

Vaja 7: Prikazovanje medicinskih slik

Pripravila: Žiga Bizjak & Tomaž Vrtovec

Navodila

Objekti, ki jih opazujemo v dvodimenzionalnih (2D) slikah, so v splošnem tridimenzionalni (3D) in jih lahko zato boljše analiziramo na podlagi 3D slik, ki jih predstavimo kot funkcije $f(x; y; z)$ slikovnih vrednosti (intenzitet) v odvisnosti od prostorskih koordinat x , y in z . Prikazovanje 3D slik v 2D omogočajo načini prikazovanja v obliki prereзов in projekcij.

1. Datoteka 'Vaja07/lung_001.nii.gz' vsebuje 3D 8-bitno sivinsko sliko, zajeto s tehniko računalniške tomografije (CT). Naložite sliko s pomočjo knjižnice nibabel.
2. Na osnovi 3D slike $f(x; y; z)$ lahko določimo pravokotne ravninske prereze, in sicer:

- stranski (lateralni, sagitalni): $c_{x[x=x_c]}(y, z) = f(x_c, y, z)$,
- čelni (frontalni, koronalni): $c_{y[y=y_c]}(x, z) = f(x, y_c, z)$,
- prečni (transverzalni, aksialni): $c_{z[z=z_c]}(x, y) = f(x, y, z_c)$,

kjer so x_c , y_c in z_c izbrani položaji posameznih prereзов. Napišite funkcijo za določanje poljubnega pravokotnega ravninskega prereза:

```
def getPlanarCrossSection(iImage, iNormVec, iLoc):  
    # ...  
    # your code goes here  
    # ...  
    return oCS
```

kjer vhodni argument `iImage` predstavlja 3D sliko, `iNormVec` normalni vektor ravnine prereза ($\vec{n}_x = (1, 0, 0)$ za stransko, $\vec{n}_y = (0, 1, 0)$ za čelno ali $\vec{n}_z = (0, 0, 1)$ za prečno ravnino), `iLoc` pa položaj prereза v 3D sliki (x_c , y_c oz. z_c , v slikovnih elementih). Izhodni argument `oCS` predstavlja želeni pravokotni ravninski prereз.

Določite stranski pravokotni ravninski prereз dane 3D slike pri $x_c = 256$.

3. Napišite funkcijo za prikazovanje 2D slik:

```
def displayImage(iImage, iName),  
    # ...  
    # your code goes here  
    # ...
```

kjer vhodni argument `iImage` predstavlja 2D sliko, ki jo želite prikazati, `iName` pa ime (naslov) slike, ki ga uporabite za razlikovanje med več prikazanimi slikami. Prikažite stranski pravokotni ravninski prereз dane 3D slike, ki ste ga pridobili pod točko 2.

4. Na osnovi 3D slike $f(x, y, z)$ lahko določimo pravokotne ravninske projekcije, in sicer:

- stranska (lateralna, sagitalna): $p_x(y, z) = PF_k(f(k, y, z)); \quad k = 1, 2, \dots, X$,
- čelna (frontalna, koronalna): $p_y(x, z) = PF_k(f(x, k, z)); \quad k = 1, 2, \dots, Y$,
- prečna (tranzverzalna, aksialna): $p_z(x, y) = PF_k(f(x, y, k)); \quad k = 1, 2, \dots, Z$,

kjer je PF poljubna funkcija točk. Med najbolj razširjenimi vrstami projekcij je t.i. projekcija maksimalnih vrednosti oz. MIP (ang. maximal intensity projection), ki za funkcijo točk uporablja maksimalno vrednost.

Napišite funkcijo za določanje poljubne pravokotne ravninske projekcije:

```
def getPlanarProjection(iImage, iNormVec, iFunc),
    # ...
    # your code goes here
    # ...
    return oP
```

kjer vhodni argument $iImage$ predstavlja 3D sliko, $iNormVec$ normalni vektor ravnine projekcije (\vec{n}_x , \vec{n}_y ali \vec{n}_z), $iFunc$ pa funkcijo točk PF v obliki Matlabove funkcije (npr. 'max' za MIP). Izhodni argument oP predstavlja želeno pravokotno ravninsko projekcijo.

Določite in prikažite stransko pravokotno ravninsko projekcijo vrste MIP.

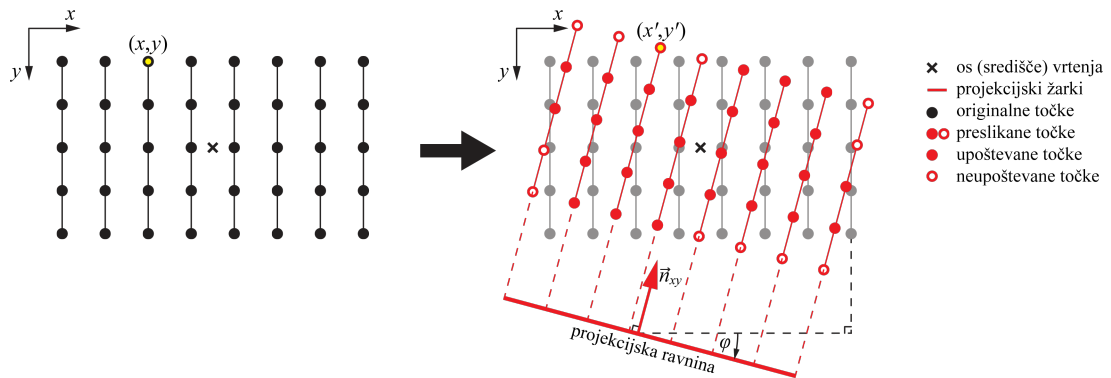
Vprašanja

Odgovore na sledeča vprašanja zapišite v poročilo, v katerega vstavite zahtevane izrise in programske kode.

1. Priložite slike naslednjih pravokotnih ravninskih prerezov dane 3D slike:
 - stranski prerez na položaju $x_c = 60$ slikovnih elementov,
 - čelni prerez na položaju $y_c = 35$ slikovnih elementov,
 - prečni prerez na položaju $z_c = 90$ slikovnih elementov.
2. Priložite slike stranske, čelne in prečne pravokotne ravninske projekcije dane 3D slike, pri čemer za funkcijo točk PF uporabite:
 - maksimalno vrednost,
 - povprečno vrednost.
3. 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.
4. Določite poljubno čelno poševno ravninsko projekcijo na podlagi normalnega vektorja na ravnino projekcije v obliki $\vec{n}_{xy} = (n_x, n_y, 0)$, ki določa kot rotacije φ ravnine projekcije. Uporabite najbolj preprost pristop k določevanju poševne projekcije, in sicer da koordinate točk oz. slikovnih elementov (x, y) v čelnem pravokotnem pogledu povežete v daljice ter izračunate koordinate zavrtenih točk (x', y') :

$$\begin{aligned}x' &= x \cos \varphi - y \sin \varphi, \\y' &= x \sin \varphi + y \cos \varphi,\end{aligned}$$

pri čemer os (središče) vrtenja postavite v središče prečne ravnine. Projekcije nato računate vzdolž posameznih zavrtenih daljic (koordinate zaokrožite na najbližji slikovni element), pri čemer ne upoštevate točk, ki so izven slike. Glede na to, da gre za čelno projekcijo, postopek ponovite za vsak prerez vzdolž koordinatne osi z .



Algoritem implementirate v obstoječi funkciji `getPlanarProjection()` tako, da dodate naslednji pogojni stavek:

```
>> elseif iNormVec(3) == 0
>>     << vaš algoritem >>
>> end
```

Priložite programsko kodo algoritma ter slike čelnih poševnih ravninskih projekcij vrste MIP za normalne vektorje $\vec{n}_1 = (3,83, 9,24, 0)$, $\vec{n}_2 = (1, 1, 0)$ in $\vec{n}_3 = (9,24, 3,83, 0)$. Naštejte in obrazložite nekaj pomanjkljivosti uporabljenega pristopa k določanju poševnih ravninskih projekcij.

