Algoritem za detekcijo QRS kompleksov

L Seminar

Obdelava biomedicinskih signalov in slik 2020/21, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

Peter Horvat [63160129]

I. Povzetek

Za detekcijo QRS kompleksov je na voljo veliko različnih algoritmov, en izmed njih je tudi algoritem, avtorjev HC Chen in SW Chen[1], ki detekcijo opredeli v tri zaporedne korake. S pomočjo filtriranja in iskanja lokalnih maksimumov algoritem zazna QRS kompleks oziroma srčni utrip[2]. Na podlagi testiranja je razvidno, da lahko algoritem pravilno zazna več kot 99,5% srčnih utripov.

II. Uvod

Implementacija, ki je v nadaljevanju opisana je bila razvita v okolju Matlab ter testirana na MIT-BIH bazi[3]. Algoritem deluje na podlagi algoritma HC Chen in SW Chen[1]. Razdeljen je v tri faze. V prvi fazi nad vhodnim signalom, ki je bil pretvorjen iz .dat in .hea datotek v .mat datoteko, uporabimo visokopasovni filter, kot je opisano v dokumentaciji. Temu sledi nizkopasovno filtriranje. V zadnji fazi pa lagoritem na podlagi specifikacij določi ali je v zaznani točki dejansko bil srčni utrip. Zaznani srčni utripi se nato shranijo seznam, ki se v postopku evalviranja uporabi za testiranje. Za testiranje se rezultati algoritma primerjajo z rezultati, ki so bili ročno pregledani s strani strokovnjakov. Končni rezultat so datoteke, ki vsebujejo točke kjer je algoritem zaznal srčni utrip, poleg tega pa tudi tekstovna datoteka z analizo algoritma. Ta vsebuje podatke o uspešnosti algoritma in rezultate za posamezne signale.

III. METODOLOGIJA

Algoritem je razdeljen v tri faze, vsaka izmed faz je ključnega pomena za pravilno delovanje algoritma. Prav tako pa je pomembno tudi pravilno zaporedje filtriranja, prvo visokopasovno filtriranje in nato nizkopasovno filtriranje. V nadaljevanju je opisana implementacija posameznih korakov ter dodatne izboljšave, ki niso omenjene v dokumentaciji[1]. Poleg tega pa je opisan tudi postopek testiranja in pretvorba .dat in .hea datotek v .mat datoteko.

• Pretvorba signala v .mat format

Za pretvorbo signala v .mat format sem uporabil ukaz wfdb2mat, ki za vsako datoteko iz baze, ki ima pripadajočo .hea datoteko ustvari datoteko formata .mat, ki jo v nadaljevanju algoritem prejme, kot parameter.

• Visokopasovno filtriranje

Pri visokopasovnem filtriranju sem sledil postopkom, ki so opisani v dokumentaciji[1]. Zračunal sem vrednost y1 in vrednost y2 ter rezultat shranil v istoimensko spremenljivko. Nato sem od y2 odštel y1. Pri tem sem parameter M nastavil na vrednost 7. V dokumentaciji[1] je sicer bila za testiranje uporabljena vrednost 5 vendar sem se na podlagi rezultatov testiranja odločil za vrednost 7 saj so bili pri tej vrednosti rezultati boljši. Za seštevanje sem uporabil vgrajeno funkcijo sum().

Nizkopasovno filtriranje

V fazi nizkopasovnega testiranja sem ponovno sledil navodilom v dokumentaciji[1]. Z uporabo for zanke

sem iteriral čez rezultat visokopasovnega filtriranja. Pri tem sem za interval seštevanja izbral vrednost 30 saj je ta vrednost vračala najboljše rezultate. Preden sem vrednosti seštel pa sem posamezno vrednost še kvadriral. Za seštevanje sem ponovno uporabil vgrajeno funkcijo sum().

• Iskanje QRS kompleksov

Za detekcijo QRS kompleksov sem na začetku definiral prag, ki je enak lokalnemu maksimumu na začetnem delu seznamu vrednosti. Ta del seznama je enak rezultatu nizkopasovnega filtriranja med indeksoma 1 in premikajočega okna. Nato sem z uporabo for zanke iteriral čez prej omenjen seznam z koram, ki je enak vrednosti premikajočega okna. Za velikost premikajočega okna sem ugotovil, da dobim najboljše rezultate, če je vrednost nastavljena na 90. V vsakem koraku iteracije sem določil lokalni maksimum, ter ga primerjal s pragom. V koliko je bil lokalni maksimum večji od praga sem preveril še ali je v zaznani točki dejanski srčni utrip ali kakšna anomalija. To sem implementiral tako, da sem v točki, ki kjer je bil zaznan domnevni srčni utrip pogledal v preteklost in prihodnost in preveril ali več vrednosti presega prag. V koliko 6 vrednosti presega prag v vsako smer, se algoritem nadaljuje v zadnjo fazo odločevanja ali je v točki srčni utrip ali ne. Zadnja stopnja testiranja preverja ali je trenutno zaznani domnevni srčni utrip prehitro sledil prejšnjemu. Testiranje je pokazalo, da ta dodatna funkcionalnost pri detekciji poveča uspešnost algoritma za približno 19%. V koliko zaznani srčni utrip uspešno izpolni vse kriterije se točko v kateri je bil zaznan srčni utrip doda v seznam. To pa je implementirano tako, da v točko v rezultatu prejšnje faze, kjer je bil zaznan srčni utrip vnesem številko 1. Kot omenjeno v dokumentaciji[1], se v primeru detekciji srčnega utripa posodobi tudi prag z uporabo formule iz dokumentacije[1]. Na koncu sledi le še funkcija fidn(), ki poišče vse vrednosti 1 v seznamu ter vrne vektor, ki vsebuje vse indekse oziroma točke v katerih je bil zaznan srčni utrip.

Evalvacija

Za evalviranje sem uporabi ukaz bxb[4], ki premerja rezultat algoritma z ročno določenimi srčnimi utripi. Na podlagi primerjav se ustvari datoteka v kateri so shranjeni rezultati za vsako .mat datoteko iz baze MIT-BIH[3]. Zraven tega pa se v datoteko doda tudi izračunana občutljivost in odstotek pravilnega napovedovanja srčnih utripov. Vrednosti sta odstotka povprečja vseh odstotkov za posamezno datoteko. S tem dobimo boljšo predstavo o tem kako uspešen in natančen je algoritem pri detekciji srčnih utripov oziroma QRS kompleksov.

```
#I/Sin/bash

if (=f.up.nut]; then

in wup_tit

if (=f.up.nut]; then

if (=f.up.nut); the
```

Figure 1. Skripta za evalviranje rezultatov

Dodane izboljšave

Dodani izboljšavi v algoritmu sta dve, prva na podlagi preteklih in prihodnjih točk v signalu določi ali domnevni srčni utrip ni samo hiter narastek signala zaradi šumov, napak pri beleženju ali kaj podobnega. Druga izboljšava pa spremlja ali je bil prejšnji srčni utrip v neposredni bližini drugega. S tem algoritem izloči točke, ki so v neposredni bližini dejanskega dejanskega srčnega utripa.

• Detekcija dejanskih QRS kompleksov

Figure 2. Programska koda iskanja QRS kompleksov

IV. Rezultati

Za izboljševanje algoritma sem algoritem večkrat testiral z različnimi parametri. Najboljše rezultate je algoritem vračal pri naslednjih vrednostih parametrov.

| Vrednost |
|----------|
| 7 |
| 30 |
| 0.6 |
| 0.16 |
| 90 |
| 6 |
| 2 |
| |

Povprečna vrednost občutljivosti algoritma je 97,52%, odstotek pravilno zaznanih srčnih utripov pa je 98,10%. Ti rezultati so izračunani na podlagi signalov iz MIT-BIH baze[3]. Odstotek se lahko spremeni, če uporabimo drugo bazo.

V. Diskusija

Prej opisan algoritem je že dober približek algoritma, ki je opisan v dokumentaciji[3]. Tam namreč piše, da algoritem

pravilno zazna 99,5% srčnih utripov. Rezultati mojega algoritma bi se lahko izboljšali z uporabo več dodatni izboljšav, podobnih tema, ki sta že dodani. Na ta način bi šume in zunanje dejavniki lahko skoraj popolnoma izločili. S tem bi se približali omenjeni uspešnosti 99,5%.

References

- H. Chen and S.-W. Chen, "A moving average based filtering system with its application to real-time qrs detection," in *Computers in Cardiology*. IEEE, 2003, pp. 585–588.
- in Cardiology. IEEE, 2003, pp. 585–588.
 "Qrs complex," Accesible: https://en.wikipedia.org/wiki/QRS_complex, [Accesed: 10.12.2020].
- [3] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Eng in Med and Biol*, vol. 20, no. 3, pp. 45–50, 2001.
- [4] "Physionet bxb," Accesible: https://www.physionet.org/ physiotools/wag/bxb-1.htm, [Accesed: 10.12.2020].