

# Cannyjev detektor robov

## 2. naloga pri predmetu Obdelava biomedicinskih signalov in slik

Ivan Antešić<sup>1</sup>

Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, Večna pot 113,  
Slovenija

`ia6382@student.uni-lj.si`

**Keywords:** Zaznava robov · Canny · Procesiranje slik · CT.

### 1 Uvod

V okviru druge domače naloge smo v MATLAB okolju implementirali Cannyjev algoritem za zaznavo robov [1] in ga preizkusili na slikah računalniške tomografije (ang. computed tomography, CT). Testne slike smo pridobili iz baze na voljo na spletni strani laboratorija<sup>1</sup>. Implementirali in primerjali smo tudi nekatere izboljšave za povezovanje robov.

### 2 Implementacija

#### 2.1 Osnovni algoritem

Na kratko opišimo osnovni Cannyjev algoritem, ki sestoji iz petih korakov.

1. Glajenje originalne, vhodne slike z Gaussovim filtrom.
2. Izračun magnitud in kotov gradienta iz prvih odvodov v vertikalni ter horizontalni smeri za vsak piksel.
3. Tanjšanje robov. Za vsak piksel zaokrožimo gradient na eno od 4 smeri in v sosesčini 3x3 preverimo ali obstaja v določeni smeri piksel z večjo magnitudo. Če je pogoj izpolnjen zatremo magnitudo piksla.
4. Piksle razdelimo v dve skupini glede na dve mejni vrednosti  $TH$  in  $TL$ . Piksle z magnitudo večjo od  $TH$  označimo kot močne robne piksele, tiste z magnitudo med  $TH$  in  $TL$  pa kot šibke. Močni piksli predstavljajo binarno sliko robov.
5. Povežemo robove s preverjanjem sosednosti okoli močnih robnih pikslov - vse šibke piksele v okolici 3x3 spremenimo v močne piksele in tako pridobimo končni rezultat.

---

<sup>1</sup> laboratorij

## 2.2 Izboljšava povezovanja robov

Za zadnji korak algoritma smo želeli implementirati in preizkusiti več pristopov. Najenostavnejše je preprosto enkrat preveriti vse piksele in spremeniti šibke sosedne v močne če se v njihovi sosednosti nahaja močan piksel. Slabost takšnega pristopa je, da ne upoštevamo na novo pridobljenih močnih pikslov.

Zato smo implementirali prilagojen pristop imenovan Ena komponenta naenkrat (del algoritma [3]).

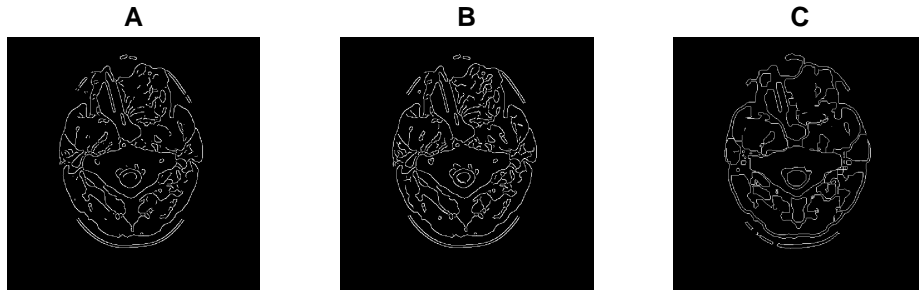
1. V sklad dodaj naslednji nepregledani močni piksel.
2. Odstrani piksel iz sklada in preglej vse njegove sosedne. Če je sosed šibek ga spremeni v močnega, označi kot pregledanega in ga dodaj v sklad.
3. Dokler sklad ni prazen ponavljaj korak 2, nato nadaljuj s korakom 1 dokler niso pregledani vsi močni piksli.

Poleg tega smo implementirali še pristop povezovanja robov imenovan Lokalno procesiranje (opisan v knjigi [2]).

1. Glajenje originalne, vhodne slike z Gaussovimi filtrom.
2. Izračun magnitud in kotov gradienta iz prvih odvodov v vertikalni ter horizontalni smeri za vsak piksel.
3. Izračun binarne slike, kjer je vak piksel 1 če velja da je njegova inteziteta večja od mejne vrednosti  $TM$  in kot gradienta v intervalu  $[A - AT, A + AT]$ , kjer sta  $A$  in  $AT$  vnaprej določeni vrednosti.
4. Zapolnimo vrzeli manjše kot  $K$  binarne slike v horizontalni in vertikalni smeri.
5. Stanjšamo robove z morfološko operacijo tanjšanja.

Potrebnost je omeniti, da smo ugotovili, da je algoritem deloval boljše če smo pred tanjšanjem, morfološko izločili meje objektov (angl. Morphological Boundary Extraction).

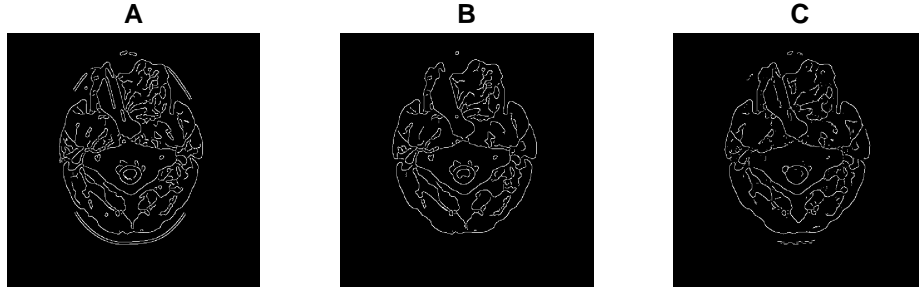
Primerjava vseh treh pristopov je vidna na sliki 1.



**Fig. 1.** Končni rezultat detekcije robov za tri različne načine povezovanja robov: osnovni algoritem (A), ena komponenta naenkrat (B) in lokalno procesiranje (C). Vrednosti parametrov:  $TH = 0.2 * \max(\text{magnituda})$ ,  $TL = 0.1 * \max(\text{magnituda})$ ,  $TM = 0.3 * \max(\text{magnituda})$ ,  $A = 0$ ,  $AT = 180$ ,  $K = 0.05 * \text{širinaSlike}$ .

### 2.3 Morfološko odpiranje

Po preverjanju načinov povezovanja robov se je najbolje izkazal algoritem Ena komponenta naenkrat zato smo ga poskusili izboljšati še z morfološko operacijo odpiranja, ki odstrani majhne slabo povezane podrobnosti v sliki. Ker operacija ne deluje na že stanjšanih robovih smo odpiranje preizkusili na vhodni sliki in na sliki magnitud pred operacijo tanjšanja. Pri tem smo morali sliko po odpiranju še enkrat zgladiti, da zmanjšamo vpliv šuma. Na sliki 2 so prikazani rezultati testiranja.



**Fig. 2.** Primerjava končnih rezultatov detekcije robov brez morfološke operacije odpiranja (A), z odpiranjem vhodne slike (B) in z odpiranjem magnitude (C). Vrednosti parametrov:  $TH = 0.2 * \max(magnituda)$ ,  $TL = 0.1 * \max(magnituda)$ , velikost kvadratnega operatorja je enaka 5x5 pikslov.

## 3 Rezultati

Z algoritmom za povezovanje robov Ena komponenta naenkrat, brez morfološkega odpiranja smo testirali detekcijo robov na sedmih primerih. Rezultati so vidni na slikah 3, 4, 5, 6, 7, 8, 9 v prilogi A. Vrednosti parametrov smo s testiranjem ocenili na  $TH = 0.2 * \max(magnituda)$ ,  $TL = 0.1 * \max(magnituda)$ . V primeru 6 pa smo zmanjšali mejne vrednosti na  $TH = 0.1 * \max(magnituda)$ ,  $TL = 0.05 * \max(magnituda)$  in tako pridobili več robov za ta primer.

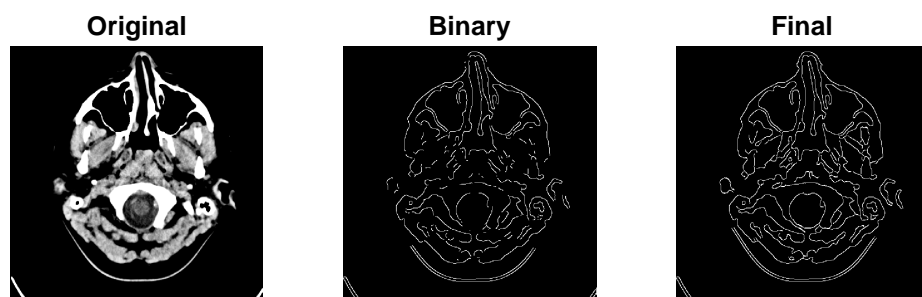
## 4 Zaključek

Implementirali smo Cannyjev algoritem za detekcijo robov in preizkusili tri pristope povezovanja robov. Za najboljšega se je izkazal algoritem Ena komponenta naenkrat, za najslabšega pa Lokalno procesiranje. Preizkusili smo tudi vpliv morfološkega odpiranja na algoritem in ugotovili, da lahko izbriše tudi ključne robove zato ne priporočamo njegovo uporabo pri računalniški tomografiji. V nadaljnjih raziskavah bi lahko preizkusili algoritem Globalno procesiranje s Houghovo transformacijo (opisano v knjigi [2]).

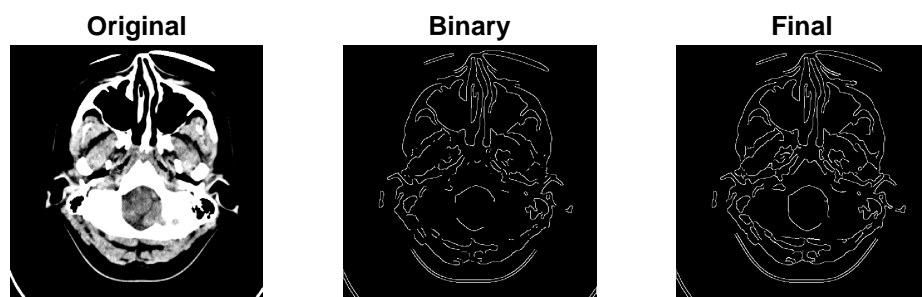
## Literatura

1. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-8**(6), 679–698 (Nov 1986). <https://doi.org/10.1109/TPAMI.1986.4767851>
2. Gonzalez, R.C., Woods, R.E.: Digital image processing. Prentice Hall, Upper Saddle River, N.J. (2008), <http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>
3. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(6), 583–598 (June 1991). <https://doi.org/10.1109/34.87344>

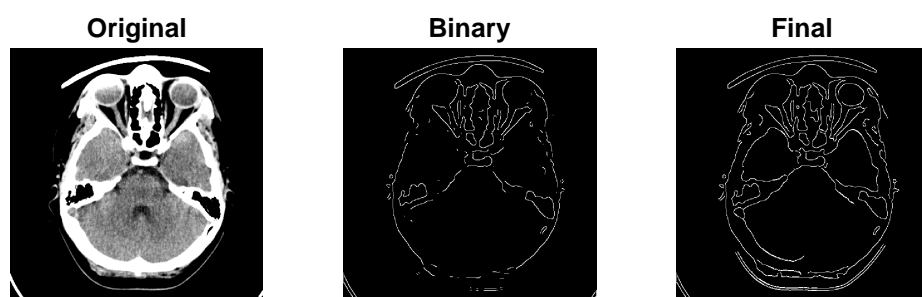
## A Priloga



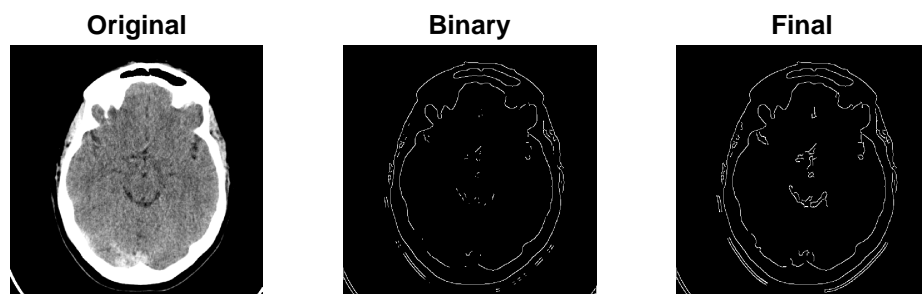
**Fig. 3.** Primer delovanja detekcije robov v treh korakih: vhodna slika (Original). binarna slika robov (Binary) in končna slika po povezovanju robov (Final).



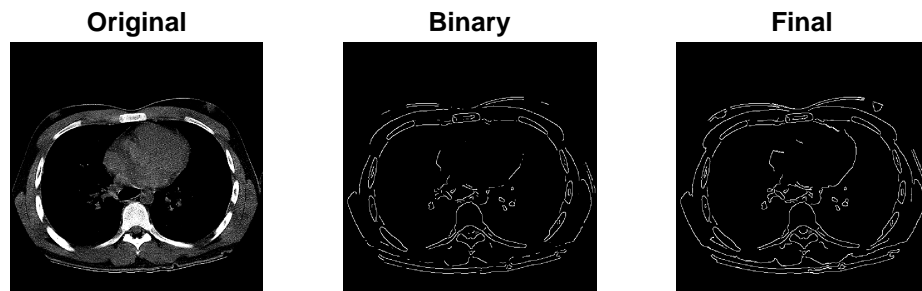
**Fig. 4.** Primer delovanja detekcije robov v treh korakih: vhodna slika (Original). binarna slika robov (Binary) in končna slika po povezovanju robov (Final).



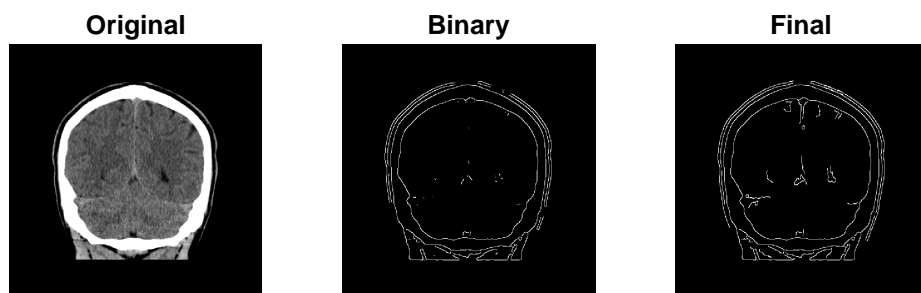
**Fig. 5.** Primer delovanja detekcije robov v treh korakih: vhodna slika (Original). binarna slika robov (Binary) in končna slika po povezovanju robov (Final).



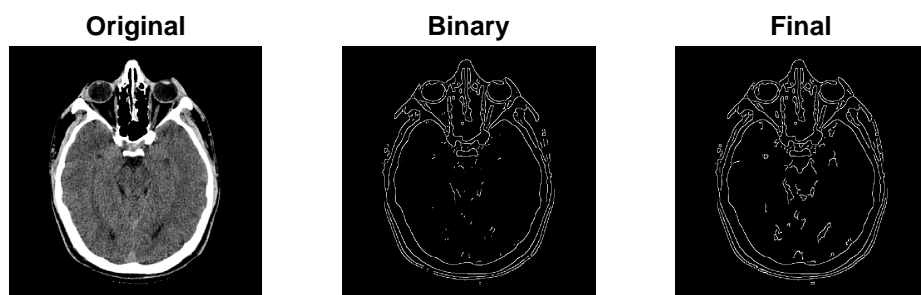
**Fig. 6.** Primer delovanja detekcije robov v treh korakih: vhodna slika (Original). binarna slika robov (Binary) in končna slika po povezovanju robov (Final).



**Fig. 7.** Primer delovanja detekcije robov v treh korakih: vhodna slika (Original). binarna slika robov (Binary) in končna slika po povezovanju robov (Final).



**Fig. 8.** Primer delovanja detekcije robov v treh korakih: vhodna slika (Original), binarna slika robov (Binary) in končna slika po povezovanju robov (Final).



**Fig. 9.** Primer delovanja detekcije robov v treh korakih: vhodna slika (Original), binarna slika robov (Binary) in končna slika po povezovanju robov (Final).