

# Poročilo 4. vaje pri predmetu OVS

## Prikazovanje 3D slik v 2D

Tina Zwitter 64200432

3. november 2020

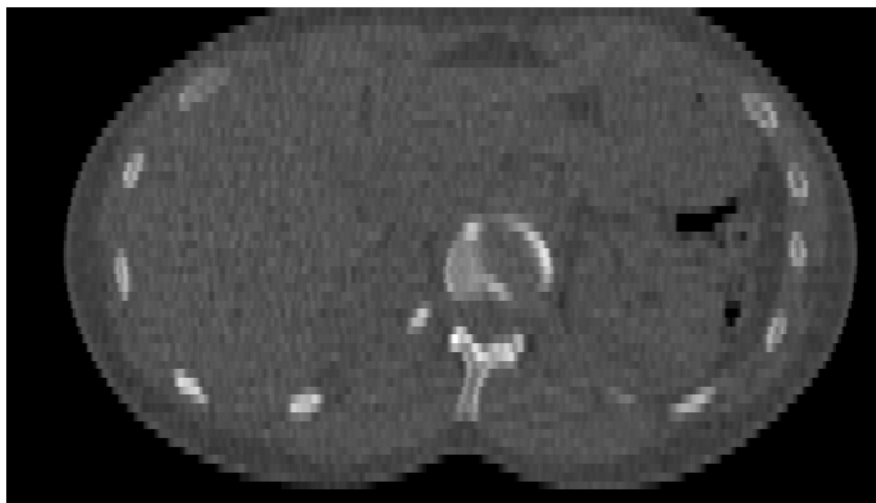
# 1 Pravokotne ravninske projekcije



Slika 1: Čelni prerez



Slika 2: Stranski prerez



Slika 3: Prečni prerez

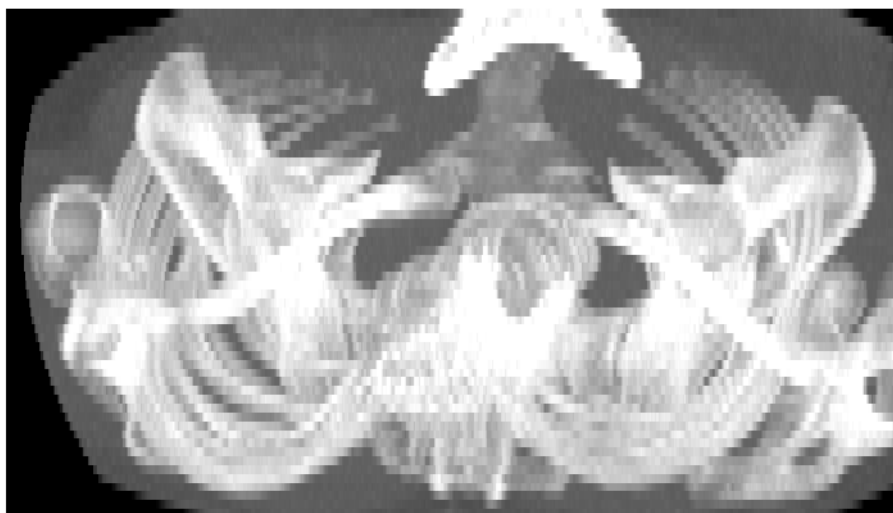
## 2 Pravokotne ravninse projekcije z maksimalno in povprečno vrednostjo



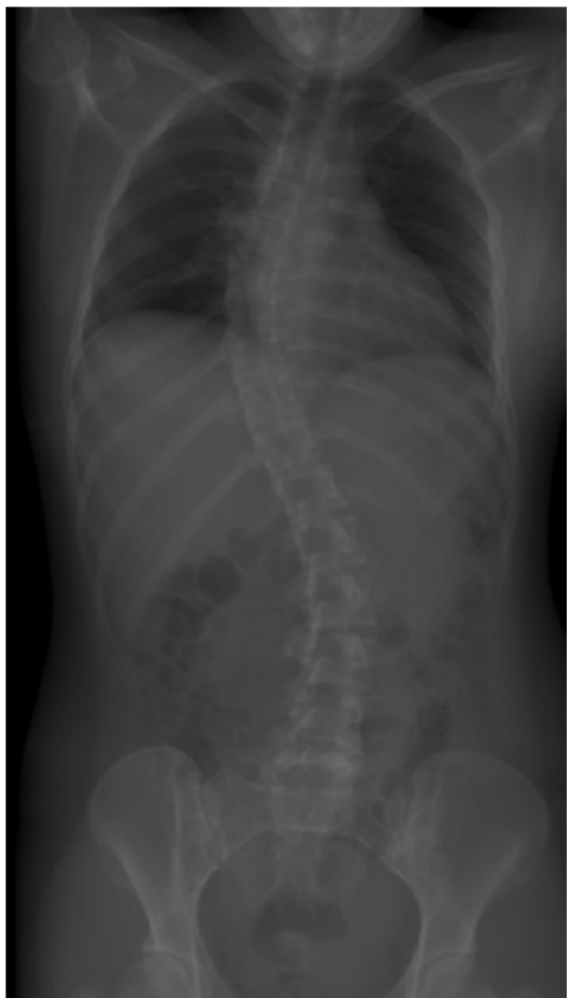
Slika 4: Čelna projekcija maksimalnih vrednosti prerezov



Slika 5: Stranska projekcija maksimalnih vrednosti prerezov



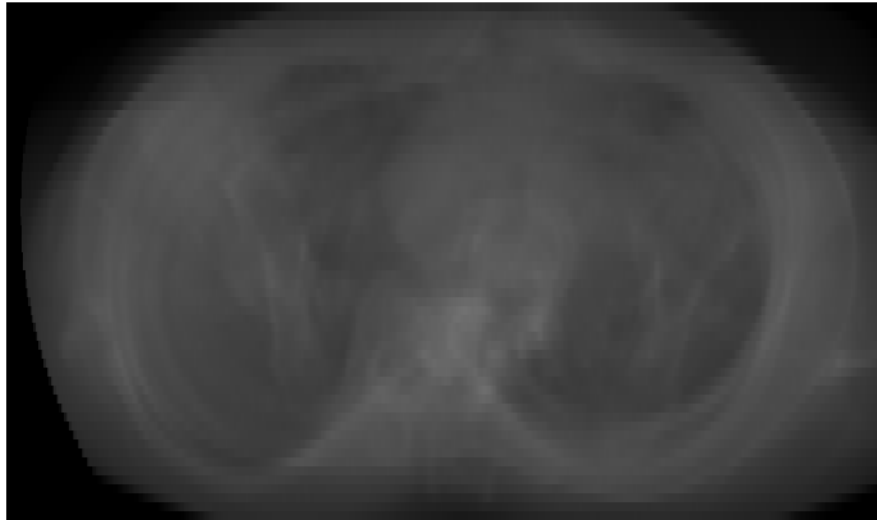
Slika 6: Prečna projekcija povprečnih vrednosti prerezov



Slika 7: Čelna projekcija povprečnih vrednosti prerezov



Slika 8: Stranska projekcija povprečnih vrednosti prerezov



Slika 9: Prečna projekcija povprečnih vrednosti

### 3 Smiselnost projekcij

*Katere vrste projekcij, pri katerih za funkcijo točk uporabite maksimalno vrednost, minimalno vrednost, povprečno vrednost, vrednost mediane, vrednost standardnega odklona oz. vrednost variance, je v primeru prikazovanja CT slik človeškega telesa sploh smiselno računati? Obrazložite odgovor.*

Najbolj smiselne so pravokotne čelna, stranska in prečna projekcija. Za funkcijo točk lahko vzamemo maksimalno vrednost, če nas zanimajo npr. Kosti. Pri minimalni vrednosti lahko vidimo, kje je zrak in s tem vidimo, ali je kaj v pljučih, kar ne bi smelo biti. S standardnem odklonom lahko izračunamo kakšen je šum slike zaradi zunanjih pojavov. Pri mediani lahko opazimo ali je neko področje bolj gosto kosti ali manj. Uporabljajo se tudi poševni ravninski prerezi, kjer lahko vidimo neko poškodbo, ki jo pri čelnem pogledu prekriva kakšna kost. Uporabne so pa tudi ne ravninske projekcije, naprimer projekcija na ploskev, ki oklepa hrbtenico. Tako bi lahko hrbtenico brez popačenj prenesli na ravnino. Pri ne-ravninskih projekcijah je treba vedeti, kaj delamo, kakšno ploskev vzamemo. Dobimo lahko velika popačenja, ker se nam lahko projekcijski žarki sekajo med sabo.

### 4 Funkcija `loadImage3D`

*Spremenite funkcijo `loadImage3D()` tako, da 3D sliko najprej naložite kot vektor, nato pa ta vektor preoblikujete v 3D matriko s pomočjo Matlabove funkcije `reshape()`. Priložite programsko kodo tako spremenjene funkcije.*

```

function oImage = loadImage3D(iPath, iSize, iType)
% funkcija zapiše raw sliko v obliki matirke
% vhodni podatki :
%   iPath - vhodna datoteka
%   iSize - velikost slike
%   iType - tip podatkov, v katerem je zapisana datoteka
% izhodni podatek:
%   oImage - slika zapisana v matriki

% naloži 3D sliko
fid = fopen(iPath, 'r');
all = fread(fid, prod(iSize), iType);
oImage = reshape(all, iSize(2), iSize(1), iSize(3));

fclose(fid);
end

```

## 5 Rotirana ravnina

S to nalogo sem imela nekaj težav, kar se odraža tudi pri stilu in hitrosti kode.

```

elseif iNormVec(3) == 0
    xn = iNormVec(1);
    yn = iNormVec(2);
    %izračunamo kot glede na normo
    phi = atan(xn/yn);
    rotirana = zeros(N_cor, N_sag, N_ax);
    for x = 0:N_sag-1
        for y = 0:N_cor-1
            %rotacija in translacija indeksa
            x2 = round((x-N_sag/2) * cos(phi) - (y-N_cor/2) * sin(phi)+ N_sag/2);
            y2 = round((x-N_sag/2) * sin(phi) + (y -N_cor/2) * cos(phi) + N_cor/2);
            for z = 0:N_ax-1
                %pogledamo, ali pademo "ven"
                if x2<0 | y2<0 | x2>=N_sag | y2>=N_cor
                    rotirana(y+1,x+1,z+1) = 0;
                else
                    %pogledamo kaj je na tem indeksu
                    rotirana(y+1,x+1,z+1) = iImage(y2+1,x2+1,z+1);
                end
            end
        end
    end

    end

    end

% na preseke dodamo funkcijo
oP = zeros(N_ax,N_sag);

```

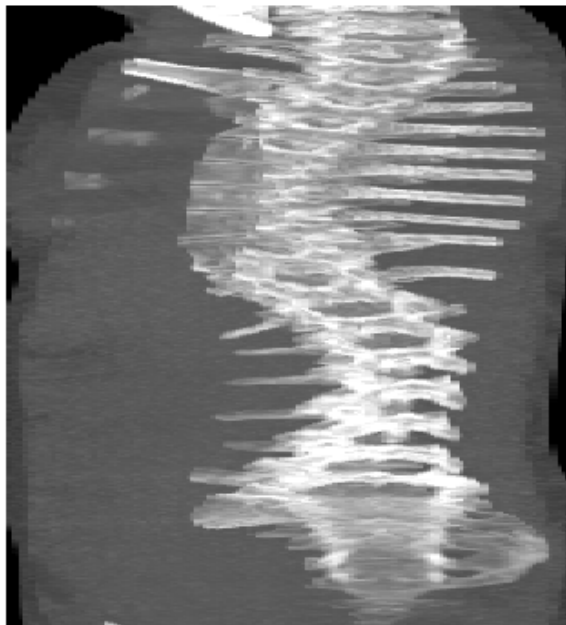


```

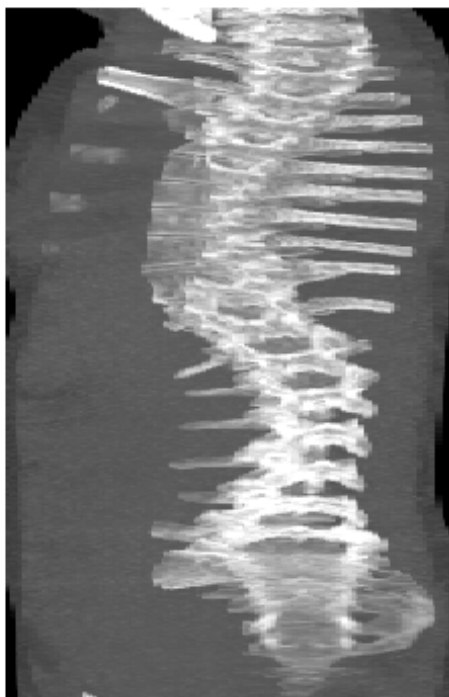
for z=1:N_ax
    for x = 1:N_sag
        oP(z,x) = feval(iFunc, rotirana(:,x,z));
    end
end

% odrežemo črnino na levi in desni
for xod = 1: N_sag
    if max(oP(:,xod))
        k = xod;
        break
    end
end
for xod = N_sag:-1:1
    if max(oP(:,xod))
        k2 = xod;
        break
    end
end
oH = (0:(k2-k)-1).*(iDim(1).*cos(phi)+iDim(2)*sin(phi));
oV = (0:N_ax-1).*iDim(3);
oP = oP(:,k:k2);
end

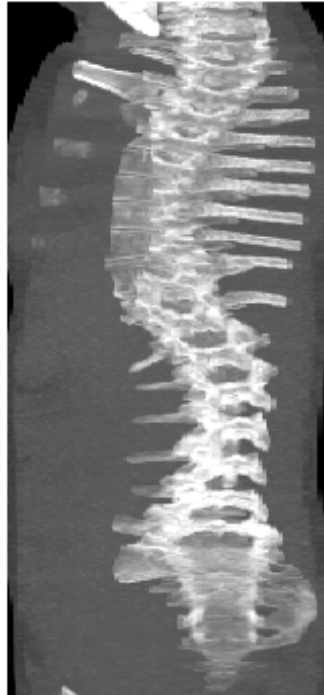
```



Slika 10: Primer maksimalne vrednosti prerezov pri  $\vec{n}_1 = (3, 83, 9, 24, 0)$



Slika 11: Primer maksimalne vrednosti prerezov pri  $\vec{n}_2 = (1, 1, 0)$



Slika 12: Primer maksimalne vrednosti prerezov pri  $\vec{n}_3 = (9, 24, 3, 83, 0)$

Slabosti uporabljenega pristopa je, da izgubimo podatke o sliki, ker nobena točka ne pade na njih (zaradi rotacije). Tako lahko spregledamo kakšno poškodbo. Pri pristopu uporabimo interpolacijo, tako da ni nujno da zajamemo vse točke (ker morda kakšna točka ni najbližji sosed.) Pri Stranski projekciji lahko pride do tega, da je v sliki le del organa, nato del organa pade iz slike in nam taka slika ne koristi. Moj algoritem je tudi zelo časovno potraten, verjemam da se ga da zelo optimizirati.

## Literatura

- [1] **Maximum intensity projection**, Wikipedia, dostopno na: [https://en.wikipedia.org/wiki/Maximum\\_intensity\\_projection](https://en.wikipedia.org/wiki/Maximum_intensity_projection) [ogled 3.11.2020]
- [2] **CT scan**, Wikipedia, dostopno na: [https://en.wikipedia.org/wiki/CT\\_scan](https://en.wikipedia.org/wiki/CT_scan) [ogled 3.11.2020]
- [3] **Računalniška tomografija (CT) VI**, IJS, dostopno na: <http://www-f9.ijs.si/~krizan/sola/medfiz/slides/fiz-anat-slik/7FAS-CT.pdf> [ogled 3.11.2020]