma\} = \{0, 0, 0.6\}$

2D Gaussova funkce - symetrická (izotropní)

1. $\sigma_x = \sigma_y = \sigma$

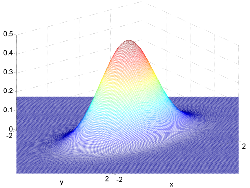


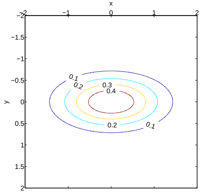
Obrázek: $\{\mu_x, \mu_x, \sigma\} = \{0, 0, 0.6\}$

- $f(x,y) = \dfrac{1}{2\pi\sigma^2} e^{-\dfrac{(x-\mu_x)^2 + (y-\mu_y)^2}{2\sigma^2}}$

2D Gaussova funkce - asymetrická (anizotropní)

1. $\sigma_x \neq \sigma_y$

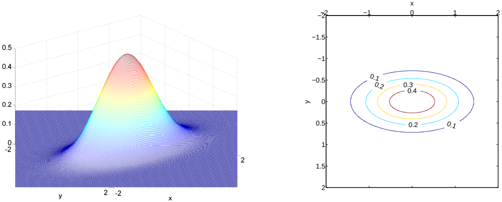




Obrázek: $\{\mu_x, \mu_x, \sigma_x, \sigma_y\} = \{0, 0, 0.8, 0.4\}$

2D Gaussova funkce - asymetrická (anizotropní)

1. $\sigma_x \neq \sigma_y$



Obrázek: $\{\mu_x, \mu_x, \sigma_x, \sigma_y\} = \{0, 0, 0.8, 0.4\}$

- $f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right)}$

Maska Gaussiánu

```
function G = gauss(N, sigma)
npul = (N-1)/2;
[x,y] = meshgrid(-npul:npul);
G=1/(2*pi*sigma^2)*exp(-(x.^2 + y.^2)/(2*sigma^2));
G = G / sum(G(:));
```

Maska Gausiánu

```
function G = gauss(N, sigma)
npul = (N-1)/2;
[x,y] = meshgrid(-npul:npul);
G=exp(-(x.^2 + y.^2)/(2*sigma^2));
G = G / sum(G(:));
```

```
[x, y] = meshgrid(-3:3);
```

x =

-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3
-3	-2	-1	0	1	2	3

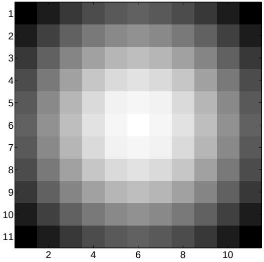
y =

-3	-3	-3	-3	-3	-3	-3
-2	-2	-2	-2	-2	-2	-2
-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3

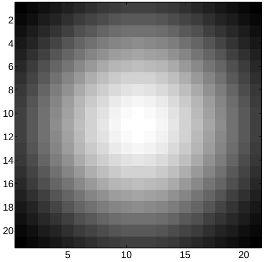
x.^2 + y.^2 =

18	13	10	9	10	13	18
13	8	5	4	5	8	13
10	5	2	1	2	5	10
9	4	1	0	1	4	9
10	5	2	1	2	5	10
13	8	5	4	5	8	13
18	13	10	9	10	13	18

Maska Gausiánu



(3.1) gauss(11,5)



(3.2) gauss(21,7)

1. definice SNR:

1. definice SNR:
   - $SNR = 10 \log \left(\frac{D(f)}{D(n)} \right)$
     $D(f) \approx \sigma_f^2 \qquad D(n) \approx \sigma_n^2;$

1. definice SNR:
   - $SNR = 10 \log \left(\frac{D(f)}{D(n)} \right)$
     $D(f) \approx \sigma_f^2 \qquad D(n) \approx \sigma_n^2;$
2. Odvození

1. definice SNR:
   - $SNR = 10\log\left(\frac{D(f)}{D(n)}\right)$
   $D(f) \approx \sigma_f^2 \qquad D(n) \approx \sigma_n^2;$
2. Odvození
   - $\frac{SNR}{10} = \log\left(\frac{D(f)}{D(n)}\right)$

1. definice SNR:
   - $SNR = 10 \log\left(\frac{D(f)}{D(n)}\right)$
     $D(f) \approx \sigma_f^2 \qquad D(n) \approx \sigma_n^2;$
2. Odvození
   - $\frac{SNR}{10} = \log\left(\frac{D(f)}{D(n)}\right)$
   - $10^{\frac{SNR}{10}} = \frac{D(f)}{D(n)}$

1. definice SNR:
   - $SNR = 10 \log \left(\frac{D(f)}{D(n)} \right)$
     $D(f) \approx \sigma_f^2 \qquad D(n) \approx \sigma_n^2;$
2. Odvození
   - $\frac{SNR}{10} = \log \left(\frac{D(f)}{D(n)} \right)$
   - $10^{\frac{SNR}{10}} = \frac{D(f)}{D(n)}$
   - $D(n) = \frac{D(f)}{10^{\frac{SNR}{10}}}$

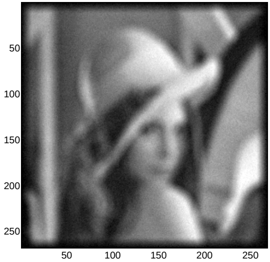
1. definice SNR:
   - $SNR = 10 \log\left(\frac{D(f)}{D(n)}\right)$
     $D(f) \approx \sigma_f^2 \qquad D(n) \approx \sigma_n^2;$
2. Odvození
   - $\frac{SNR}{10} = \log\left(\frac{D(f)}{D(n)}\right)$
   - $10^{\frac{SNR}{10}} = \frac{D(f)}{D(n)}$
   - $D(n) = \frac{D(f)}{10^{\frac{SNR}{10}}}$
   - $\sigma_n \approx \sqrt{\frac{\sigma_f^2}{10^{\frac{SNR}{10}}}}$ ...nápověda funkcí: `var()`, `randn()`

Bílý šum

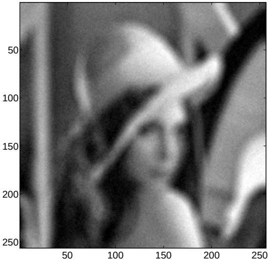
```
function W = whiteNoise(I, SNR)
MinI = min(I(:));
MaxI = max(I(:));
S = sqrt(var(I(:))/(10^(SNR/10)));
W = I + S*randn(size(I));
W(W<MinI) = MinI;
W(W>MaxI) = MaxI;
```

Poškození obrázku

```
function D = demage(I, H, SNR)
D = conv2(I,H);
D = whiteNoise(D, SNR);
```



(3.3) demage(I,kruh(7,15),20)



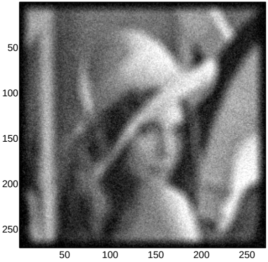
(3.4) demage1(I,kruh(7,15),20)

Poškození obrázku

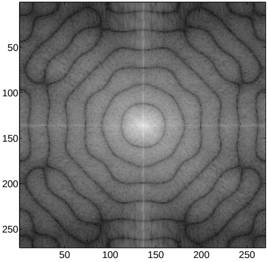
```
function D = demage1(I, H, SNR)
D = conv2(I,H,'same')./conv2(ones(size(I)),H,'same');
D = whiteNoise(D, SNR);
```

Rozmazání obrázku - kruhem

▶



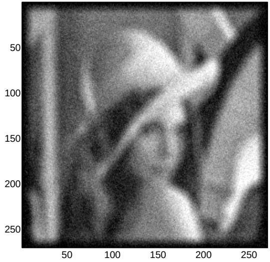
(3.5) demage(I,kruh(7,15),12)



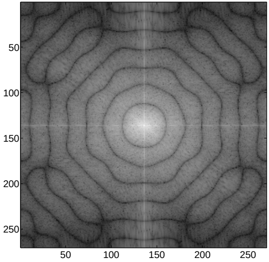
(3.6) fr. spektrum při SNR=100

Rozmazání obrázku - kruhem

▶



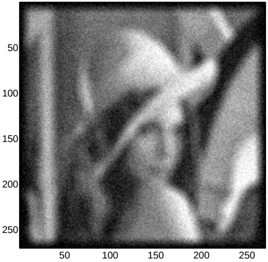
(3.7) demage(I,kruh(7,15),12)



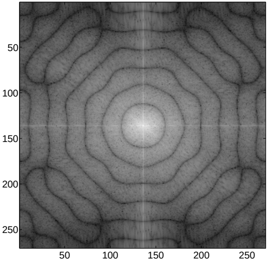
(3.8) fr. spektrum při SNR=100

- D=demage(I,kruh(7,15),100);

Rozmazání obrázku - kruhem



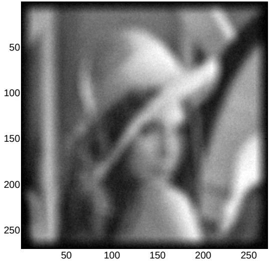
(3.9) demage(I,kruh(7,15),12)



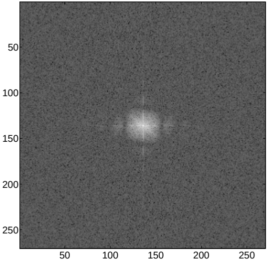
(3.10) fr. spektrum při SNR=100

- D=demage(I,kruh(7,15),100);
- F= fftshift(log(abs(fft2(D))+1));

Rozmazání obrázku - Gaussianem



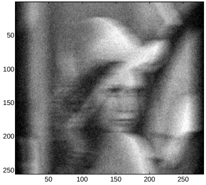
(3.11) demage(I,gauss(15,7),20)

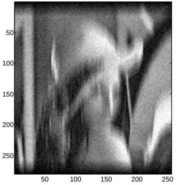


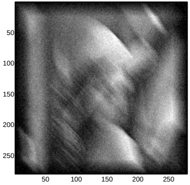
(3.12) fr. spektrum

zobr(fftshift(log(abs(fft2(demage(I,gauss(15,7),20)))+1)));

Rozmazání obrázku - pohybem



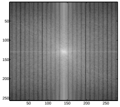


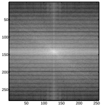


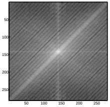
(3.13) horizontálním: demage(I,ones(25,1),12)

(3.14) vertikálním: demage(I,ones(1,25),12)

(3.15) diagonálním: demage(I,eye(25),12)







horní řádek - demage(I,gauss(11,3),SNR)
dolní řádek - demage(I,gauss(21,7),SNR)
SNR = 50, 20, 10, 5

Inverzní filtr

1. konvoluce:

Inverzní filtr

1. konvoluce:
   - $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$
     $* : L_1 \times L_1 \to L_1$

Inverzní filtr

1. konvoluce:
   - $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$
     $* : L_1 \text{x} L_1 \rightarrow L_1$
2. konvoluční teorém:

Inverzní filtr

1. konvoluce:
   - $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$
     $* : L_1 \text{x} L_1 \to L_1$
2. konvoluční teorém:
   - $\mathcal{F}(f * g) = [\mathcal{F}(f)].[\mathcal{F}(g)] = F.G$

Inverzní filtr

1. odvození:

Inverzní filtr

1. odvození:
   - $Z = U.H$

Inverzní filtr

1. odvození:
   - $Z = U.H$
   - $\rightarrow U = \dfrac{Z}{H}$

Inverzní filtr

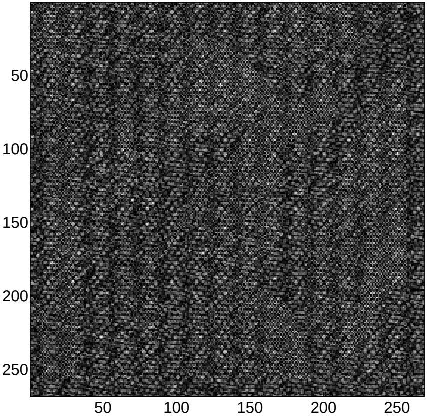
1. odvození:
   - $Z = U.H$
   - $\rightarrow U = \dfrac{Z}{H}$
   - $\rightarrow u = \mathcal{F}^{-1}\left(\dfrac{Z}{H}\right)$

Inverzní filtr

```
function u = inverse(z, h)
Z = fft2(z);
H = fft2(h, size(z,1), size(z,2));
U = Z ./ H;
u = abs(ifft2(U));
```

Inverzní filtr - SNR = 90, 80, 70; K=kruh(7,13);

Inverzní filtr - SNR = 60; K=kruh(7,13);



Definice Wienerova filtru

1. získaný odhad má mít minimální odchylku od originálu

Definice Wienerova filtru

1. získaný odhad má mít minimální odchylku od originálu
   - $E(\| f' - f \|^2 \longrightarrow min)$ střední kvadratická chyba
     E...střední hodnota
     f'...odhad
     f...originál

Definice Wienerova filtru

1. získaný odhad má mít minimální odchylku od originálu
   - $E(\| f' - f \|^2 \longrightarrow min)$ střední kvadratická chyba
     E...střední hodnota
     f'...odhad
     f...originál
2. má to být lineární filtr, tedy to má být násobení ve frekvenční oblasti

Definice Wienerova filtru

1. získaný odhad má mít minimální odchylku od originálu
   - $E(\| f' - f \|^2 \longrightarrow min)$ střední kvadratická chyba
     E...střední hodnota
     f'...odhad
     f...originál
2. má to být lineární filtr, tedy to má být násobení ve frekvenční oblasti
   - $F' = G.R$
     G...je zašuměný obrázek
     R...je transformační matice

Definice Wienerova filtru

1. odvozen filtr:
$$R(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$

Definice Wienerova filtru

1. odvozen filtr:
$$R(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$
2. $S_n(u,v)/S_f(u,v) \approx SNR^{-1}$

Definice Wienerova filtru

1. odvozen filtr:
$$R(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$
2. $S_n(u,v)/S_f(u,v) \approx SNR^{-1}$
3. $C.\bar{C} = |C|^2$

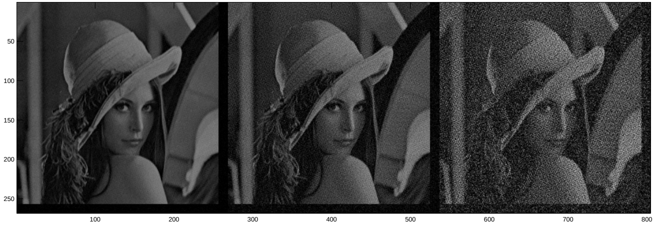
Definice Wienerova filtru

1. odvozen filtr:
$$R(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$
2. $S_n(u,v)/S_f(u,v) \approx SNR^{-1}$
3. $C.\bar{C} = |C|^2$
   - Matlab funkce: conj()

Wienerův filtr

```
function R = wiener(g, h, SNR)
H = fft2( h, size(g,1), size(g,2));
W = conj(H) ./ (abs(H).^2 + 1/SNR);
FI = fft2(g) .* W;
R = abs(ifft2(FI));
```

Wienerův filtr - SNR = 60, 50, 40; K=kruh(7,13);



Určete typ poškození u obrázků yiXX.pgm a vylepšete je

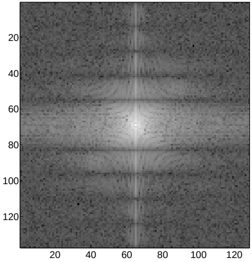
1. Použijte tyto příkazy:

Určete typ poškození u obrázků yiXX.pgm a vylepšete je

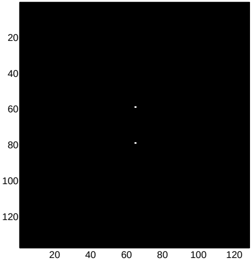
1. Použijte tyto příkazy:
   - ```
     m1 = log(abs(fft2(f).^2));
     m2 = real(fft2(m1));
     mi = min(m2(:));
     m3 = m2 < 0.9*mi;
     ```
     ```

y1XX.pgm - určení druhu poškození

▶



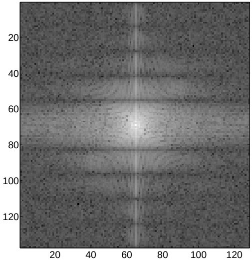
(3.19) zobr(fftshift(m1));



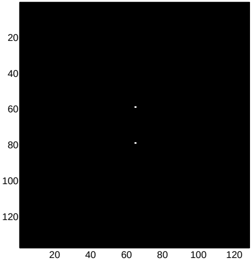
(3.20) zobr(fftshift(m3));

y1XX.pgm - určení druhu poškození

▶



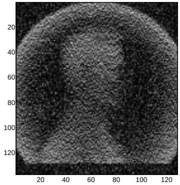
(3.21) zobr(fftshift(m1));



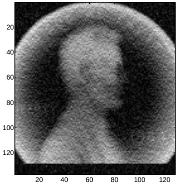
(3.22) zobr(fftshift(m3));

- ⇒ vertikální poškození - ones(10,1)

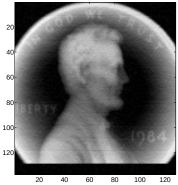
y1XX.pgm - zobr(wiener(y1XX,ones(10,1),XX));



(3.23) y120



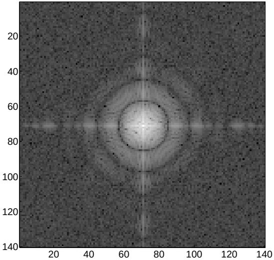
(3.24) y130



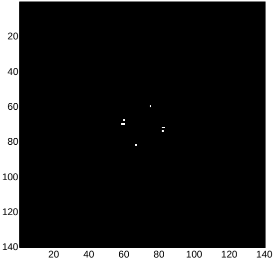
(3.25) y150

y2XX.pgm - určení druhu poškození

▶



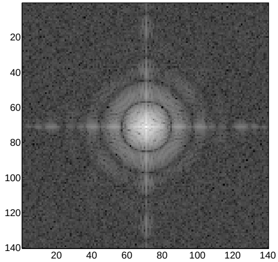
(3.26) zobr(fftshift(m1));



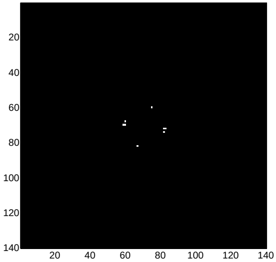
(3.27) zobr(fftshift(m3));

y2XX.pgm - určení druhu poškození

▶



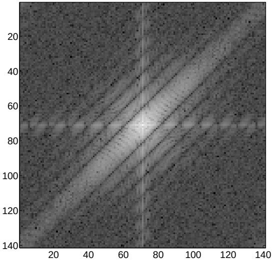
(3.28) zobr(fftshift(m1));



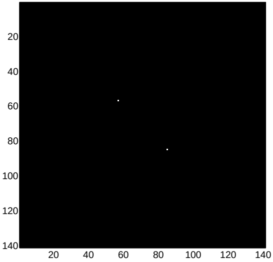
(3.29) zobr(fftshift(m3));

- $\Rightarrow$ obrázky jsou rozostřené jinou implementací kruhu (s odchylkami hranice), takže zde nám to moc nechodí :(

y3XX.pgm - určení druhu poškození



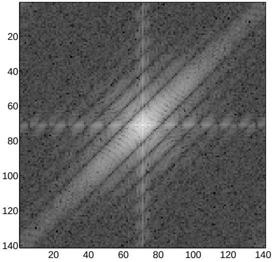
(3.30) zobr(fftshift(m1));



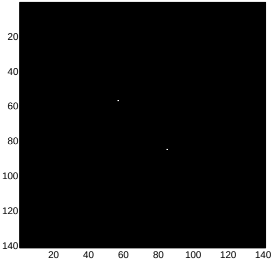
(3.31) zobr(fftshift(m3));

y3XX.pgm - určení druhu poškození

▶



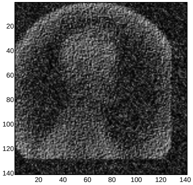
(3.32) zobr(fftshift(m1));



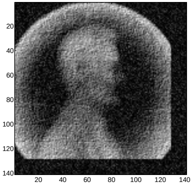
(3.33) zobr(fftshift(m3));

- $\Rightarrow$ diagonální poškození - eye(14)

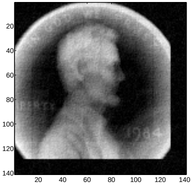
y3XX.pgm - zobr(wiener(y1XX,ones(10,1),XX));



(3.34) y320



(3.35) y330



(3.36) y350

ROZ2 - Cv. 1 - Dekonvoluce - výsledky

Adam Novozámský, novozamsky@utia.cas.cz
Honza Kotera, kotera@utia.cas.cz

27. října 2011