

Segmentacija volumetričnih podatkov z modelom aktivnih krivulj (Snake)

Ivan Antešić¹

¹ Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Slovenija

Povzetek

V okviru seminarske naloge sem implementiral model aktivnih krivulj (znan tudi kot Snake). Model sem uporabil na množici volumetričnih podatkov celice in segmentiral strukture mitohondrijev. V sledečem poročilu na kratko opišem Snake model, možne razširitve in mojo implementacijo. Na koncu so predstavljeni in komentirani rezultati segmentacije.

1. Uvod

Segmentacija (ločevanje pikslov strukture od preostalih pikslov slike) je ključen prvi korak pri analiziranju bioloških in medicinskih slik. Segmentirane podatke lahko vizualiziramo, jim sledimo ter tako odkrivamo lastnosti opazovane strukture. Ročno segmentiranje je dolgotrajno in naporno, zato želimo proces avtomatizirati. Ker so napake v interpretaciji medicinskih slik nedopustne, je namesto popolne avtomatizacije potrebno narediti algoritem, ki človeku olajša in pohitri delo, še vedno pa dopušča ročno določitev okvirne strukture. Točno za ta problem je primerna uporaba modela Snake.

2. Pregled področja

2.1. Snake

Snake je model aktivnih parametričnih krivulj s katerimi iščemo meje struktur v binarnih (dvobarvnih) slikah [KWT88]. Kompleksna krivulja je sestavljena iz zlepkov večjih preprostih krivulj, ki povezujejo njena vozlišča. Na krivuljo delujemo z določenimi silami ter jo tako preoblikujemo, da se čimbolj prilaga robu iskane strukture.

Krivulja je v spodnji formuli definirana z energijo E . v je oznaka za vozlišča krivulje, odvisnih od časa t in parametrične spremenljivke s na intervalu $(0,1)$. I predstavlja notranjo P pa zunanjo energijo.

$$\int_0^1 E(v) = \int_0^1 I(v(s)) + \int_0^1 P(v(s)) \quad (1)$$

I določi v kakšni meri se krivulja lahko definira, P pa poveže obliko krivulje z intenziteto slike. Na začetku je vozlišča potrebno postaviti v strukturo in določiti začetno obliko krivulje, ki jo algoritem deformira. Končna krivulja bo enaka obliki pri minimumu E .

2.2. T-Snake

T-Snake ali model topoloških aktivnih krivulj je nadgradnja prvotnega Snake modela. Model ohrani možnost ročnega vpliva na iskanje strukture, vendar poskuša proces bolj avtomatizirati z povečanjem neodvisnosti deformacije krivulje od začetne postavitve vozlišč [MT00].

To dosežemo, tako da sliko z triangulacijo razdelimo na več trikotnikov in po določenem številu korakov originalnega Snake algoritma, krivuljo reparametriziramo. Točke kjer krivulja seka robne trikotnike (tiste trikotnike, ki ležijo na meji trenutne strukture) se vzame za nova vozlišča. Poleg večje topološke neodvisnosti ima tak pristop prednost, da posledično zaradi reparametrizacije poveže več sekajočih krivulj v eno samo. Kot že rečeno, T-snake za osnovni del uporablja originalni Snake algoritem, ki ga zaradi lažje implementacije diskretiziramo.

Naslednja formula, nam pove, da je položaj i -tega vozlišča x_i v času $(t+\Delta t)$ enak prejšnjemu položaju v času t , ki ga deformirajo notranje in zunanje sile.

$$x_i^{(t+\Delta t)} = x_i^{(t)} - \frac{\Delta t}{\gamma} (a\alpha_i^{(t)} + b\alpha_i^{(t)} - p_i^{(t)} - f_i^{(t)}) \quad (2)$$

$\alpha(t)$ in $\alpha(t)$ predstavljata notranji sili, ki stremita k temu, da ohranjata razdalje in položaje med sosednjimi vozlišči ter tako nasprotujeta zunanjim silam. Parametra a in b predstavljata uteži sil s katerima lahko prilagajamo algoritem.

p je napihovalna sila, ki vozlišče potisne navzven v smeri njegove normale proti robu strukture, ali navznoter, če vozlišče leži zunaj iskane strukture. To doseže z množenjem normale z $+1$ če vozlišče leži na območju pikslov intenzitete 1, ali pa obrne smer z množenjem z -1 če se vozlišče nahaja okoli pikslov intenzitete 0. Ker se vozlišče lahko nahaja med piksli je potrebno za določitev vrednosti bilinearno interpolirati sosednje intezitete. To

nam omogoči, da strukturo segmentiramo z večjo natančnostjo, kot pa je resolucija slike.

f je sila, ki skrbi, da smer deformacije usmerja proti pomembnim robovom z računanjem gradienta slike, ki jo filtriramo z Gaussovim filtrom za iskanje robov.

2.3. Dual-T-Snakes

Naslednja nadgradnja algoritma je model dveh topoloških aktivnih krivulj, kjer naenkrat uporabimo dve krivulji za segmentiranje - eno zunaj strukture ki se krči in drugo, ki se nahaja znotraj strukture in se širi proti robu. Takšen pristop omogoča večjo robustnost in odpornost na artefakte slike, še posebej različne šume, ki lahko pokavrijo minimum energije krivulje E . Cena izboljšave je večja poraba virov [GSO03].

3. Implementacija

Za segmentacijo sem implementiral diskretiziran Snake model, ki ga za osnovo uporablja T-snake. Rešitev sem napisal v programskem okolju in jeziku Matlab. Vhodni podatki so binarne slike velikosti 250x250 pisklov, ki predstavljajo rezine volumetričnih podatkov celičnih struktur mitohondrija. Ena od teh rezin je prikazana na sliki 1.

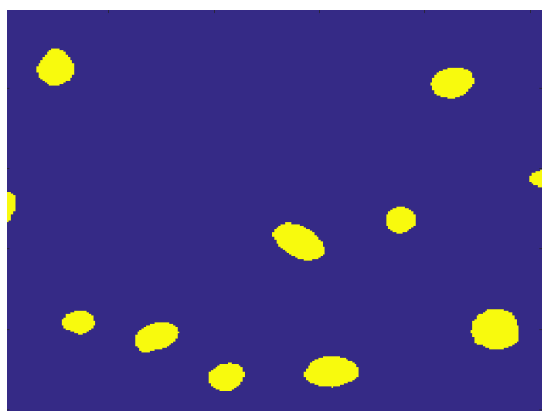


Figure 1: Binarna slika, ki predstavlja rezino mitohondrijev (rumene strukture).

Algoritem se požene z izbiro središča strukture, ki jo želimo segmentirati. Okoli središča se nato generirajo vozlišča krivulje. V zanki vsakemu vozlišču spreminjamo položaj z notranjimi in zunanji silami (točne formule posameznih sil so opisane v članuku [MT00]), glede na izračunane normale. Z pravilno nastavitvijo uteži sil vsak časovni korak premaknemo vozlišča proti robu, kar je vidno na sliki 2.

Ko vozlišča padejo izven mej strukture se premaknejo v obratno smer. Da krivulja ne bi v nedogled oscilirala okoli roba je potrebno določiti ustavitvene pogoje. Vsakič, ko algoritem zazna spremembo smeri se zmanjša vpliv napihovalne sile, kar postopoma zmanjšuje dolžino premika in ustali vozlišča v njihov končni položaj. Če se napihovalna sila zmanjša na 0 ali pa če se vozlišča v večih iteracijah ne premaknejo dovolj iz svojega položaja, ustavimo algoritem.

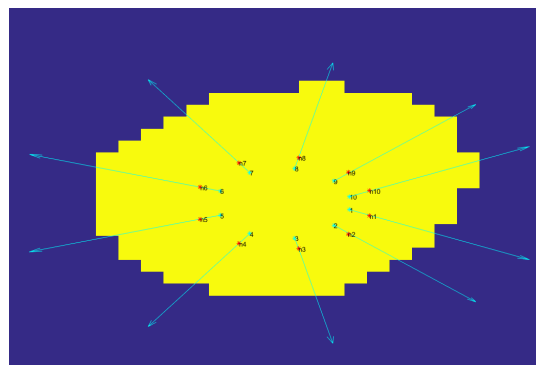


Figure 2: Nov položaj vozlišč krivulje (rdeče točke), ki so se v smeri normal premaknili iz prejšnjega stanja proti robu strukture.

Poleg tega za vsak slučaj še omejim število iteracij v katerih se premikajo vozlišča.

Po končani segmentaciji vozlišča ležijo na robu strukture (slika 3). Njihove koordinate nato zapišemo v tekstovno datoteko. Iz končnih vozlišč se izračuna novo središče in celoten algoritem se ponovi za vsako rezino, kjer se nahaja izbrana struktura. Vsako rezino preverimo ali se vozlišča še nahajajo v strukturi in ko se ne ustavimo algoritem.

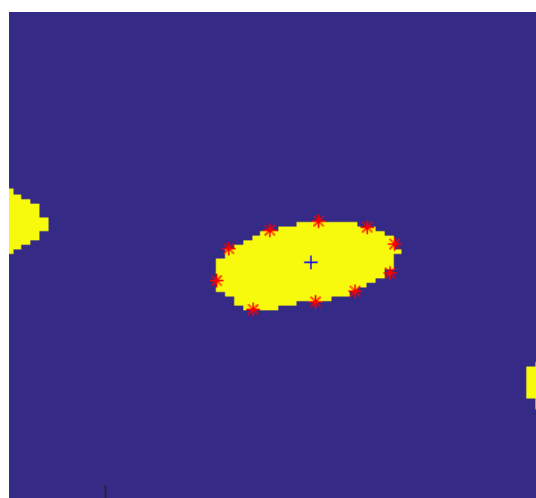


Figure 3: Vozlišča krivulje, označena z rdečimi pikami, segmentirajo izbrano strukturo.

4. Rezultati

Po končanem algoritmu lahko tekstovno datoteko, ki vsebuje koordinate vozlišč krivulje na vseh rezinah, vizualiziramo kot oblak točk in vidimo obliko segmentiranega mitohondrija. Za podane volumetrične podatke algoritem deluje dobro, saj so vse strukture podobnih, eliptičnih oblik, kar je primerno za segmentiranje z netopološkim Snake modelom. Med testiranjem je algoritem praviloma zajel vse željene strukture in ni nikoli zašel.

Časovna zahtevnost algoritma je $O(k*j*i)$: je k število rezin, na katerih se nahaja segmentirana struktura, j je število iteracij potrebnih za segmentacijo ene rezine, i pa število vozlišč krivulje.

Za segmentacijo iste strukture, ki leži na 48 rezinah, algoritem potrebuje za 10 točk povprečno 27 sekund, z povprečnim številom iteracij j enakim 11. Za 50 točk z povprečno sedmimi j iteracijami potrebuje 75 sekund. Za 100 točk porabi 6 iteracij in 130 sekund. Število iteracij z povečanjem števila vozlišč pada, ker se večkrat zamenja smer in posledično zmanjša vpliv napihovalne sile. Posledica je manjša natančnost segmentacije, ki jo odpravimo z povečanjem uteži napihovalne sile.

Kot je vidno na slikah 4 in 5, je struktura bolj podrobno segmentirana z večjim številom vozlišč. Izbrati več kot 50 vozlišč za podano velikost slike, ni smiselno saj se posamezne točke zlijejo skupaj v krivuljo, s tem pa ne pridobimo dovolj na dodatni podrobnosti, da bi upravičili počasnejše delovanje.

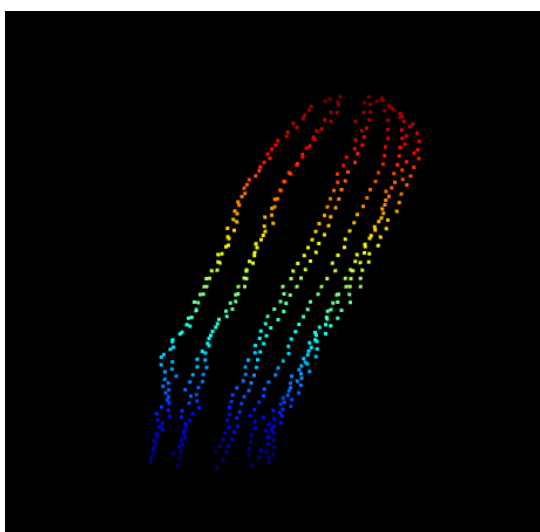


Figure 4: Oblak točk za segmentirani mitohondrij z desetimi vozlišči krivulje.

5. Zaključek

Za segmentacijom volumetričnih podatkov mitohondrijev sem implementiral osnovni diskretizirani model aktivnih krivulj Snake. Na podanih primerih algoritem deluje zadovoljivo natančno in hitro. Če bi imeli podane strukture bolj različnih ali zapletenih oblik, osnovni algoritem ne bi bil več primeren. V takem primeri, bi morali implementirati reparametrizacijo modela T-snake ali uporabiti model dveh krivulj Dual-T-Snakes, če bi vhodni podatki imeli dodan šum.

Literatura

- [GSO03] GIRALDI G., STRAUSS E., OLIVEIRA A.: Dual-t-snakes model for medical imaging segmentation. *Pattern Recogn. Lett.* 24, 7 (Apr. 2003), 993–1003. URL: [http://dx.doi.org/10.1016/S0167-8655\(02\)00223-4](http://dx.doi.org/10.1016/S0167-8655(02)00223-4), doi:10.1016/S0167-8655(02)00223-4. 2

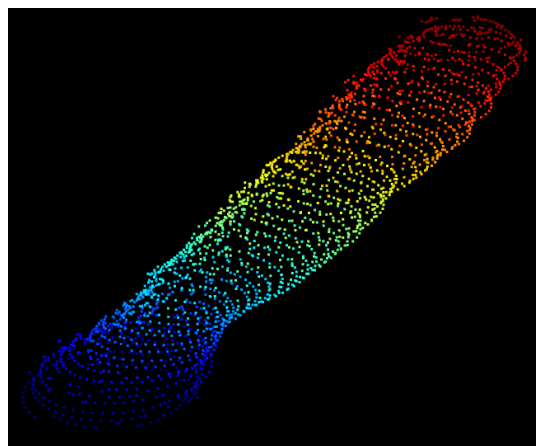


Figure 5: Oblak točk za segmentirani mitohondrij z petdesetimi vozlišči krivulje.

- [KWT88] KASS M., WITKIN A., TERZOPOULOS D.: Snakes: Active contour models. *International Journal of Computer Vision* 1, 4 (Jan 1988), 321–331. URL: <https://doi.org/10.1007/BF00133570>, doi:10.1007/BF00133570. 1
- [MT00] MCINERNEY T., TERZOPOULOS D.: T-snakes: Topology adaptive snakes. *Medical Image Analysis* 4, 2 (2000), 73–91. URL: <http://nukweb.nuk.uni-lj.si/login?url=http://search.ebscohost.com/login.aspx?authtype=ip&custid=s6232602&profile=eds.1,2>