Přednáška 3

Aritmeticke operace s obrazky

Obrazky chapeme, jako matice, je mozne s nimi pracovat jako s ciselnymi maticemi.

Priklad pouziti:

Soucet - morfing obrazku

Rozdil - hledani zmen v obraze

Nasobeni - vynasobeni obrazku nejakou maskou - region of interest (ROI)

Podil - Odstraneni stinu (pokud zname jeho funkci) Rozdil

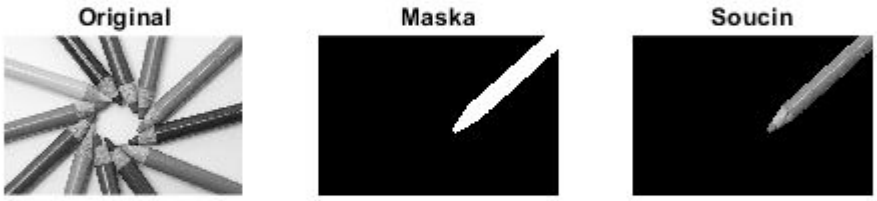
```
% nacteni obrazku a vytvoreni druheho
I = imread('pastelky_gray.png');
I2 = bitand(I,254); %odstraneni informace z nejmene vyznamneho bitu

figure
subplot(1,3,1)
imshow(I)
title('Original')
subplot(1,3,2)
imshow(I2)
title('Upraveny')
subplot(1,3,3)
imshow(I-I2,[])
title('Rozdil')
```


```
% Na prvni pohled obrazky vypadají stejne na 3. je videt, ze rozdily mezi nimi
% jsou (bile pixely)
%
```

Soucin

pro nasobeni matic prvek po prvku se pouziva .* (* predstavuje klasicke nasobeni matic). Stejne tak u deleni.

```
I = imread('pastelky_gray.png');

% ROI maska
maska = (rgb2gray(imread('maska.png')))/255;

I2 = maska.*I;
figure
subplot(1,3,1)
imshow(I)
title('Original')
subplot(1,3,2)
imshow(maska,[])
title('Maska')
subplot(1,3,3)
imshow(I2)
title('Soucin')
```



UKOL 1

Prolinani dvou obrazku - soucet dvou obrazku, ktere jsou vynásobeny koeficienty predstavujici jednotlive pruhlednosti. (soucet by mel byt roven jedne. Tedy: pruhlednost * I + (1-pruhlednost) * J

```
I = rgb2gray(imread('pr1a.png'));
J = imread('pr1b.png');

% TODO
soucet=[];

subplot(1,3,1)
imshow(I)
title('Obrazek 1')
subplot(1,3,2)
imshow(J)
title('Obrazek 2')
subplot(1,3,3)
imshow(soucet)
title('Soucet')
```

Soucet

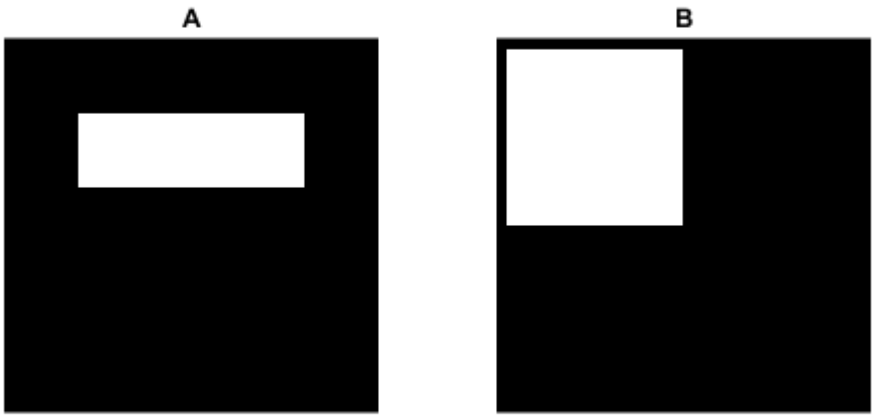
Obrazek 1

Obrazek 2

Logicke operace and, or, xor, not

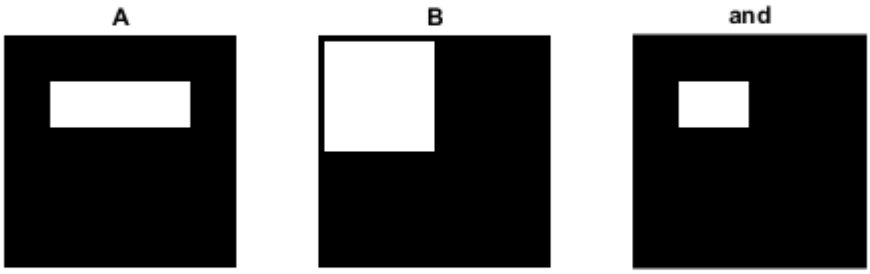
operace nad cernobilymi obrazky (vetsinou predstavujici nejakou oblast)

```
%vytvoreni 2 obrazku
A = logical(zeros(500));
B = logical(zeros(500));

A(100:200, 100:400) = 1;
B(15:250, 15:250) = 1;

figure
subplot(1,2,1)
imshow(A)
title('A')
subplot(1,2,2)
imshow(B)
title('B')
```



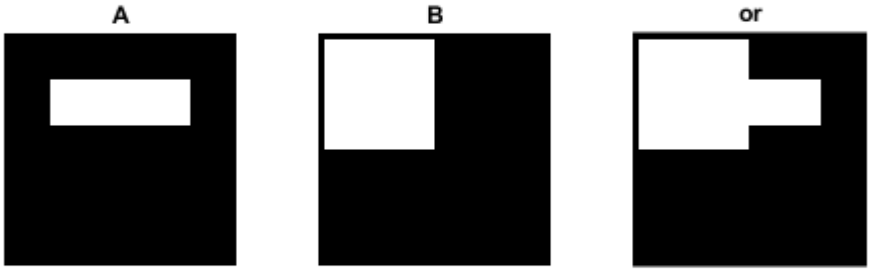
AND

```
figure
subplot(1,3,1)
imshow(A)
title('A')
subplot(1,3,2)
imshow(B)
title('B')
subplot(1,3,3)
imshow(and(A,B))
title('and')
```



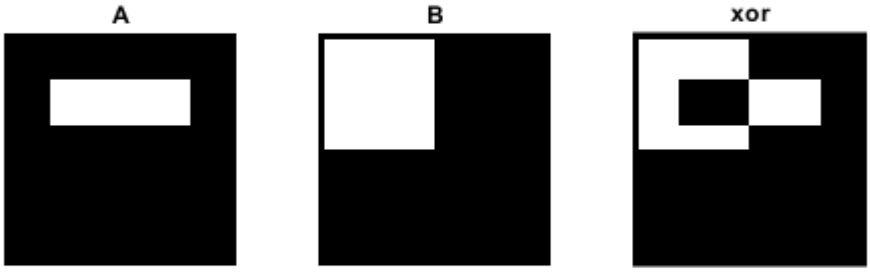
OR

```
figure
subplot(1,3,1)
imshow(A)
title('A')
subplot(1,3,2)
imshow(B)
title('B')
subplot(1,3,3)
imshow(or(A,B))
title('or')
```



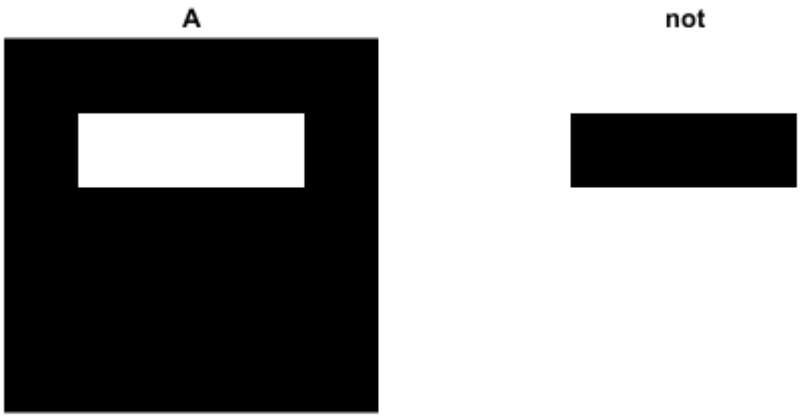
XOR

```
figure
subplot(1,3,1)
imshow(A)
title('A')
subplot(1,3,2)
imshow(B)
title('B')
subplot(1,3,3)
imshow(xor(A,B))
title('xor')
```



NOT

```
figure
subplot(1,2,1)
imshow(A)
title('A')
subplot(1,2,2)
imshow(not(A))
title('not')
```



Geometricke transformace

```
% tform = maketform(typ, parametry, ...)
% typ : 'affine', 'projective', 'custom', 'box', 'composite'

%'custom' pro uzivatelsky definovane transformacni funkce
% tform = maketform('custom', ndim_in, ndim_out, dopredna_fce, zpetna_fce, data)

%Priklad:
% (x,y) = T{(w,z)} = (3w,2z)
% (w,z) = T^(-1){(x,y)} = (x/3, y/2)

% dopredna_fce
dopredna_fce = @(wz,tdata) [3*wz(:,1),2*wz(:,2)]
```

```
dopredna_fce = function_handle with value:
    @(wz,tdata)[3*wz(:,1),2*wz(:,2)]
```

```
% zpetna_fce
zpetna_fce = @(xy,tdata) [xy(:,1)/3,xy(:,2)/2]
```

```
zpetna_fce = function_handle with value:
    @(xy,tdata)[xy(:,1)/3,xy(:,2)/2]
```

```
% transformacni struktura
tform1 = maketform('custom',2,2,dopredna_fce, zpetna_fce,[])
```

```
tform1 = struct with fields:
       ndims_in: 2
      ndims_out: 2
    forward_fcn: @(wz,tdata)[3*wz(:,1),2*wz(:,2)]
    inverse_fcn: @(xy,tdata)[xy(:,1)/3,xy(:,2)/2]
          tdata: []
```

aplikace transformace

aplikujeme na body [1, 1] a [3, 2]

XY = tformfwd(WZ,tform) WZ = tforminv(XY,tform)

```
WZ = [1 1;
      3 2];
XY = tformfwd(WZ,tform1)
```

```
XY = 2×2
     3     2
     9     4
```

```
WZ2 = tforminv(XY,tform1)
```

```
WZ2 = 2×2
     1     1
     3     2
```

UKOL 2

Vytvorte transformacni strukturu (tform2) pro funkce

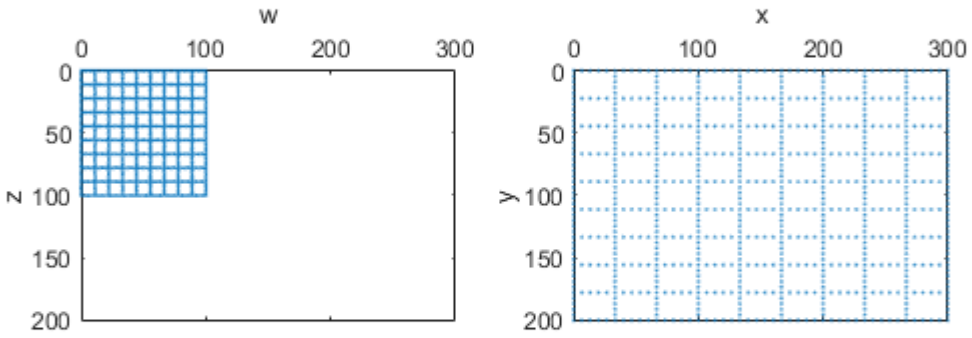
(x,y) = T{(w,z)} = (w+0.4z,z)

(w,z) = T^(-1){(x,y)} = (x-0.4y, y)

vyzkousejte na bodech [1, 1] a [3, 2]

Vizualizace

```
vistform(tform1,pointgrid([0 0; 100 100]));
```




```
%figure, vistform(tform2,pointgrid([0 0; 100 100]));
```

Afinni transformace

pomoci maketform('affine',T) nebo tform = affine2d(T)

```
%afinni transformace odpovidajici tform1
T1 = [3 0 0;
      0 2 0;
      0 0 1];

tform3 = maketform('affine',T1);
WZ = [1 1;
      3 2];
XY = tformfwd(WZ,tform3)
```

```
XY = 2×2
     3     2
     9     4
```

```
WZ2 = tforminv(XY,tform3)
```

```
WZ2 = 2×2
     1     1
     3     2
```

UKOL 3

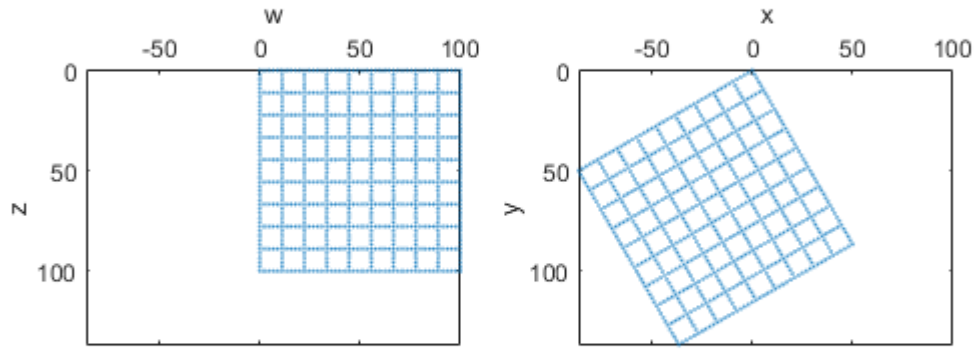
Jak vypada afinni matice odpovidajici tform2

```
%T2 = ..... doplnte
% tform4 = maketform('affine',T2);

% porovnejte s vysledky pro tform2
% WZ = [1 1; 3 2];
% XY = tformfwd(WZ,tform4)
% WZ2 = tforminv(XY,tform4)
```

Afinni tranformace - otoceni

vizualizace

```
uhel = pi/3;
T = [cos(uhel) sin(uhel) 0;
    -sin(uhel) cos(uhel) 0;
    0 0 1];
tform5 = maketform('affine',T);

vistform(tform5,pointgrid([0 0; 100 100]));
```

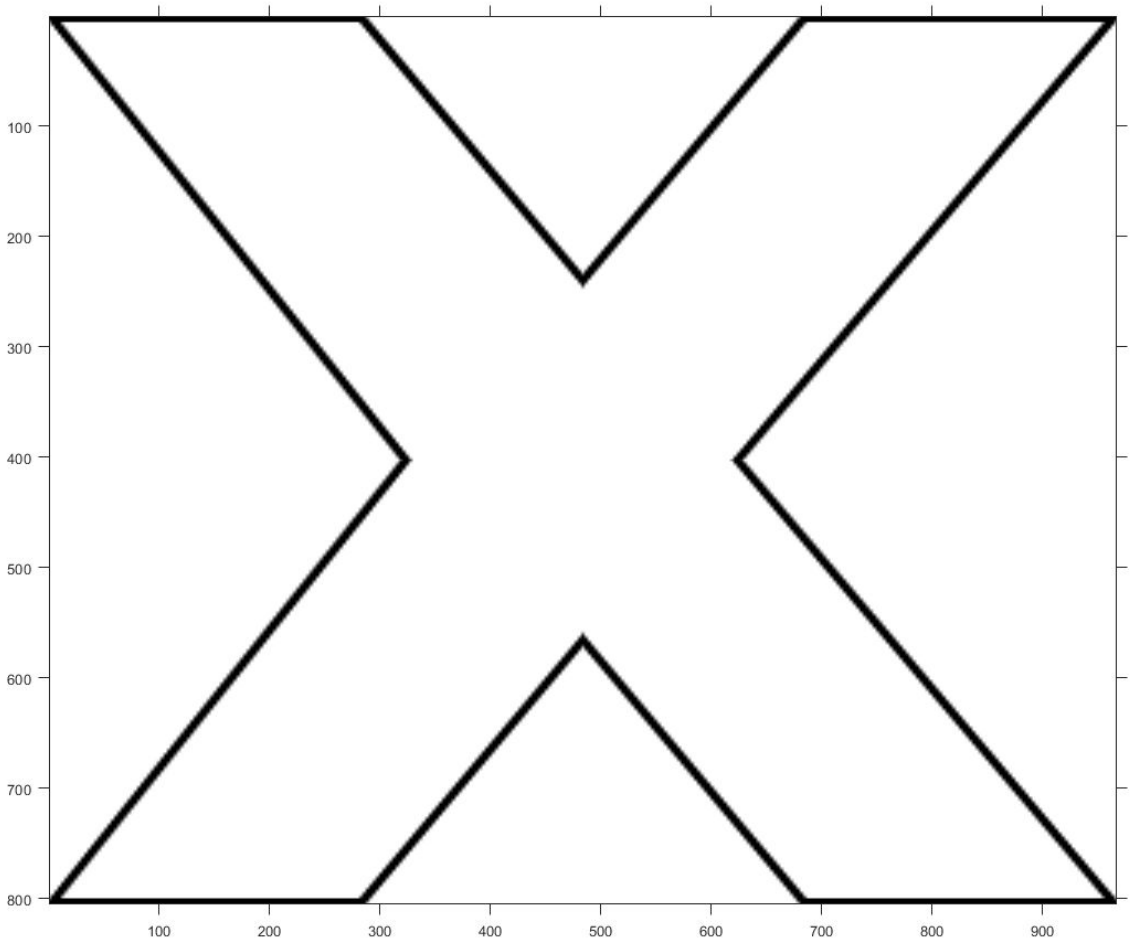


Aplikace tranformace na obrazky

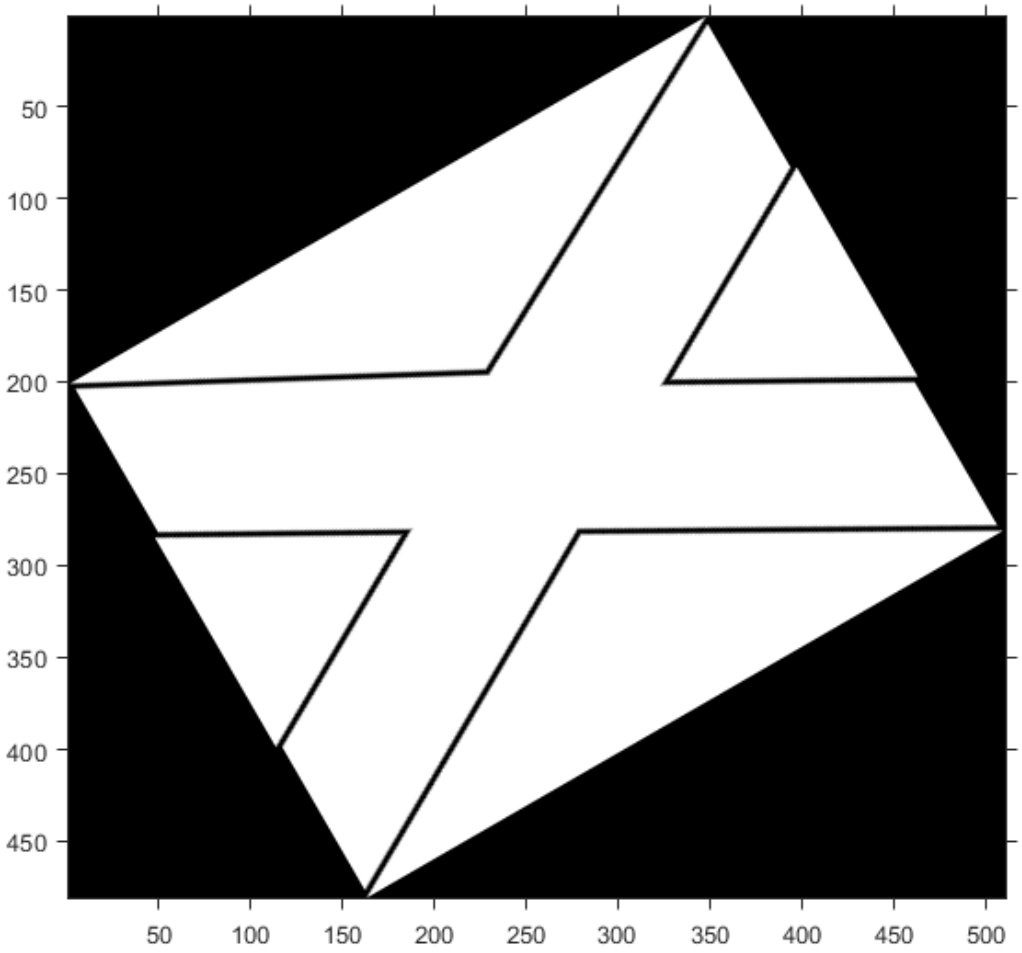
g = imtransform(f,tform); nebo g = imwarp(f,tform);

```
f = imread('X.png');
```

```
g1 = imtransform(f,tform1,'bilinear');
%g2 = imtransform(f,tform2);
%g3 = imtransform(f,tform3);
%g4 = imtransform(f,tform4);
g5 = imtransform(f,tform5);

figure,
imshow(g1);
axis on;
```

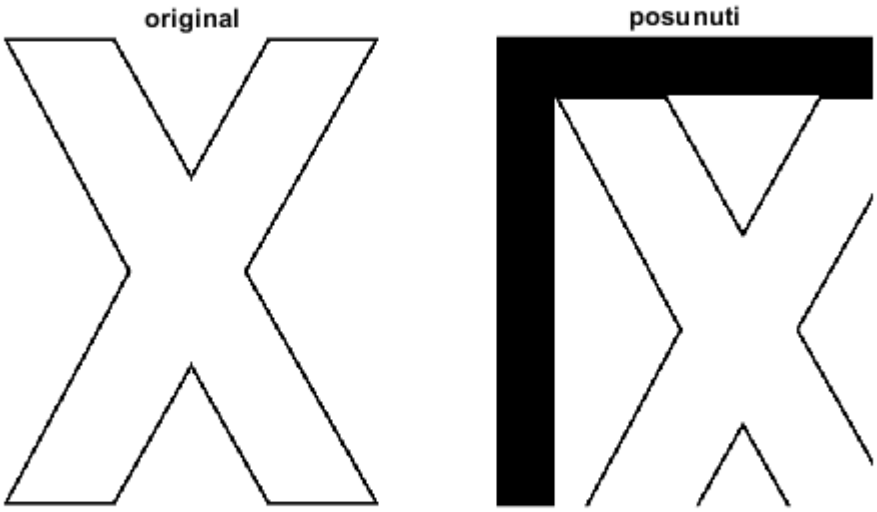



```
figure,
imshow(g5);
axis on;
```



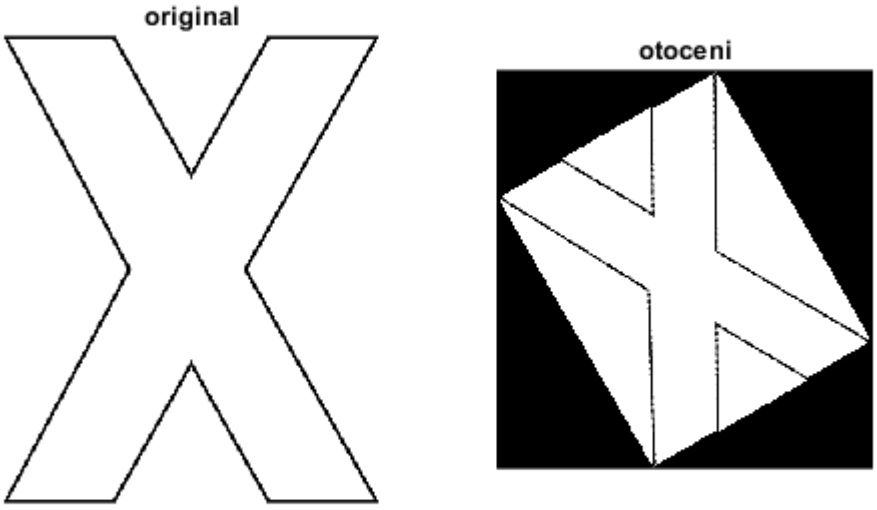
Matlab transformace

```
%imtranslate, imrotate
% Posunuti

X = imread('X.png');
X2 = imtranslate(X, [50,50]);

figure
subplot(1,2,1)
imshow(X)
title('original')
subplot(1,2,2)
imshow(X2)
title('posunuti')
```



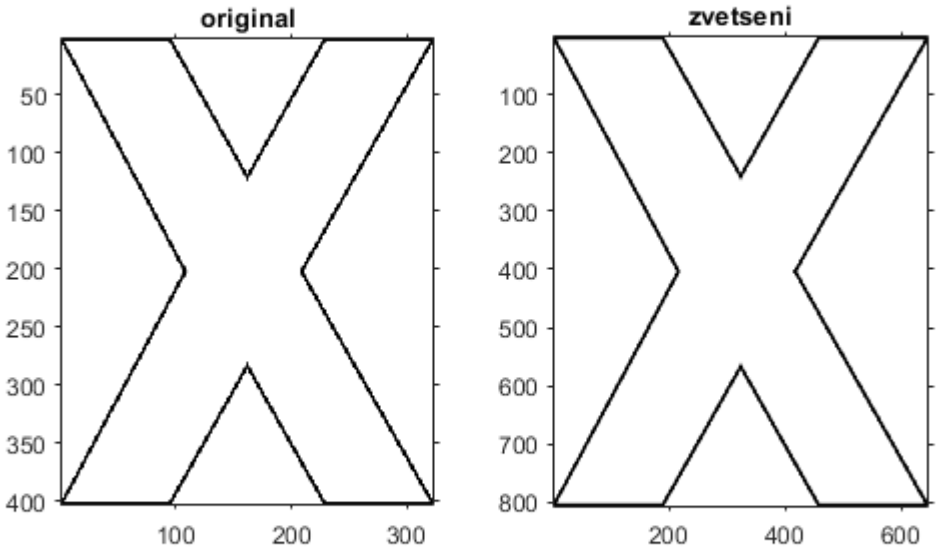
Otoceni

```
X2 = imrotate(X, 30);

figure
subplot(1,2,1)
imshow(X)
title('original')
subplot(1,2,2)
imshow(X2)
title('otoceni')
```



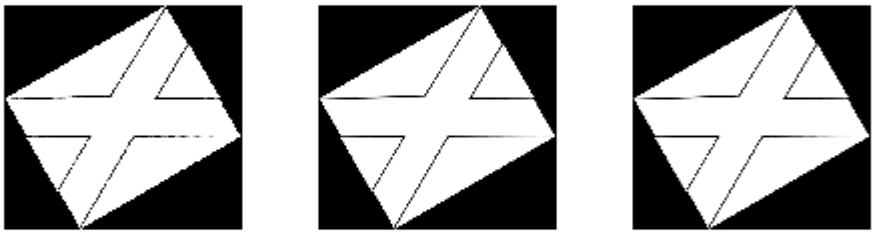
Zmena meritka

```
X2 = imresize(X, 2);

figure
subplot(1,2,1)
imshow(X)
title('original')
axis on
subplot(1,2,2)
imshow(X2)
title('zvetseni')
axis on
```



Interpolace

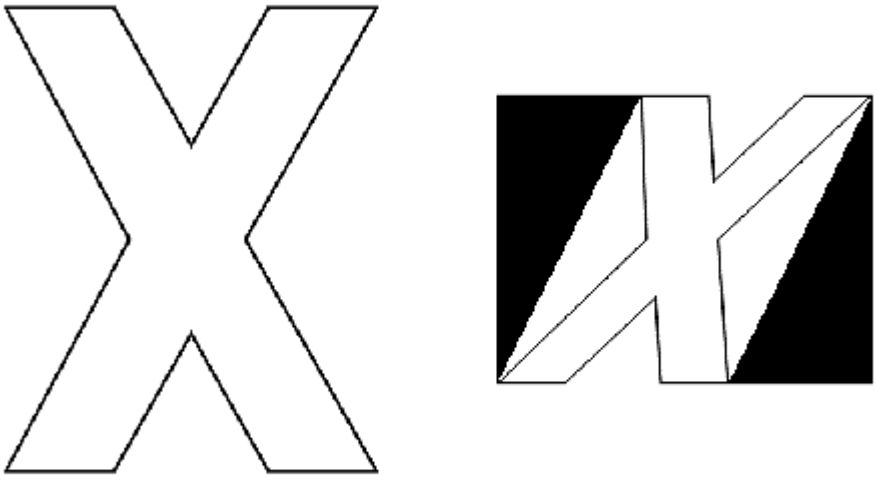
```
g5a = imtransform(X,tform5, 'nearest');
g5b = imtransform(X,tform5, 'bilinear');
g5c = imtransform(X,tform5, 'bicubic');

figure,
subplot(1,3,1), imshow(g5a);
subplot(1,3,2), imshow(g5b);
subplot(1,3,3), imshow(g5c);
```



Registrace

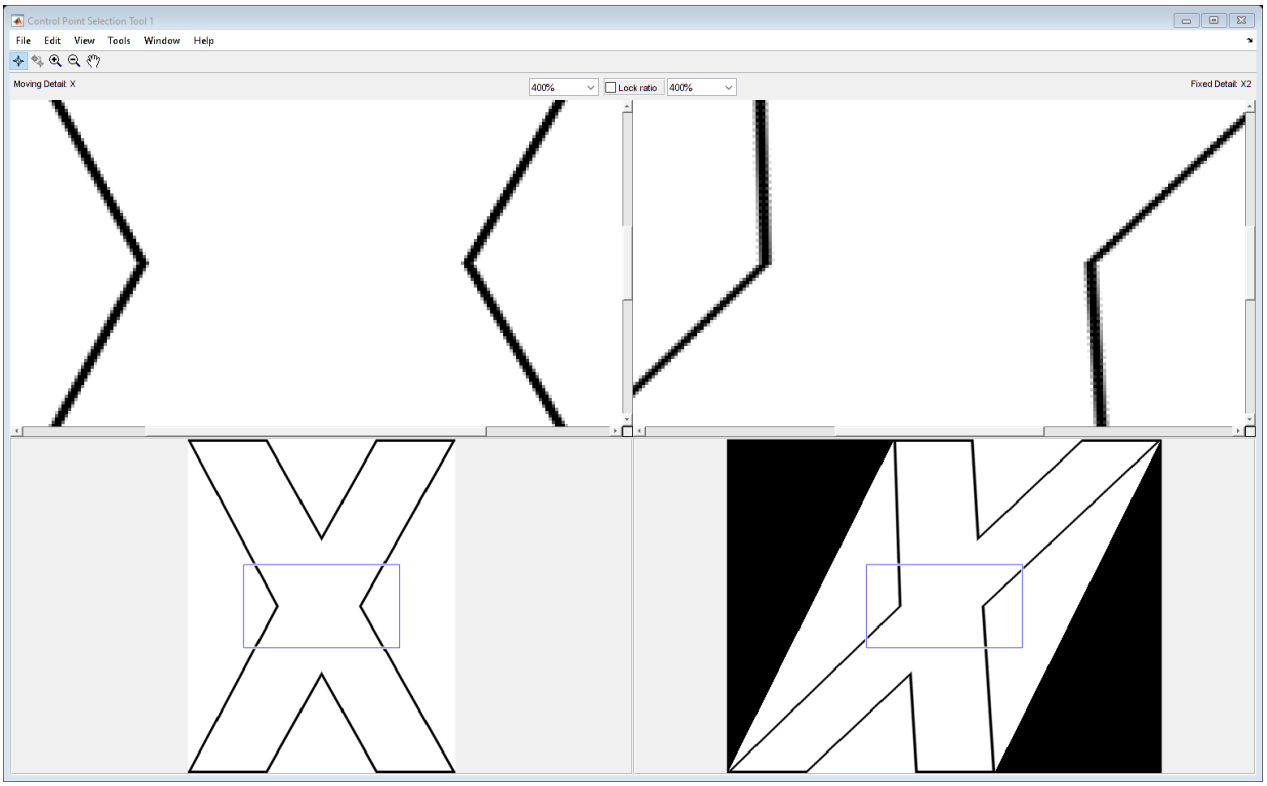
Nacteni obrazku a vytvoreni noveho pomoci afinni trnasformace

```
X = imread('X.png');
tform = affine2d([1 0 0; -0.5 1 0; 0 0 1]);

X2 = imwarp(X,tform);

figure
subplot(1,2,1)
imshow(X)
subplot(1,2,2)
imshow(X2)
```



vyber kontrolnich bodu

```
cpselect(X,X2);
```



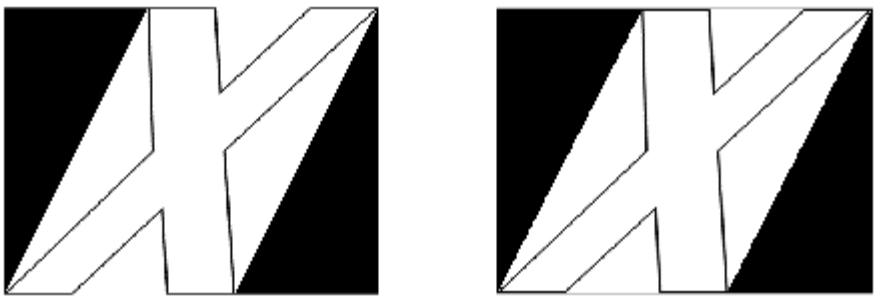
hledani transformace

```
tform2 = fitgeotrans(movingPoints,fixedPoints,'affine')
```

```
tform2 =
  affine2d with properties:

                T: [3×3 double]
    Dimensionality: 2
```

Aplikace nalezene transformace

```
X3 = imwarp(X,tform2);
figure
subplot(1,2,1)
imshow(X2)
subplot(1,2,2)
imshow(X3)
```



UKOL 4

Naprogramujte funkci registrace, ktera jako vstup bere 2 obrazky, ktere obsahuji cerny obdelnik (obdelnik1.png a obdelnik2.png) . Předpokladejme, ze obdelniky na obou obrazcich jsou ty same, jen druhy obdelnik je posunuty a deformovany operaci zvetseni/zmenseni v osach x a y. Funkce vraci 3 hodnoty – vektor posunutí a 2 koeficienty predstavující miru zvetseni/zmensení v ose x a v ose y.

```
% Pro nasledujici matice by byl vystup posunuti = [2,1], zvetseni_x = 1.5,
```

```
% zvetseni_y = 2

M1 = [ 1 1 1 1 1 1;
       1 0 0 1 1 1;
       1 0 0 1 1 1;
       1 1 1 1 1 1;
       1 1 1 1 1 1;
       1 1 1 1 1 1];

M2 = [ 1 1 1 1 1 1;
       1 1 1 1 1 1;
       1 1 1 0 0 0;
       1 1 1 0 0 0;
       1 1 1 0 0 0;
       1 1 1 0 0 0];
```