

# Poročilo 6. vaje pri predmetu OVS

## Geometrijske preslikave slik

Tina Zwitter 64200432

17. november 2020

# 1 Dodajanje interpolacije prvega reda

```
function oImage = transformImage(iType, iImage, iDim, iP, iBgr, iOrder)
%transformImage transformira dano sliko z tipom transformacije.
%vhodni parametri:
% iType - tip tranforamcije 'affine' ali 'radial'
% iImage - slika, podana v matrični obliki
% iDim - dimenzija pikslov podana v [dimX dimY]
% iP - matrika dane transformacije
% iBgr - barva odzadja
% iOrder - željen red interpolacije 0 ali 1
%izhodni parametrrri:
% oImage - transformirana slika

%dimenzija slike
[Y, X] = size(iImage);
%inicializacija izhodne matrike
oImage = ones(Y,X) *iBgr;
for y = 0:Y-1
    for x = 0:X-1
        %koordinate trenutne točke
        pt = [x y].*iDim;
        %afina geometrijska preslikava
        if strcmp(iType, 'affine')
            pt = iP*[pt';1];
            pt = pt(1:2)';
        end
        if strcmp(iType, 'radial')
            U = getRadialValue(pt,iP{1})';
            pt = [U*iP{2}(:,1), U*iP{2}(:,2)];
        end
        pt = pt./iDim;
        %interpolacija reda 0
        if iOrder == 0 %princip najbližjega slikovnega elementa
            px = round(pt);
            if px(1) <X &&px(2)<Y && px(1)>=0 && px(2)>=0% preverimo veljavnost lokacije
                oImage(y+1,x+1) = iImage(px(2)+1,px(1)+1);
            end
        elseif iOrder == 1 %interpolacija 1. reda
            px = floor(pt);
            if px(1) < size(iImage,2) && px(2) < size(iImage,1) && 0<px(1) && 0< px(2)%
                p1 = abs(pt(1)- px(1)) * abs(pt(2)-px(2));
                p2 = abs(pt(1) - (px(1)+1)) * abs(pt(2) -px(2));
                p3 = abs(pt(1) - px(1)) * abs(pt(2) - (px(2) +1));
                p4 = abs(pt(1) - (px(1)+1)) * abs(pt(2) - (px(2) +1));

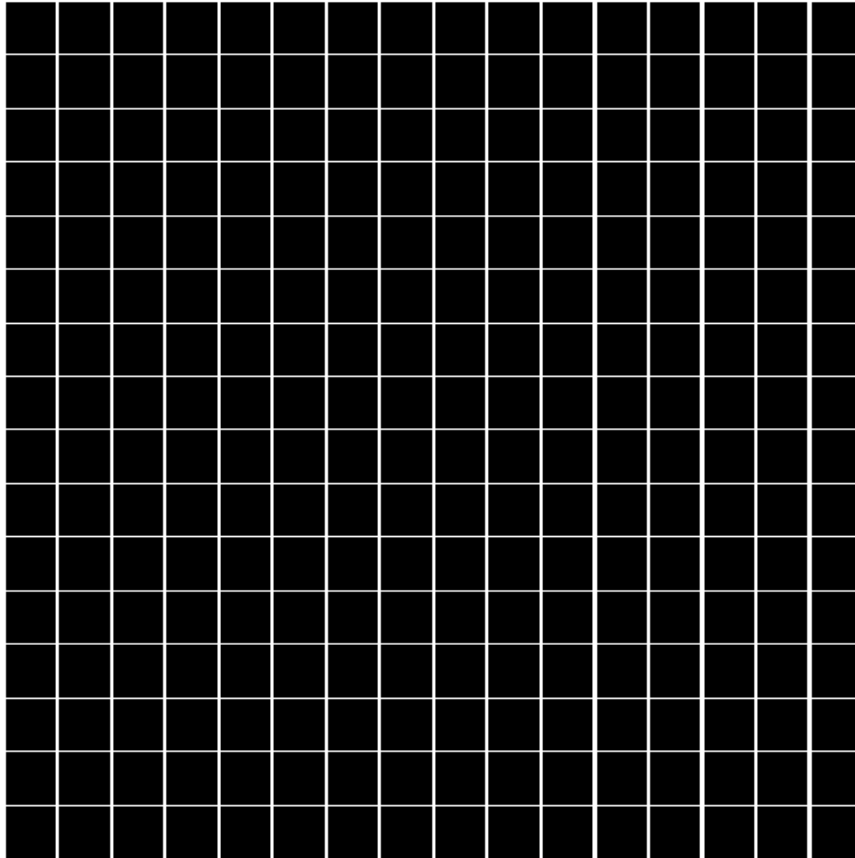
                sa = iImage(px(2),px(1));
                sb = iImage(px(2),px(1)+1);
                sc = iImage(px(2)+1,px(1));
```

```

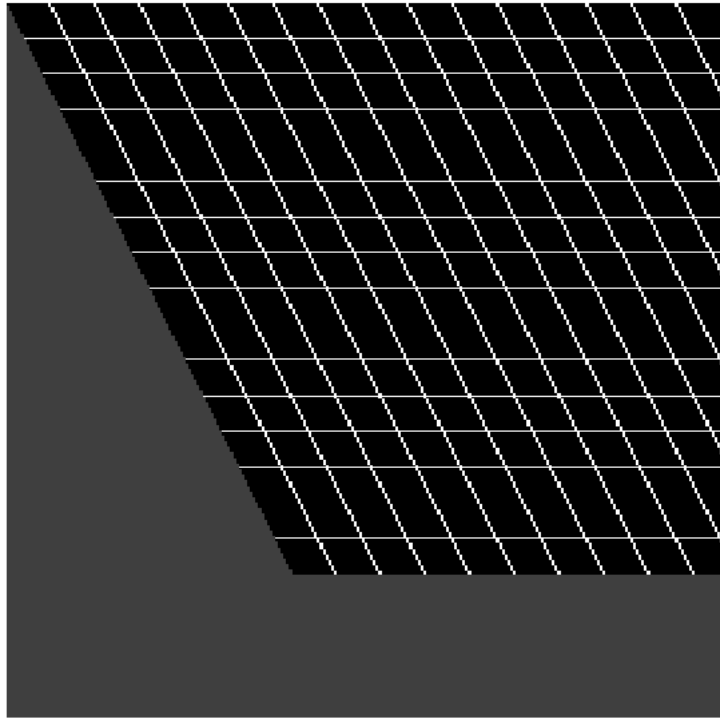
        sd = iImage(px(2)+1,px(1)+1);
        s = floor(p4 * sa + p3 * sb + p2 * sc +p1 * sd);
        oImage(y+1,x+1) = s;
    end
end

end
end
end

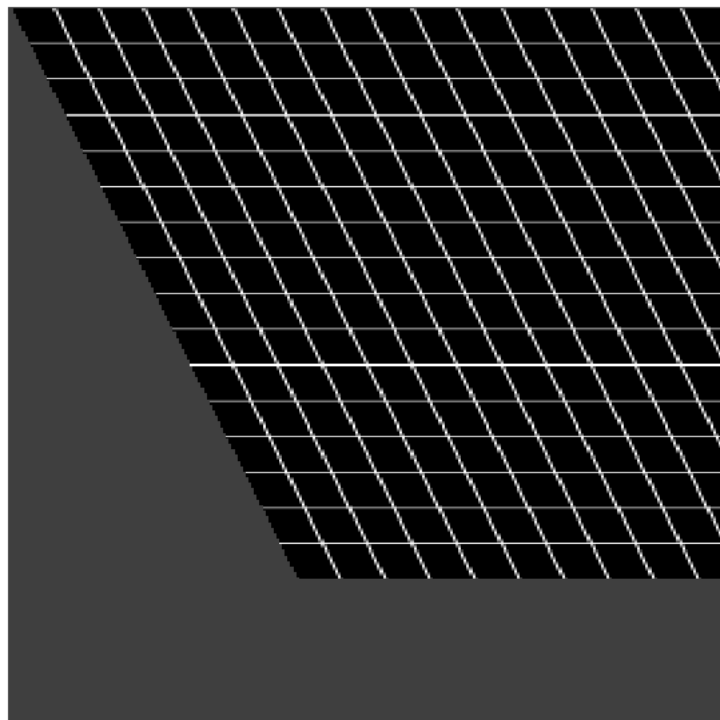
```



Slika 1: Originalna slika



Slika 2: Slika, po rotaciji in striženju interpolirana z ničtim redom



Slika 3: Slika, po rotaciji in striženju interpolirana z prvim redom

## 2 Afine preslikave



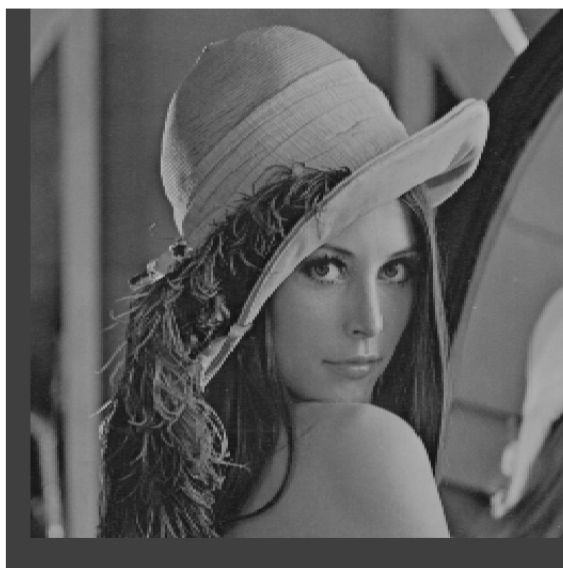
Slika 4: Originalna slika

### 2.1 Skaliranje s parametri $k_x = 0.7$ in $k_y = 1.4$



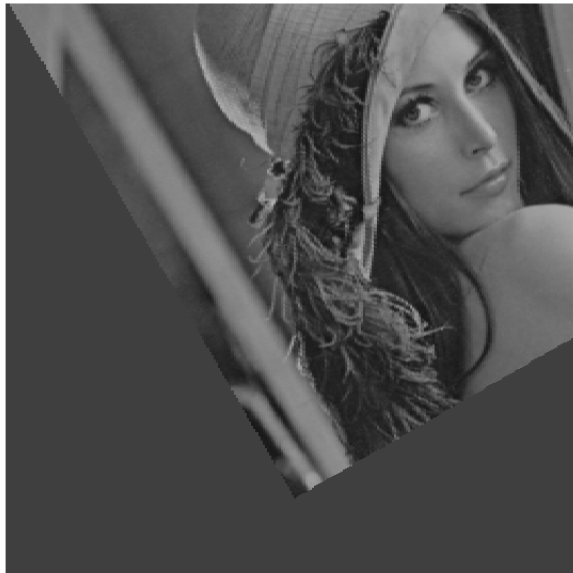
Slika 5: Skalirana slika

## 2.2 translacija s parametri $t_x = 20\text{mm}$ in $t_y = -30\text{mm}$



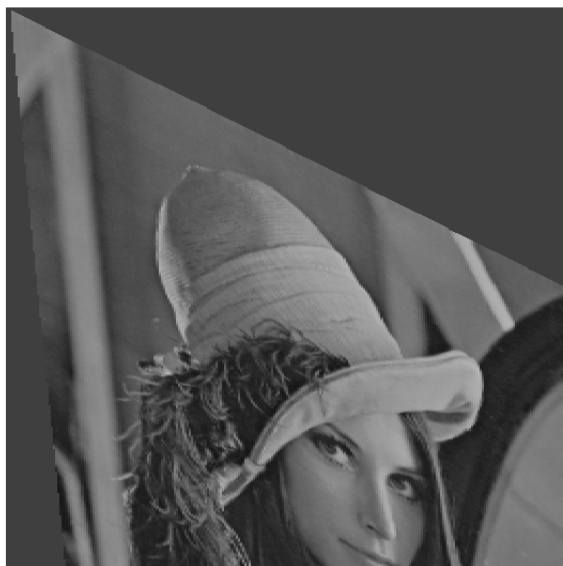
Slika 6: Translirana slika

## 2.3 Rotacija z $\varphi = -30^\circ$



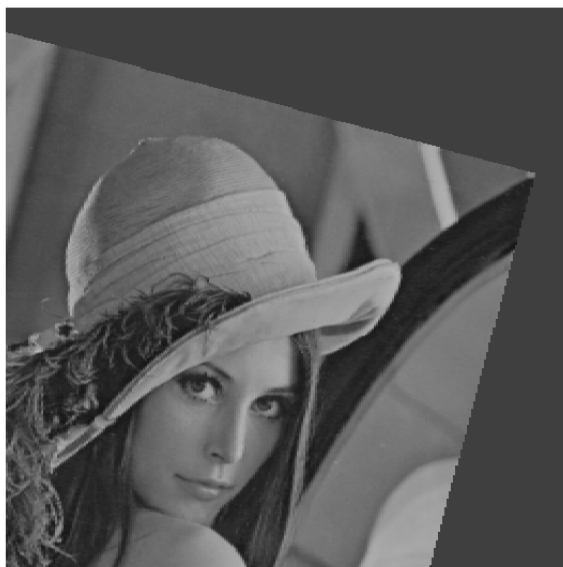
Slika 7: Rotirana slika

## 2.4 Strig s parametri $g_{xy} = 0.1$ in $g_{yx} = 0.5$



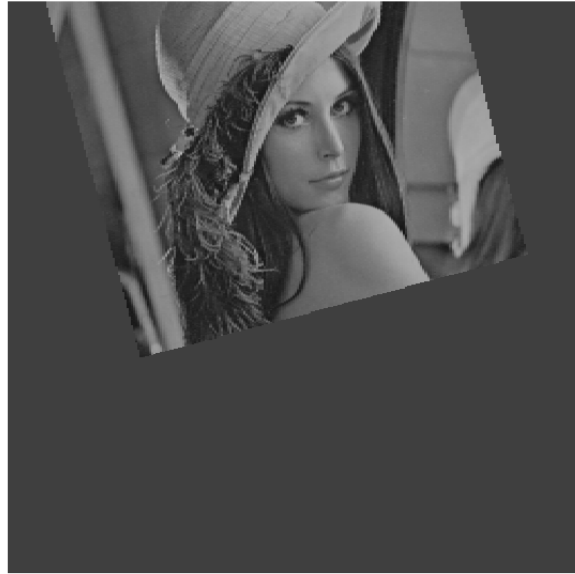
Slika 8: Slika s strigom

## 2.5 Translacija in rotacija $t_x = -10\text{mm}$ , $t_y = 20\text{mm}$ , $\varphi = 15^\circ$



Slika 9: Rotirana in translirana slika

## 2.6 Skaliranje, translacija in rotacija $k_x = k_y = 0.7, t_x = 30\text{mm}, t_y = -20\text{mm}, \varphi = -15^\circ$



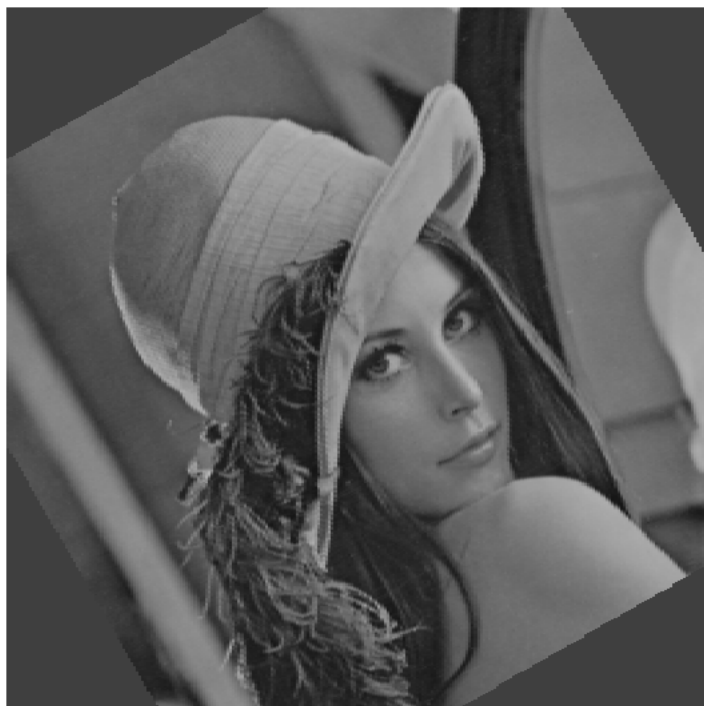
Slika 10: Rotirana in translirana in skalirana slika

## 3 Preslikavi

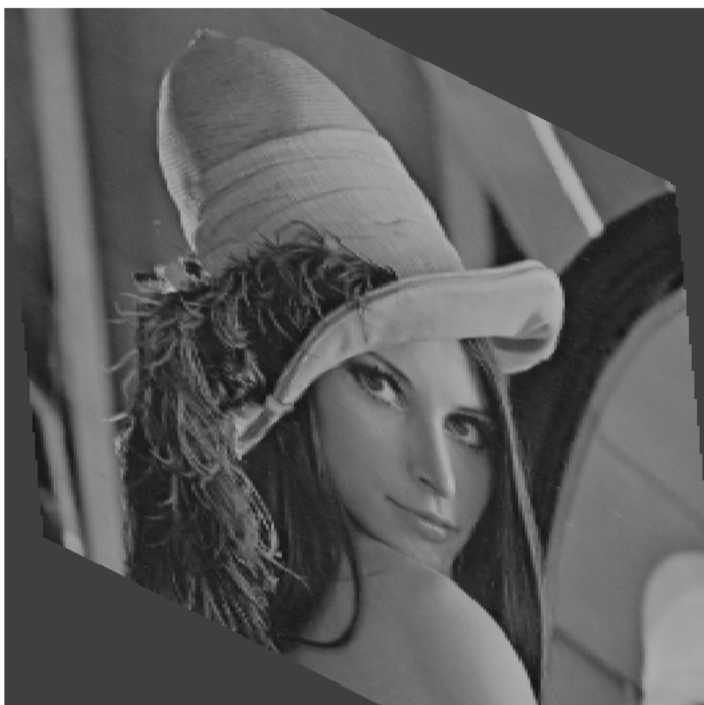
Preslikava iz vprašanja 2e se imenuje Togi premik. Zanj je značilno, da ohranja razdalja. Torej razdalja originalnih točk je enaka razdalji preslikanih točk.  $d(A, B) = d(A', B')$  Togi premik je injektivna preslikava (kar pomeni, da se dve različni točki ne moreta preslikati v isto.) Preslikava iz vprašanja 2f pa je podobnostna transformacija. Za podobnostno transformacijo je značilno, da je izogonalna, kar pomeni, da ohranja pravokotnost koordinatnih osi. S tem se tudi ohranja velikost kotov in vzporednost. Dolžine in položaji točk v tej preslikavi pa se ne ohranjajo. Preslikava je podobnostana, če velja  $k_x = k_y$



## 4 Preslikave z izhodiščem v sredini slike



Slika 11: Rotacija v središču slike



Slika 12: Strig v središču slike

```

phi = -30;
rotacijapremik = getParameters('affine',[1,1] ,
                                [-imsize(1)/2 *pxDim(1), -imsize(2)/2 *pxDim(2)],
                                phi, [0 0]);
premik = getParameters('affine',[1,1],
                        [imsize(1)/2*pxDim(1), imsize(2)/2*pxDim(2)],0,[0,0]);
RT= premik*rotacijapremik;
transf = transformImage('affine',slika , pxDim, inv(RT),63,1);
slika_rotirana = displayImage(transf, 'rotirana_sredina',gX, gY);
saveas(slika_rotirana, 'tretja_a.png')

gxyd = 0.1;
gyxd = 0.5;
strig_premik = getParameters('affine',[1 1],
                              [-imsize(1)/2 *pxDim(1), -imsize(2)/2 *pxDim(2)],
                              0, [gxyd, gyxd]);
RT2 = premik*strig_premik;
slika_strizenje = transformImage('affine',slika , pxDim, inv(RT2) ,63,1);
strig_slika = displayImage(slika_strizenje, 'Striženje slika',gX, gY);
saveas(strig_slika, 'tretja_b.png')

```

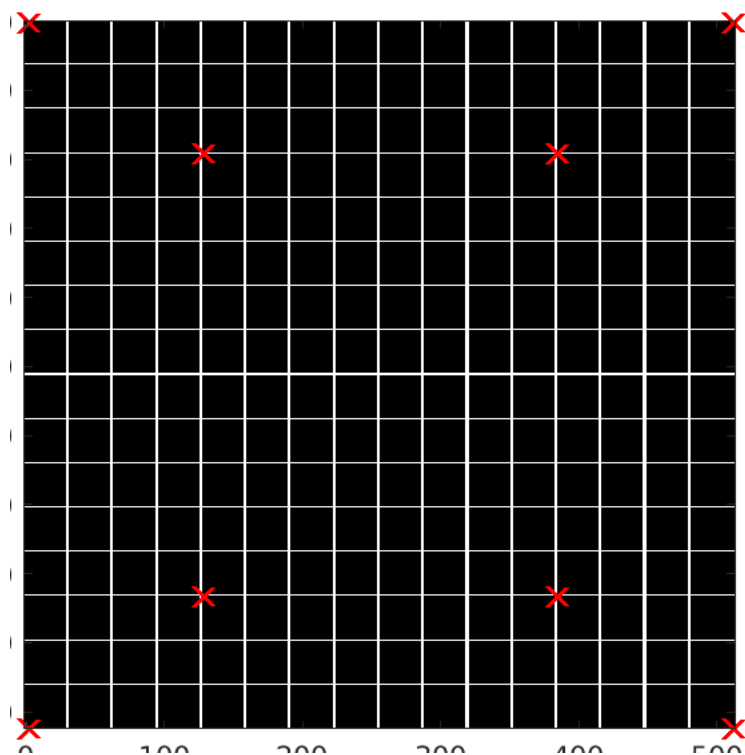
## 5 Radialna preslikava



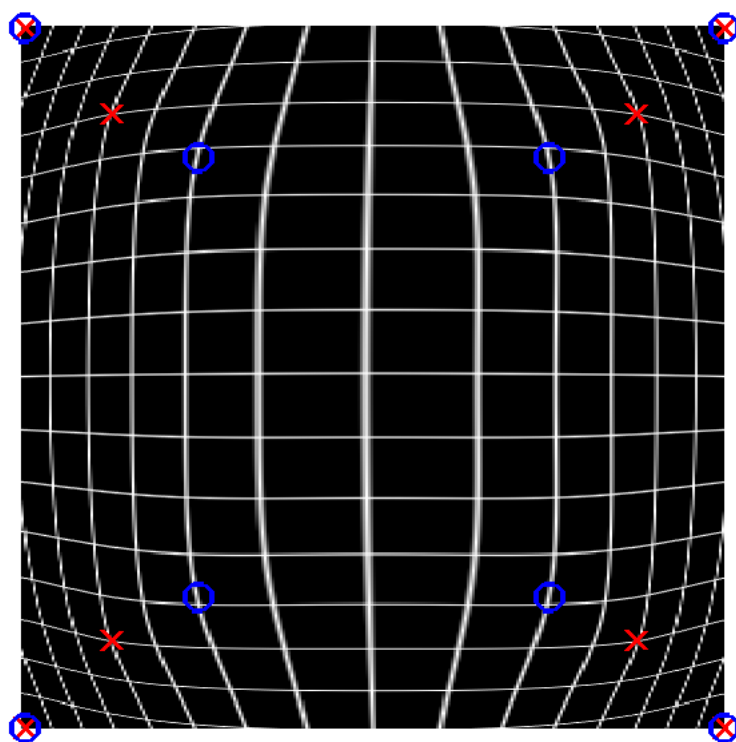
Slika 13: Originalna slika s kontrolnimi točkami



Slika 14: Preslikana slika s preslikanimi kontrolnimi točkami



Slika 15: Originalna slika s kontrolnimi točkami



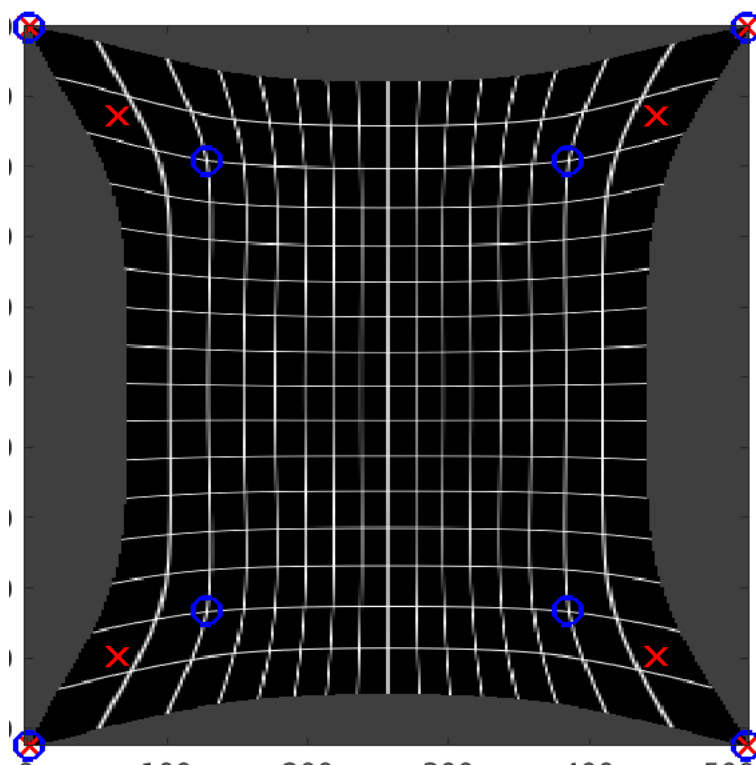
Slika 16: Preslikana slika s preslikanimi kontrolnimi točkami

Vidimo, da preslikava ne deluje pravilno. Kotne točke, se ne spremenijo, ker so enake (torej se preslikajo same vase). Če opazujemo notranje točke, vidimo, da se preslikane kontrolne točke preslikajo v originalne kontrolne točke. Podobno kot pri afini preslikavi moramo tudi tukaj vzeti

inverz radialne preslikave oz. slikati preslikane kontrolne točke v kontrolne točke, da dobimo prav rezultat. To se zgodi ker preslikava ne slika iz prostora indeksov v prostor indeksov.



Slika 17: Pravilno preslikana slika



Slika 18: Pravilno preslikana slika

Kot vidimo preslikava sedaj deluje pravilno. Kontrolne točke res preslika v preslikane kontrolne točke.

## Literatura

- [1] **Togi premik**, Wikipedia, dostopno na: [https://sl.wikipedia.org/wiki/Togi\\_premik](https://sl.wikipedia.org/wiki/Togi_premik)  
[ogled 16.11.2020]
- [2] **Podobnostna transformacija**, M. Kuhar, FGG, dostopno na: [http://fgg-web.fgg.uni-lj.si/~mkuhar/Pouk/RSG/gradivo/Podobnostna\\_transformacija-gradivo.pdf](http://fgg-web.fgg.uni-lj.si/~mkuhar/Pouk/RSG/gradivo/Podobnostna_transformacija-gradivo.pdf)  
[ogled 16.11.2020]
- [3] **Geometrija in transformacije**, FRI, dostopno na: [https://ucilnica.fri.uni-lj.si/pluginfile.php/159572/mod\\_resource/content/0/12%20Geometrija%20in%20transformacije.pdf](https://ucilnica.fri.uni-lj.si/pluginfile.php/159572/mod_resource/content/0/12%20Geometrija%20in%20transformacije.pdf)  
[ogled 16.11.2020]