

Obdelava biomedicinskih signalov in slik

Detekcija obrisov človeških organov z uporabo detektorja robov Canny

Jakob Maležič

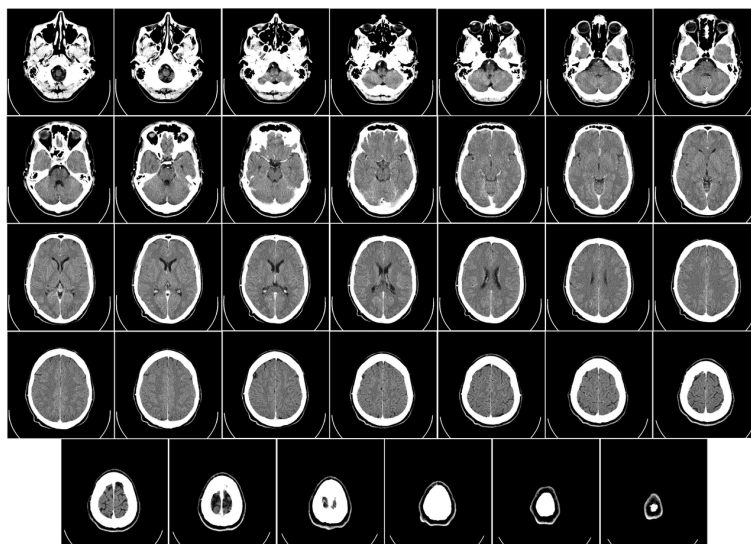
18. januar 2022

Povzetek

Pri vajah smo spoznali različne metode detekiranja robov. V tej nalogi je predstavljen detektor robov Canny za detekcijo obrisov človeških organov na slikah računalniške tomografije (CT). Slike smo pridobili iz baze CTMRI-DB¹. Algoritem smo razvili v programskem okolju Matlab.

1 Uvod

Slike CT se največ uporabljajo preventivno ali za iskanje bolezni pri ljudeh z višjo verjetnostjo pojavitve raka debelega črevesja [2], bolezni srca in ožilja [1] ter tudi bolezni dihal [3]. Na slikah CT 1 lahko vidimo celoten prerez glave z dodanim intravenskim kontrastnim medijem.



Slika 1: Primer slike CT (vir: https://en.wikipedia.org/wiki/CT_scan).

¹<https://lbcsi.fri.uni-lj.si/OBSS/Data/CTMRI/>

2 Metode

Algoritem, ki ga bomo opisali, je implementacija detektorja robov Canny. Sestavljen je iz šestih delov: Gaussovo glajenje, horizontalni in vertikalni gradient s Prewittovo masko, izračun magnitude in kota, dušitev ne-maksimalnih vrednosti, določanje močnih in šibkih pikslov s histerezo ter povezovanje robov.

2.1 Gaussovo glajenje

Gaussovo glajenje je rezultat zabrisanja slike z Gaussovo funkcijo. V Matlabu je to zelo preprosto narediti s konvolucijo nad piksli slike, kateri na vhod podamo Gaussovo matriko. Uporabimo ga za zmanjšanje šuma v slikah in zmanjšanje števila podrobnosti, ki nas ne zanimajo. Kako Gaussovo glajenje spremeni sliko lahko vidimo na sliki 2.



Slika 2: Primerjava originalne slike in slike obdelane z Gaussovim glajenjem.

2.2 Horizontalni in vertikalni gradient s Prewittovo masko

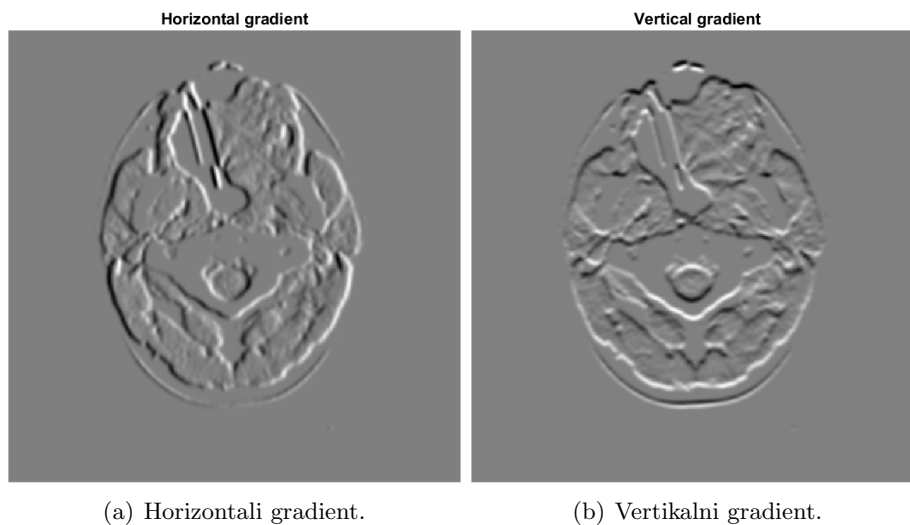
Za izračun magnitude in kota moramo najprej izračunati horizontalni ter vertikalni gradient. To prav tako naredimo s konvolucijo nad piksli slike in Prewittovo masko 1. Če gradient prikažemo na sliki, izgleda kot da smo dodali vir svetlobe in senčenje. Slednje lahko vidimo na sliki 3.

$$k_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} k_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (1)$$

2.3 Izračun magnitude in kota

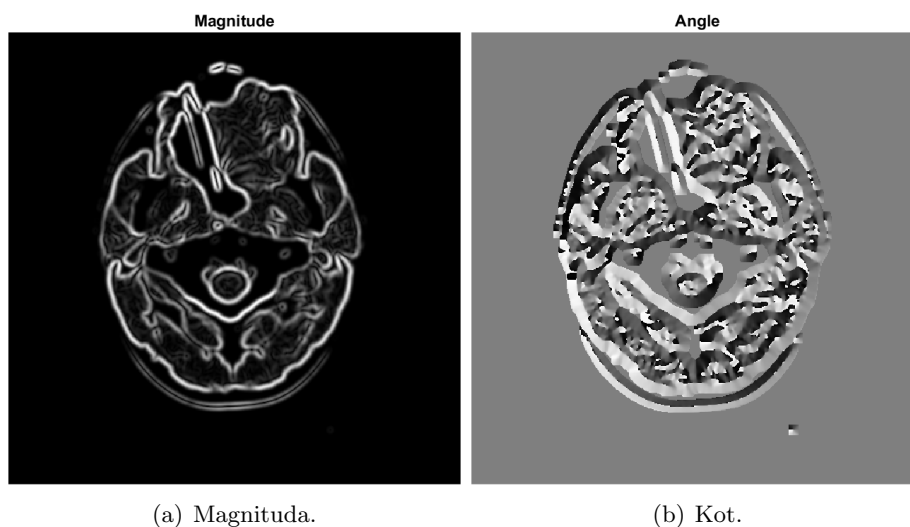
Iz pridobljenih gradientov lahko sedaj zelo preprosto izračunamo magnitudo in kot posameznih pikslov:

$$\begin{aligned} mag(x, y) &= \sqrt{(gradient_x^2 + gradient_y^2)} \\ angle(x, y) &= atan2(gradient_x, gradient_y) \end{aligned} \quad (2)$$



Slika 3: Primerjava gradientov.

Vizualizacijo magnitude in kota lahko vidimo na sliki 4. Vrednosti uporabimo za dušenje ne-maksimalnih vrednosti.



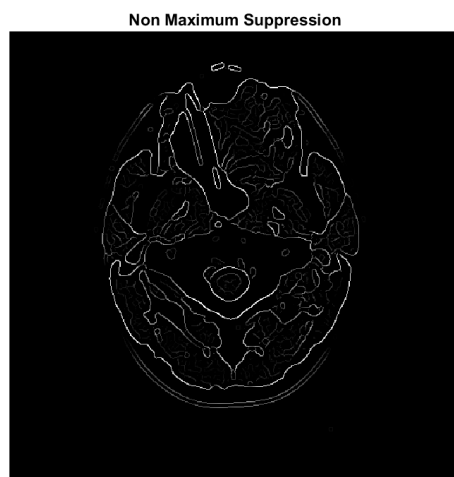
Slika 4: Magnituda in kot.

2.4 Dušenje ne-maksimalnih vrednosti

Iz slike 4 - magnituda lahko vidimo, da so robovi precej debeli. Robove zato z dušenjem ne-maksimalnih vrednosti stanjšamo. To naredimo v dveh korakih:

1. Iz štirih možnih smeri: horizontalna, vertikalna, 45° in -45° najdemo tisto, ki je najbližje kotu piksla.
2. Preverimo, če je magnituda trenutnega piksla manjša od magnitude vsaj enega od sosednjih pikslov v izračunani smeri, nastavimo magnitudo trenutnega piksla na 0.

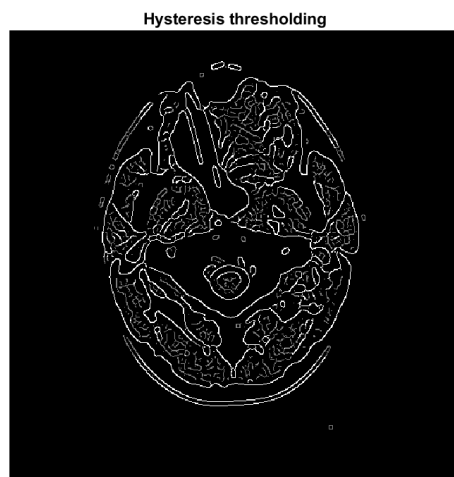
Rezultat dušenja lahko vidimo na sliki 5.



Slika 5: Dušenje ne-maksimalnih vrednosti.

2.5 Določanje močnih in šibkih pikslov s histerezo

Iz rezultata dušenja lahko vidimo različno intenzivnost posameznih pikslov. Ker želimo prikazati zgolj robove, določimo prag intenzitete za močne piksele, za katere smo prepričani da so del roba in prag za šibke piksele, za katere nismo popolnoma prepričani ali so del roba. Vsem ostalim pikslom, ki imajo intenziteto manjšo od obeh pragov, nastavimo intenziteto na 0. Tako dobimo sliko 6, na kateri smo dodatno vsem šibkim pikslom nastavili intenziteto na 50%, da jih boljše vidimo.

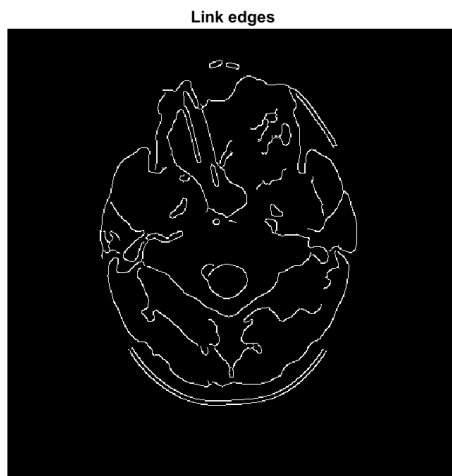


Slika 6: Pragovna histereza.

2.6 Povezovanje robov

V zadnjem koraku se moramo odločiti katere šibke piksele obdržati in katere zavreči. Slednje naredimo z rekurzivno funkcijo, ki za vsak močen piksel preveri njegove sosedo. Če je kateri od sosedov šibek piksel, ga popravi na močnega, tako da mu nastavi intenziteto na 100% in na popravljenemu

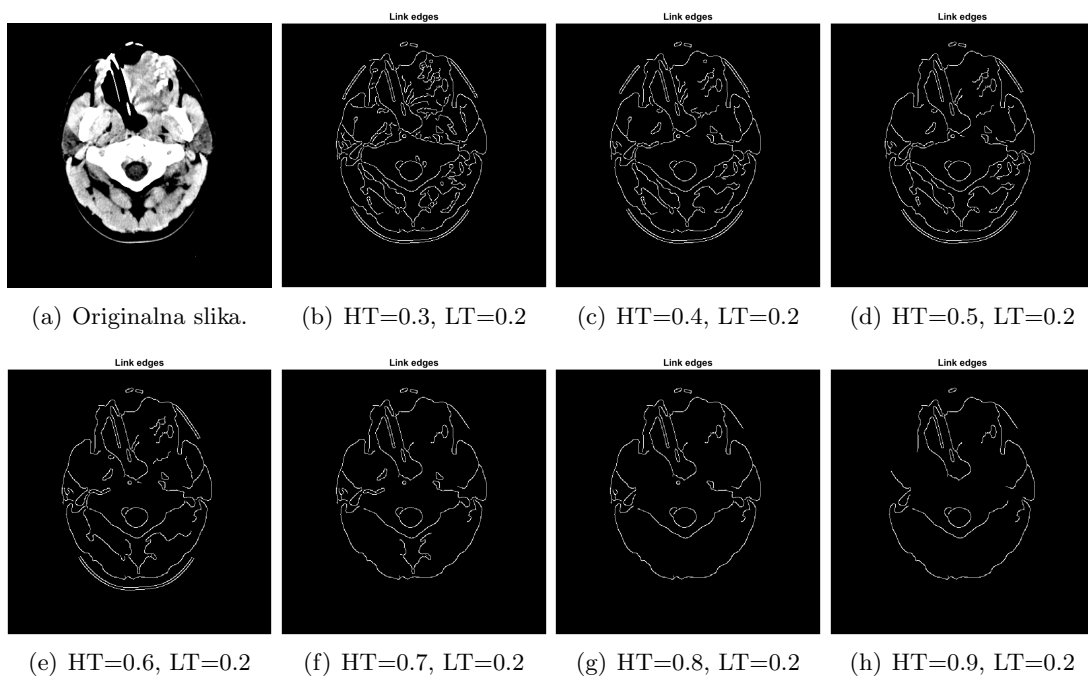
pikslu izvede rekurzivni klic. Ko se izvajanje rekurzije konča, intenziteto vseh ostalih šibkih pikslov nastavimo na 0. Tako nam ostanejo samo piksli, ki so povezani z močnimi piksli. To lahko vidimo na sliki 7.



Slika 7: Povezani robovi.

3 Rezultati

Na sliki 8 lahko vidimo kako prag za močne piksele vpliva na rezultat algoritma. Več primerov slik CT pred in po obdelavi, pa je na voljo v dodatku.



Slika 8: Primerjava različnih pragov.

4 Diskusija

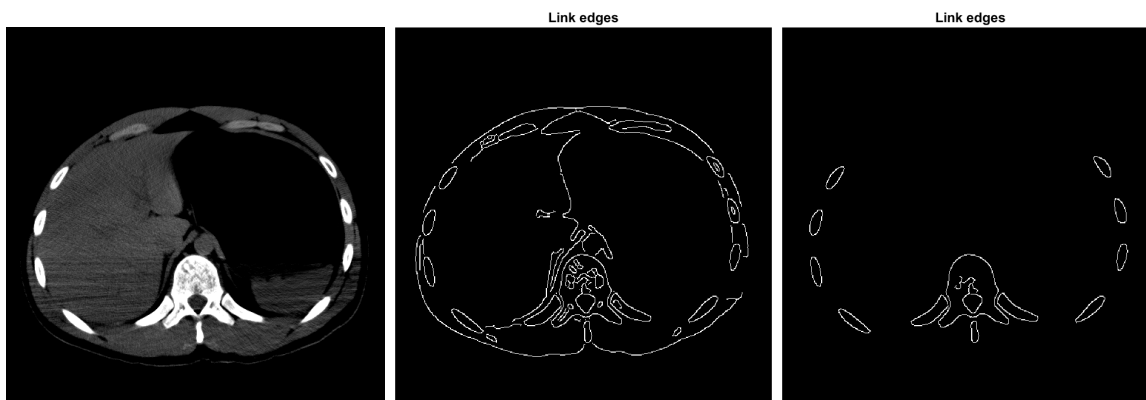
V predstavljenem delu smo opisali algoritem za zaznavanje robov Canny, s katerim lahko precej dobro zaznamo robove slik CT. Detektor zazna robove glavnih delov in jih ustrezno pobarva. V kolikor bi želeli obarvati še robove bolj podrobnih delov, bi morali temu ustrezno prilagoditi prag, ki smo ga določili pri določanju močnih in šibkih pikslov s histerezo.

Literatura

- [1] Jolly, M. Automatic Segmentation of the Left Ventricle in Cardiac MR and CT Images. *International Journal Of Computer Vision*. **70**, 151-163 (2006,11,1)
- [2] Pickhardt, P., Kim, D., Menias, C., Gopal, D., Arluk, G. & Heise, C. Evaluation of Submucosal Lesions of the Large Intestine. *RadioGraphics*. **27**, 1681-1692 (2007)
- [3] Hu, S., Hoffman, E. & Reinhardt, J. Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images. *IEEE Transactions On Medical Imaging*. **20**, 490-498 (2001)

Dodatek: GitHub

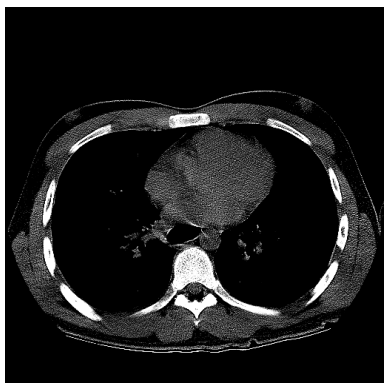
Implementacija je na voljo na: <https://github.com/Blarc/ct-images-edge-detection>.



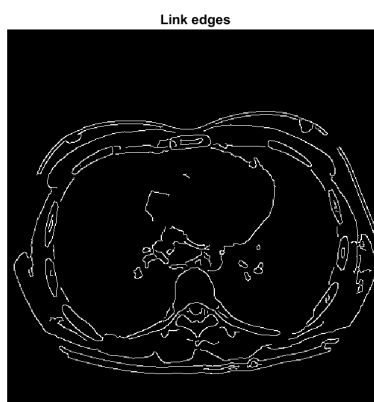
(a) Originalna slika.

(b) HT=0.2, LT=0.15.

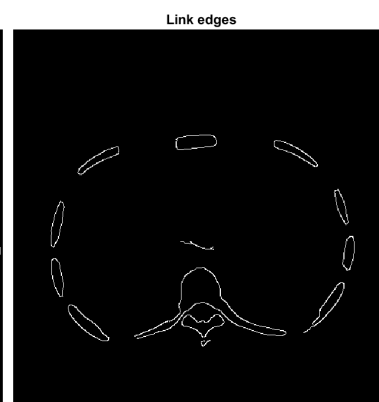
(c) HT=0.6, LT=0.2.



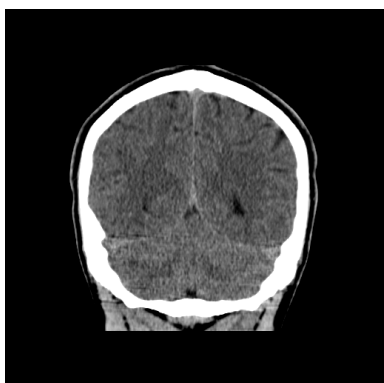
(d) Originalna slika.



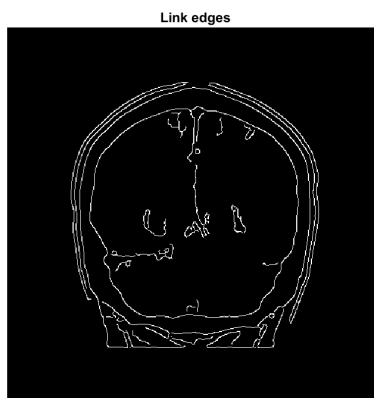
(e) $HT=0.2$, $LT=0.15$.



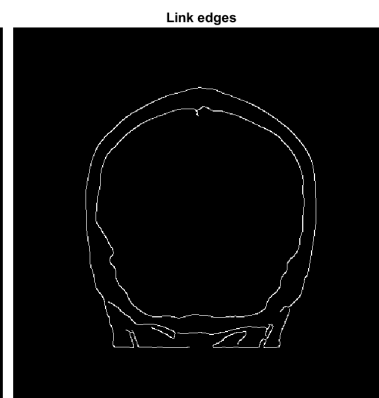
(f) $HT=0.6$, $LT=0.2$.



(g) Originalna slika.



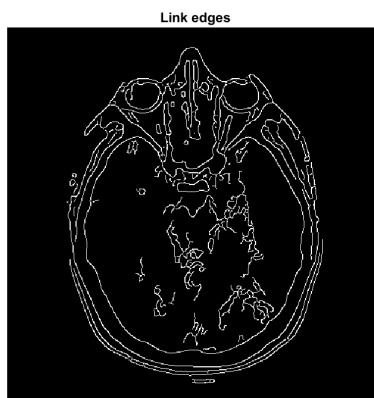
(h) $HT=0.2$, $LT=0.15$.



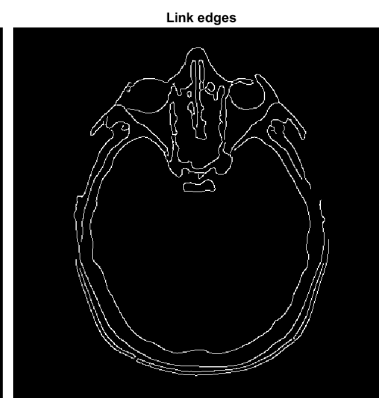
(i) $HT=0.6$, $LT=0.2$.



(j) Originalna slika.



(k) $HT=0.2$, $LT=0.15$.



(l) $HT=0.6$, $LT=0.2$.