

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 5703

**Određivanje heliciteta mikrotubula u
mikroskopskim 3D slikama**

Ivan Barišić

Zagreb, srpanj 2018.

Zahvaljujem mentoru prof. dr. sc. Siniši Šegviću na pomoći pri izradi ovog rada, Barbari Kuzmić iz Grupe Tolić s IRB-a na pomoći pri izradi biološkog opisa te mojoj obitelji na bezuvjetnoj podršci tijekom čitavog školovanja.

Sadržaj

1. Uvod	1
2. Biološki opis.....	2
3. Ulazni podaci.....	5
4. Korišteni algoritmi i metode	6
4.1. Gaussovo zaglađivanje	6
4.2. Laplaceov filtar i filtar Meksički šešir	8
4.3. Optički tok	9
4.3.1. Procjena gibanja između dvije slike na temelju proširivanja polinoma	11
4.4. Algoritmi za računanje heliciteta	13
4.4.1. Pomoćne funkcije.....	13
4.4.2. Algoritam centra blobova.....	15
4.4.3. Algoritam 8-susjedstva.....	17
4.4.4. Algoritam čitavih blobova	17
4.4.5. Algoritam svih piksela	19
5. Eksperimentalni rezultati	20
5.1. Pretprocesiranje slike.....	20
5.2. Optički tok	22
5.3. Helicitet	24
6. Zaključak.....	28
Literatura.....	30

1. Uvod

Računalni vid je grana umjetne inteligencije koja računalima pokušava omogućiti razumijevanje slika i videa. Metode i primjene računalnog vida su brojne, npr. u uređajima za interakciju s ljudima, u robotima i autonomnim vozilima za kretanje u prostoru, u medicinskoj obradi slike za postavljanje dijagnoze pacijentima ili u analizi mikroskopskih bioloških slika za što će se primjenjivati u ovom radu. Jedna od grana računalnog vida je optički tok kojim se određuje pomak elemenata slike između dvije uzastopne slike. Optički tok osobito se primjenjuje u robotici i praćenju objekata.

Stanice su osnovni građevni elementi svih živih organizama. Nove stanice nastaju procesom stanične diobe i tako se omogućuje razvoj višestaničnih organizama. Diobeno vreteno je stanična struktura koja sudjeluje u staničnoj diobi. Sastoji se od cjevastih struktura zvanih mikrotubuli. Mikrotubuli se protežu između polova diobenog vretena čineći zakrivljene lukove. Osim što čine zakrivljene lukove, mikrotubuli čine uvrnute snopove opisane helicitetom. Helicitet je faktor uvrtnja mikrotubula oko glavne osi diobenog vretena u smjeru kazaljke na satu i mjeri se u stupnjevima po određenoj udaljenosti. Helicitet mikrotubula daje uvid u sile koje sudjeluju u trenutno nepoznatim detaljima stanične diobe.

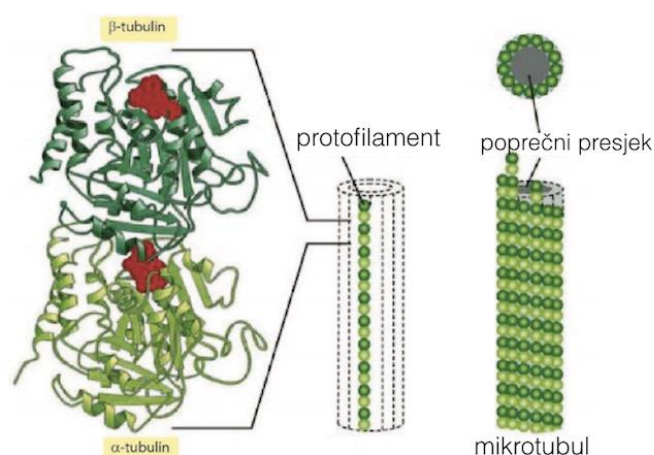
Ovaj rad pokušava pomoću optičkog toka odrediti helicitet mikrotubula iz 3D mikroskopskih slika. Osmišljena su četiri algoritma za računanje heliciteta. Ideja za algoritme preuzeta je iz diplomskog rada Zvonimira Bobana [1] gdje se helicitet računao na temelju ručnih oznaka, a algoritmi ovog rada računaju helicitet bez potrebe za ručnim označavanjem. Prije računanja heliciteta, potrebno je obraditi slike na odgovarajući način da bi se istaknuli mikrotubuli i da bi se uklonio šum na mikroskopskim slikama. To predstavlja izazov jer slike imaju mnogo šuma. Rezultati algoritama uspoređuju se s ručno izračunatim rezultatima iz Grupe Tolić s Instituta Ruđer Bošković na temelju [1].

U sljedećem poglavlju daje se detaljniji biološki opis. Zatim se opisuju metode za uklanjanje šuma i isticanje mikrotubula na mikroskopskim slikama. Nakon toga slijede matematički opisi optičkog toka općenito i konkretne metode računanja optičkog toka

korištene u ovom radu. U poglavlju iza toga opisani su osmišljeni algoritmi. Onda su dani eksperimentalni rezultati i naposljetku zaključak.

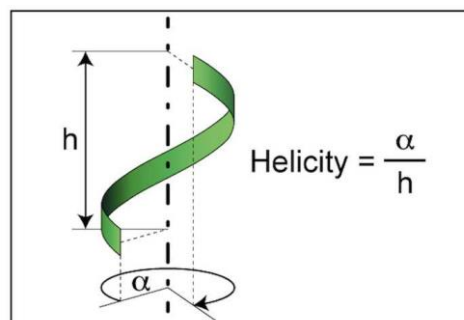
2. Biološki opis

Stanična dioba je složeni proces kojim od jedne stanice nastaju dvije nove stanice kćeri. Sastoji se od mitoze, u kojoj dolazi do podjele stanične jezgre, i citokineze, kod koje se podijeli ostatak stanice [2]. Mitoza i citokineza zajedno traju oko sat vremena, a prosječna stanica se dijeli svakih 24 sata [3]. Proces stanične diobe omogućuje rast i razvoj višestaničnog organizma. Ovim procesom dolazi do pravilne raspodjele genetičkog materijala između novonastalih stanica. Pravilnu podjelu kromosoma omogućuje kompleksna stanična struktura, nazvana diobeno vreteno. Diobeno vreteno, prikazano na slici 3, sastavljeno je od mikrotubula i pripadajućih proteina [4]. Mikrotubuli su proteinske cjevčice sastavljene od α i β podjedinica tubulina. α i β podjedinice međusobno su djelomično isprepletene. Jedan mikrotubul sadrži 13 paralelnih protofilamenata koji tvore cjevčicu. S obzirom na tip podjedinice koji se nalazi na kraju mikrotubula razlikujemo (+) i (-) krajeve mikrotubula. Ako mikrotubul završava β podjedinicom, radi se o (+) kraju mikrotubula. U slučaju da je posljednja podjedinica α tubulin, radi se o (-) kraju mikrotubula. (+) kraj mikrotubula je dinamičniji što znači da brže raste i skraćuje se. Mogućnost rasta i skraćivanja mikrotubula naziva se dinamička nestabilnost [2].

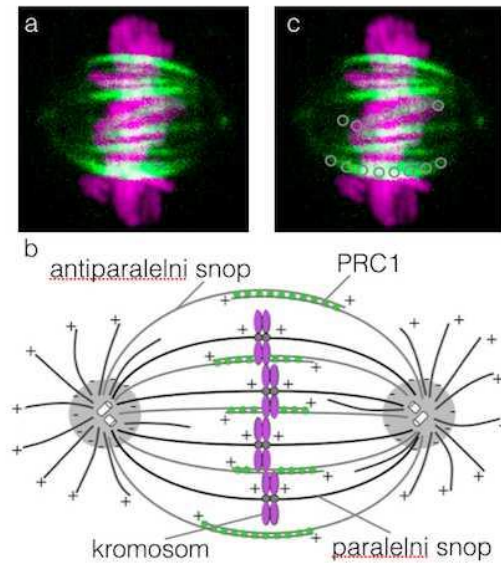


Slika 1. Prikaz pojedinačnog mikrotubula koji je građen od 13 protofilamenata čije su podjedinice α i β tubulin. [6]

U diobenom vretenu ljudskih stanica, mikrotubuli su organizirani u snopove. Pojedini snop sadrži od 12 do 22 paralelno povezana mikrotubula. Ovisno o međusobnoj orijentaciji mikrotubula razlikujemo paralelne i antiparalelne snopove. Paralelni snopovi su povezani snopovi koji izlaze iz istog pola diobenog vretena i vežu se na kinetohoru - proteinski kompleks na kromosomu. Na kromosom se veže po jedan paralelni snop iz oba pola diobenog vretena. U slučaju kad mikrotubuli rastu iz suprotnih polova diobenog vretena i povezuju se u središnjem dijelu, tvore antiparalelne snopove [4]. Na preklapajućim dijelovima antiparalelnih snopova veže se protein PRC1 (engl. *protein regulator of cytokinesis*). Taj se protein obilježava zelenim fluorescentnim proteinom GFP (engl. *green fluorescent protein*). Time se omogućuje vizualizacija fluorescencijskom mikroskopijom radi određivanja oblika snopova mikrotubula diobenog vretena. Korištene su HeLa stanice (stanice raka grlića maternice). 3D slike metafaznih stanica snimljenih fluorescencijskom mikroskopijom pokazale su kako snopovi mikrotubula tvore različite zakrivljene strukture. Tipičan oblik vanjskih snopova podsjeća na oblik slova C, dok približavanjem središnjoj osi diobenog vretena snopovi sve više podsjećaju na slovo S. S oblici snopova mikrotubula posljedica su zarotiranosti ovih snopova u smjeru kazaljke na satu s obzirom na glavnu os diobenog vretena. Mjera kojom se opisuje rotacija mikrotubula s obzirom na centralnu os nazvana je helicitet mikrotubula [5]. Računa se kao prosječna promjena kuta mikrotubula u odnosu na centralnu os diobenog vretena između dva uzastopna odreska stanice po udaljenosti između dva uzastopna odreska. Izražava se u $^{\circ}/\mu\text{m}$ [1]. Pod navedenim kutom misli se na kutnu komponentu cilindričnog koordinatnog sustava pri čemu z os predstavlja centralna os diobenog vretena. To je kut α na slici 2. U daljnjim pojavama kuta mikrotubula u odnosu na centralnu os diobenog vretena misli se na ovaj kut.



Slika 2. Definicija heliciteta. [6]



Slika 3. a) Prikaz diobenog vretena HeLa stanice snimljene fluorescencijskim konfokalnim mikroskopom. Zelenom bojom obilježen je protein PRC1, a ružičastom bojom kromosomi.

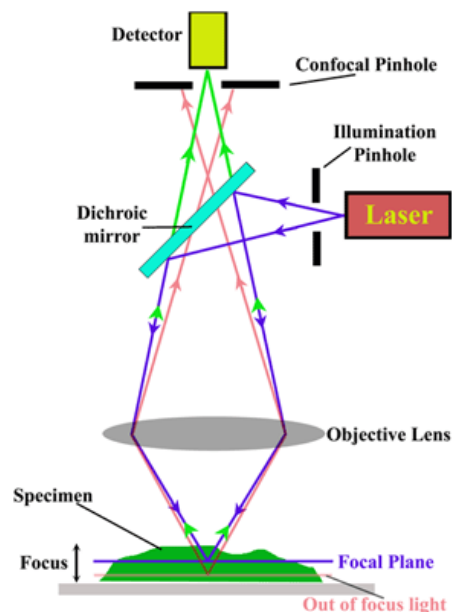
[6]

b) Pojednostavljeni prikaz diobenog vretena ljudskih stanica. Sivom i crnom bojom prikazani su mikrotubuli koji čine diobeno vreteno, zelenom bojom PRC1 koji se veže u antiparalelnim regijama, a ružičastom bojom kromosomi. Označeni su (+) i (-) krajevi snopova mikrotubula. Prilagođeno prema: [7]

c) Prikaz diobenog vretena s označenim različitim oblicima snopova mikrotubula. [6]

Korištenje fluorescencijske mikroskopije u proučavanju živih stanica jedan je od temeljnih alata suvremene stanične biologije. Obilježavanjem staničnih struktura od interesa fluorescentnim proteinima i bojama, koje se specifično vežu na određene proteine, moguće je pratiti procese unutar žive stanice [8]. Pri proučavanju procesa vezanih uz diobeno vreteno vrlo često se koristi konfokalni fluorescencijski mikroskop. Ovu vrstu mikroskopije osmislio je i patentirao 1955. godine Marvin Minsky. Kod ovakvog tipa mikroskopa ulazni otvor detektora nalazi se u optičkoj ravnini koja je konjugirana fokalnoj ravnini objektiva. Konjugirana ravnina P' ravnine P je ravnina u kojoj se nalaze slike točaka smještenih u ravnini P . Na ovaj način se postiže da samo svjetlo koje dolazi iz fokusa dolazi do detektora. Posljedica takvog načina snimanja je dobivanje oštrije i čistije slike [9]. Primjer rada konfokalnog mikroskopa prikazan je na slici 4. Ovakva mikroskopija omogućava optičko seciranje uzorka, odnosno snimanje sloj po sloj. Rekonstrukcijom

snimljenih ravnina dobiva se trodimenzionalna slika snimanih bioloških struktura, u ovom slučaju diobenog vretena [10].

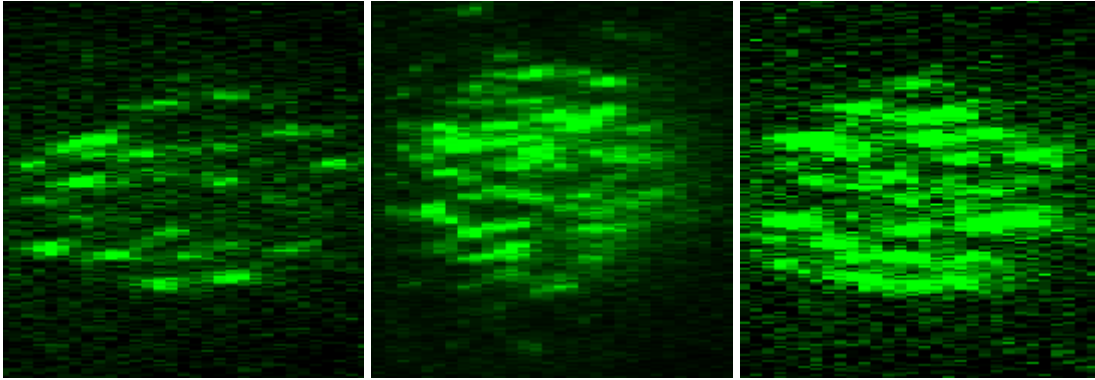


Slika 4. Primjer rada konfokalnog mikroskopa. [24]

3. Ulazni podaci

Slike stanica korištene u ovom radu predstavljene su 3D tenzorima koji su rezultat snimanja stanica konfokalnim mikroskopom. U tom 3D tenzoru, posloženi su odresci (slojevi) stanice jedan na drugoga. Broj odrezaka među stanicama se razlikuje, a najčešće se nalazi u rasponu između 100 i 160. Odrezak na dnu sadrži jedan pol diobenog vretena, a odrezak na vrhu sadrži drugi pol. Svi odresci jedne stanice imaju iste dimenzije, a dimenzije odrezaka mogu se razlikovati od stanice do stanice. Najčešće dimenzije odrezaka su 180x188 piksela. Izvorna duljina odrezaka je nakon snimanja mikroskopom bila 6 puta manja nego na odrescima korištenima za kojima se računanje heliciteta, ali je zbog preglednosti povećana na način da je svaki piksel po duljini ponovljen 6 puta. Na slici 5 prikazano je nekoliko primjera odrezaka različitih stanica.

Sve stanice snimljene su u Grupi Tolić s Instituta Ruđer Bošković.



Slika 5. Primjeri odrezaka različitih stanica.

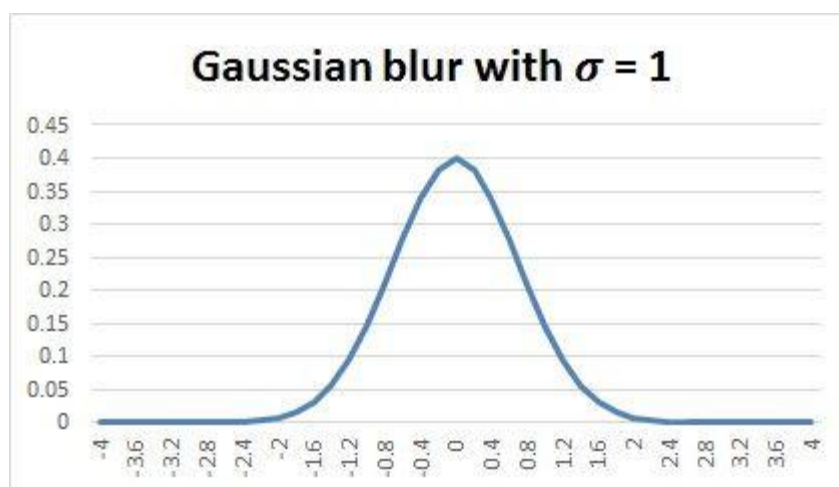
4. Korišteni algoritmi i metode

4.1. Gaussovo zaglađivanje

Gaussovo zaglađivanje je rezultat konvolucije slike i Gaussove funkcije. Tipična uporaba Gaussovog zaglađivanja je uklanjanje šuma i detalja sa slike. Primjenjuje se u pretprocesiranju slika za algoritme računalnog vida.

Jednodimenzionalna Gaussova funkcija uz standardnu devijaciju jednaku σ i očekivanje jednako 0 je:

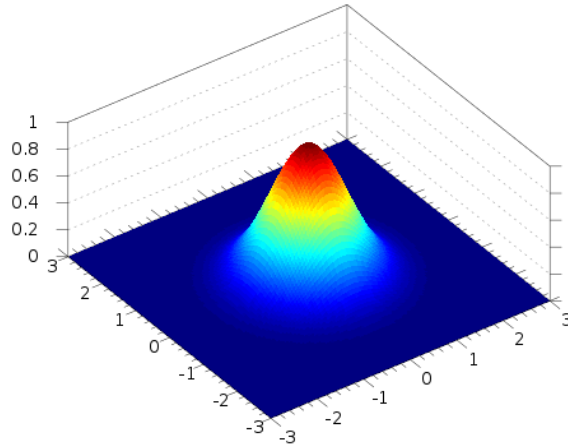
$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$



Slika 6. Jednodimenzionalna Gaussova funkcija uz $\sigma = 1$. [11]

Dvodimenzionalna Gaussova funkcija prikazana na slici 7 umnožak je dvije jednodimenzionalne Gaussove funkcije gdje je svaka u različitoj dimenziji:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$



Slika 7. Primjer dvodimenzionalne Gaussove funkcije. [12]

Množenjem s još jednom 1D Gaussovom funkcijom dobiva se 3D Gaussova funkcija:

$$G(x, y, z) = \frac{1}{\sqrt{(2\pi)^3}\sigma^3} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}} \quad (3)$$

Gaussova funkcija različita je od 0 u svakoj točki, ali u praksi, kada se koristi diskretna aproksimacija Gaussove funkcije, pikseli na udaljenostima većim od 3σ imaju dovoljno mali utjecaj da ih možemo smatrati jednakima 0 [13]. Na slici 8 prikazan je primjer diskretne aproksimacije 2D Gaussove funkcije.

	1	4	7	4	1
	4	16	26	16	4
$\frac{1}{273}$	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Slika 8. Primjer diskretne aproksimacije 2D Gaussove funkcije uz $\sigma = 1$. [14]

Prethodno navedene 2D i 3D Gaussove funkcije nazivaju se izotropnima. To znači da imaju jednake standardne devijacije u svakom smjeru, odnosno u svakoj dimenziji. Kada želimo zaglađivanje provesti u različitoj mjeri u različitim dimenzijama, koristimo anizotropne Gaussove funkcije. Kod njih se standardne devijacije u različitim dimenzijama razlikuju [15].

Kao što je već rečeno, zaglađivanje slike provodi se primjenom konvolucije nad izvornom slikom i Gaussovom funkcijom. Višedimenzionalnu konvoluciju s Gaussovom funkcijom moguće je obaviti tako da se 1D konvolucija primijeni u svakoj dimenziji. Diskretna 1D konvolucija definirana je sljedećom formulom [16]:

$$f(x) * g(x) = \sum_{i=-\infty}^{\infty} f(i)g(x - i) \quad (4)$$

4.2. Laplaceov filtar i filtar Meksički šešir

Laplaceov filtar koristi se za naglašavanje područja na kojima dolazi do nagle promjene intenziteta te se stoga koristi za detekciju rubova. Često se primjenjuje na slike na koje je prethodno primijenjeno zaglađivanje Gausovim filtrom da bi se smanjila njegova osjetljivost na šum.

Laplaceov filtar $L(x, y)$ nad slikom s intenzitetima piksela $I(x, y)$ definira se formulom:

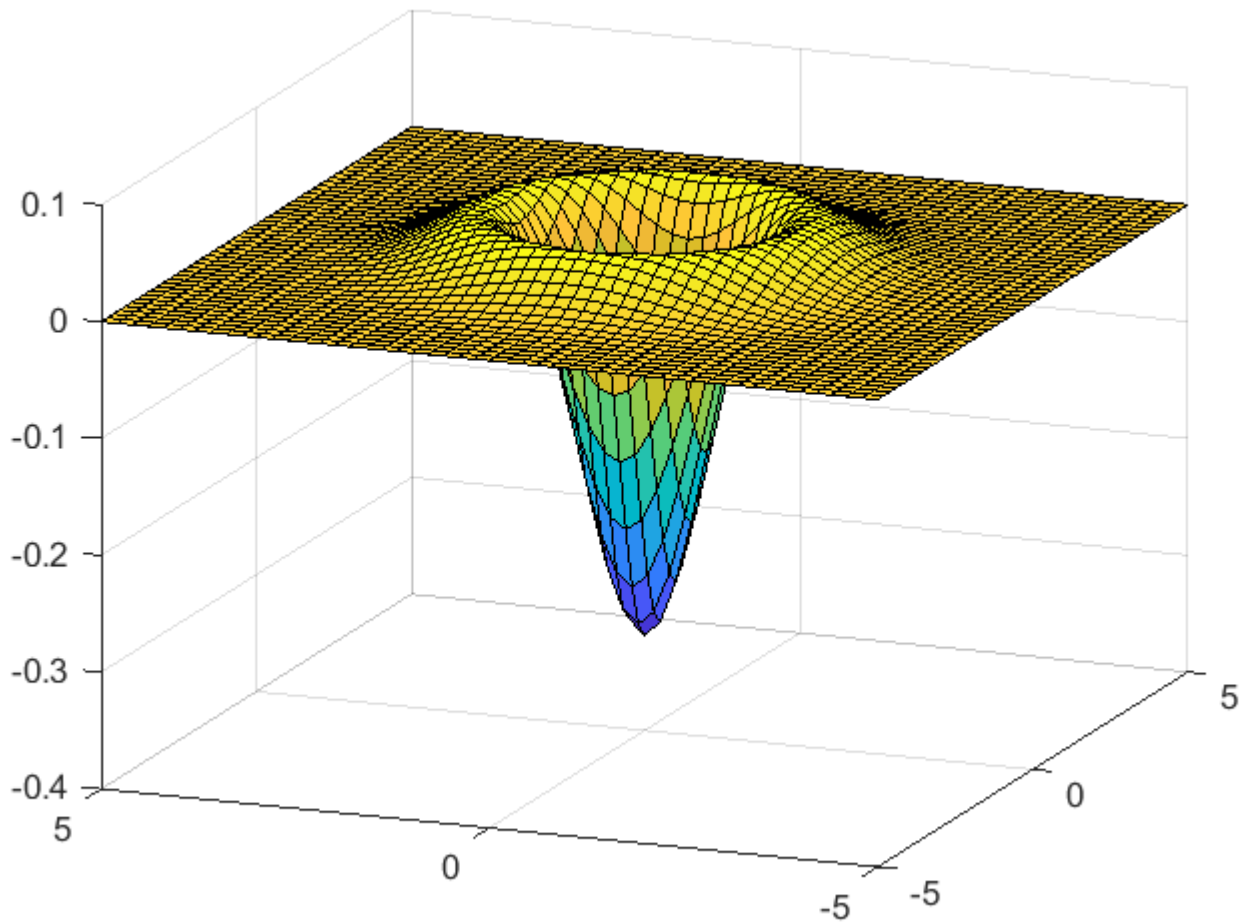
$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (5)$$

Primjenom Laplaceovog operatora nad Gausovim filtrom dobiva se filtar Meksički šešir.

Formula Meksičkog šešira $LoG(x, y)$ (engl. *Laplacian of Gaussian*) definira se kao [17]:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (6)$$

Primjer filtra Meksički šešir uz $\sigma = 1$ prikazan je na slici 9. Filtar Meksički šešir primjenjuje se za detekciju blobova.



Slika 9. Primjer filtra Meksički šešir uz $\sigma = 1$. Izrađeno alatom Matlab R2018a.

4.3. Optički tok

Optički je tok prikazuje relativno kretanje između promatrača i scene promatrajući nizove poredanih slika. Metode optičkog toka pokušavaju izračunati kretanje između dvije slike dobivene u trenucima t i $t + \Delta t$. Te se metode nazivaju diferencijalnim jer se temelje na lokalnim aproksimacijama signala slike Taylorovim redovima, tj. koriste parcijalne derivacije po prostornim i vremenskim koordinatama. Za $2D+t$ dimenzionalni slučaj piksel na lokaciji (x, y, t) s intenzitetom $I(x, y, t)$ pomaknut će se za $\Delta x, \Delta y$ i Δt između dvije slike. Otuda proizlazi ograničenje konstantosti svjetline:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (7)$$

Ako pretpostavimo da je gibanje malo, razvojem u Taylorov red dobije se:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t + \quad (8)$$

članovi višeg reda

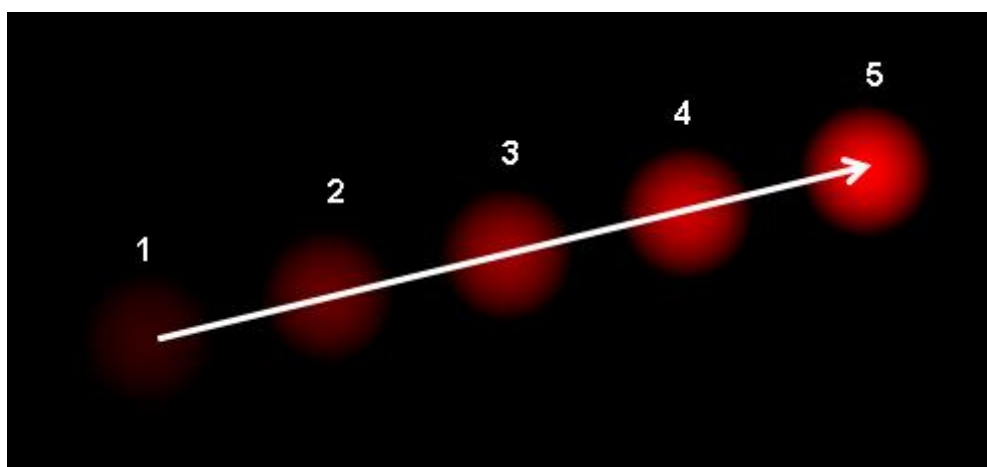
Iz prethodne dvije jednadžbe slijedi:

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0 \quad (9)$$

$$\frac{\partial I}{\partial x} \frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y} \frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t} = 0 \quad (10)$$

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0 \quad (11)$$

V_x i V_y su x i y komponente optičkog toka nad $I(x, y, t)$, a $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ i $\frac{\partial I}{\partial t}$ su derivacije slike u (x, y, t) u odgovarajućim smjerovima. (11) je jednadžba s dvije nepoznanice koja ne može biti riješena bez još jednog skupa jednadžbi koji proizlazi iz nekih dodatnih ograničenja [19].



Slika 10. Primjer vektora optičkog toka kod gibajućeg objekta. [20]

Osim ograničenja konstantnosti svjetline, još jedna pretpostavka koju algoritmi optičkog toka koriste je da susjedni pikseli imaju slično gibanje.

Algoritmi optičkog toka mogu se podijeliti na rijetke i guste. Rijetki algoritmi računaju optički tok samo za određene piksele slike, a gusti algoritmi računaju optički tok za sve piksele slike [21].

U ovom se radu za računanje optičkog toka koristi algoritam za procjenu gibanja između dvije slike na temelju polinomne aproksimacije koji spada u guste algoritme. Izvedba tog algoritma razlikuje se od temeljne ideje optičkog toka koja je dosad opisana.

4.3.1. Procjena gibanja između dvije slike na temelju polinomne aproksimacije

Algoritam je razvio Gunnar Farneback na temelju svoje doktorske disertacije u kojoj je predstavio novu transformaciju signala u kojoj signal predstavlja polinomom. Prvo je opisana ideja algoritma pa je zatim naveden detaljniji opis. Prvi korak algoritma je aproksimirati susjedstvo svakog piksela obje slike kvadratnim polinomima. Promatranjem polinomnih aproksimacija prilikom translacije između dvije slike, iz koeficijenata polinoma procjenjuje se polje pomaka.

Slijedi detaljniji opis algoritma. Aproksimira se određeno susjedstvo svakog piksela polinomom. U ovom slučaju koriste se kvadratni polinomi:

$$f(\mathbf{x}) \sim \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b} \mathbf{x} + c \quad (12)$$

pri čemu je \mathbf{A} simetrična matrica, \mathbf{b} vektor i c skalar.

Koeficijenti \mathbf{A} , \mathbf{b} i c procjenjuju se metodom težinskih najmanjih kvadrata. Težine točaka određuju se temeljem njihove pozicije u susjedstvu. Uobičajeno je da središnja točka ima najveću težinu, a težine ostalih točaka opadaju radijalno.

Ako pretpostavimo da čitav signal (sliku) možemo zapisati jednim kvadratnim polinomom i da između dvije slike postoji globalni pomak \mathbf{d} , iz polinoma

$$f_1(\mathbf{x}) = \mathbf{x}^T \mathbf{A}_1 \mathbf{x} + \mathbf{b}_1 \mathbf{x} + c_1 \quad (13)$$

$$f_2(\mathbf{x}) = f_1(\mathbf{x} - \mathbf{d}) = \mathbf{x}^T \mathbf{A}_2 \mathbf{x} + \mathbf{b}_2 \mathbf{x} + c_2 \quad (14)$$

raspisivanjem (14) i izjednačavanjem koeficijenata oba polinoma dobivamo:

$$\mathbf{A}_2 = \mathbf{A}_1 \quad (15)$$

$$\mathbf{b}_2 = \mathbf{b}_1 - 2\mathbf{A}_1 \mathbf{d} \quad (16)$$

$$c_2 = \mathbf{d}^T \mathbf{A}_1 \mathbf{d} - \mathbf{b}_1^T \mathbf{d} + c_1 \quad (17)$$

Iz (16) možemo izraziti \mathbf{d} :

$$\mathbf{d} = -\frac{1}{2}\mathbf{A}_1^{-1}(\mathbf{b}_2 - \mathbf{b}_1) \quad (18)$$

ako je \mathbf{A}_1 nesingularna matrica. Pretpostavke da čitav signal možemo zapisati jednim polinomom i da postoji globalni pomak u stvarnosti obično nisu zadovoljene. Gornje jednakosti možemo primijeniti i na stvarne signale, ali tada se pojavljuje određena razina pogreške. U stvarnim signalima, kvadratni polinom ne koristi se za transformaciju čitavog signala nego određenog dijela, tj. koeficijenti polinoma ovise o \mathbf{x} ($\mathbf{A}(\mathbf{x})$, $\mathbf{b}(\mathbf{x})$, $c(\mathbf{x})$), a također ni pomak \mathbf{d} nije globalni nego lokalni, tj. i on ovisi o \mathbf{x} ($\mathbf{d}(\mathbf{x})$).

Za realne signale više ne vrijedi

$$\mathbf{A}_2 = \mathbf{A}_1$$

pa se uvodi aproksimacija

$$\mathbf{A}(\mathbf{x}) = \frac{\mathbf{A}_1(\mathbf{x}) + \mathbf{A}_2(\mathbf{x})}{2} \quad (20)$$

i uvodi se supstitucija

$$\Delta\mathbf{b}(\mathbf{x}) = -\frac{1}{2}(\mathbf{b}_2(\mathbf{x}) - \mathbf{b}_1(\mathbf{x})) \quad (21)$$

Tako se dolazi do jednadžbe

$$\mathbf{A}(\mathbf{x})\mathbf{d}(\mathbf{x}) = \Delta\mathbf{b}(\mathbf{x}) \quad (22)$$

Traži se $\mathbf{d}(\mathbf{x})$ koji što bolje zadovoljava gornju jednadžbu za svaku točku iz susjedstva od \mathbf{x} , odnosno minimizira se vrijednost izraza

$$\|\mathbf{A}(\mathbf{x})\mathbf{d}(\mathbf{x}) - \Delta\mathbf{b}(\mathbf{x})\| \quad (23)$$

uzimajući u obzir težinu svake točke (točke udaljenije od središnje točke imaju manji utjecaj).

Ovime je objašnjena temeljna ideja Farnebäckovog algoritma za pronalaženje optičkog toka. Dodatno poboljšanje postiže se korištenjem apriorne aproksimacije polja pomaka do koje se dolazi iterativnim postupcima [22].

4.4. Algoritmi za računanje heliciteta

Glavni zadatak ovog rada je osmišljavanje algoritama za računanje heliciteta. Osmišljeno je nekoliko algoritama koji dijele neke zajedničke dijelove, a glavna razlika među njima je ta da jedni prilikom računanja heliciteta u obzir uzimaju pomak samo određenih piksela slike, a drugi u obzir uzimaju pomak svih piksela slike. Temeljna ideja za algoritme preuzeta je iz [1]. Algoritam iz tog rada sličan je prvom osmišljenom algoritmu koji je kasnije opisan, a razlika je ta što algoritam iz [1] koristi ručne oznake mikrotubula, a ovaj rad položaje mikrotubula automatski određuje računalnim metodama.

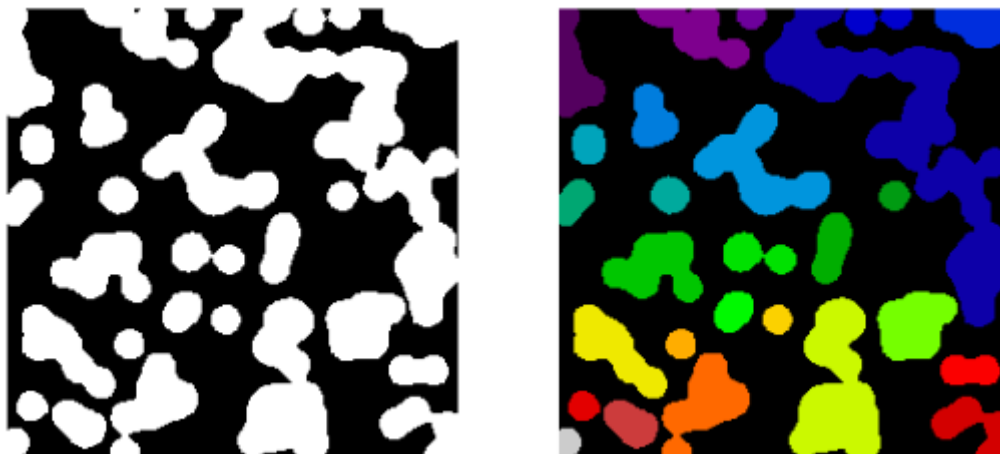
U sljedećim je algoritmima prilikom izračuna heliciteta promatrano samo 40% odrezaka stanice u središtu stanice zato što su prije dobivanja mikroskopskih slika stanice tretirane pomoću tvari koja središtu stanice daje posebnu boju kako bi se omogućilo lakše i preciznije praćenje mikrotubula.

4.4.1. Pomoćne funkcije

Svi implementirani algoritmi koriste neke zajedničke funkcije koje sudjeluju u računanju heliciteta.

Prva takva funkcija je funkcija za računanje parametara centralne osi stanice. Centralna os prolazi kroz polove stanice. Ta je os pravac u 3D prostoru. 3D pravac određen je jednom točkom i vektorom smjera. Za točku odabire se jedan od polova stanice. Vektor smjera dobije se kao vektor od jednog prema drugom polu stanice. Razlog zbog kojeg je na ovaj način potrebno računati centralnu os je zato što se polovi stanice u većini slučajeva ne nalaze točno jedan ispod drugog.

Funkcija za pronalaženje povezanih komponenti kao izlaz vraća 2D polje jednakih dimenzija kao i ulazna slika u kojem je za svaki piksel označeno pripada li pozadini ili pojedinoj povezanoj komponenti. Ako pripada pozadini, piksel je označen s 0, a ako pripada nekoj povezanoj komponenti označen je rednim brojem te povezane komponente (između 1 i ukupnog broja povezanih komponentata). Funkcija također vraća i broj pronađenih povezanih komponentata. Na slici 11 prikazan je primjer rada funkcije za pronalaženje povezanih komponentata.



Slika 11. Primjer rada funkcije za pronalaženje povezanih komponenata. [23]

Funkcija za izračun središta svake povezone komponente prolazi kroz sve piksele slike na kojoj su označene povezone komponente te za svaki piksel provjerava pripada li nekoj povezanoj komponenti. Ako pripada, ažuriraju se ukupne sume x i y koordinata piksela te povezone komponente i poveća se brojač piksela te povezone komponente. Na kraju, sume x i y koordinata podijele se ukupnim brojem piksela koji pripadaju odgovarajućoj povezanoj komponenti i tako se dobiju koordinate središta svake povezone komponente.

U implementaciji funkcije za računanje kutne komponente cilindričnog koordinatnog sustava između centralne osi i nekog piksela, prvo je potrebno izračunati preostale koordinate centralne osi na z koordinati na kojoj se nalazi zadani piksel. 3D pravac zadan u parametarskom obliku glasi:

$$x = x_0 + at \quad (24)$$

$$y = y_0 + bt \quad (25)$$

$$z = z_0 + ct \quad (26)$$

(x_0, y_0, z_0) su koordinate proizvoljne točke koja pripada pravcu, a (a, b, c) su komponente vektora smjera pravca. Iz (26) izračuna se parametar t pa se uvrsti u prve dvije jednakosti. Kut između centralne osi i nekog piksela φ računa se formulom:

$$\varphi = \text{atan2}(y_{\text{piksel}} - y_{os}, x_{\text{piksel}} - x_{os}) \quad (27)$$

Koristi se funkcija atan2 umjesto atan jer atan2 vraća rezultat između $-\pi$ i π , a atan vraća rezultat između $\frac{-\pi}{2}$ i $\frac{\pi}{2}$.

4.4.2. Algoritam centra blobova

Prvi osmišljeni algoritam računa helicitet na temelju središta blobova. Na svakom odresku stanice unutar središnjih 40% odrezaka prvo pronađe povezane komponente. Za svaku povezanu komponentu na odresku izračuna položaj njezinog središta te iz izračunatog optičkog toka uzimamo podatak o tome koliko se središnji piksel te komponente pomaknuo po x i y osima na sljedećem odresku. Zatim računamo promjenu kutne komponente cilindričnog koordinatnog sustava između početnog i pomaknutog položaja središta u odnosu na centralnu os. Taj postupak ponavljamo za svaki par uzastopnih odrezaka. Na kraju računamo prosječnu promjenu kuta dijeljenjem zbroja promjene kuta središta svih blobova ukupnim brojem blobova. Na kraju se helicitet računa kao omjer prosječne promjene kuta i razmaka između dva uzastopna odreska stanice. U ostatku rada ovaj algoritam nazivat će se algoritam centra blobova.

Algoritam 1. Algoritam centra blobova

Ulaz: odresci stanice, optički tok između odrezaka, položaji gornjeg i donjeg pola stanice, razmak između odrezaka stanice

```
centralna_os <- izračunaj parametre centralne osi(pol1, pol2)
```

```
suma_Δφ <- 0
```

```
broj_blobova <- 0
```

```
za i u 0.3*broj_odrezaka..0.7*broj_odrezaka
```

```
    označena_slika, broj_povezanih_komponenti <- nađi povezane  
        komponente(i-ti odrezak stanice)
```

```
    centri <- izračunaj centre povezanih komponenti(označena_slika,  
        broj_povezanih_komponenti)
```

```
    broj_blobova <- broj_blobova + broj_povezanih_komponenti
```

```
    za centar_komponente u centri
```

```
        Δx <- optički_tok[centar_komponente, x_os, i]
```

```
        Δy <- optički_tok[centar_komponente, y_os, i]
```

```
        Δφ <- izračunaj Δφ(centar_komponente, i, Δx, Δy, centralna os)
```

```
        suma_Δφ <- suma_Δφ + Δφ
```

```
    kraj
```

```
kraj
```

```
Δφ_prosjek <- suma_Δφ / broj_blobova
```

```
helicitet <- Δφ_prosjek / razmak_između_odrezaka
```

```
vrati helicitet
```

4.4.3. Algoritam 8-susjedstva

Osmišljen je i sličan algoritam koji se razlikuje od prethodnog u tome što u obzir ne uzima samo Δx i Δy središnjeg piksela bloba. Prvo izračuna Δx i Δy za središnji piksel i za svaki piksel njegovog 8-susjedstva pa iz izračunatih vrijednosti izračuna srednji Δx i Δy na temelju kojih računa promjenu kuta. Na taj način postiže se djelomično ispravljanje moguće greške nastale zbog pogrešno izračunatog optičkog toka u središnjem pikselu. Prilikom implementacije ovog algoritma treba paziti na slučaj kada se centar mikrotubula nalazi na rubu slike pa može doći do pristupanja elementima van slike. U stvarnosti takvi mikrotubuli ne bi trebali postojati, ali zbog šuma na slikama dolazi do pojave takvih slučajeva. Ovaj je algoritam nazvan algoritam 8-susjedstva.

4.4.4. Algoritam čitavih blobova

Razlika ovog algoritma u odnosu na prethodne je što promjenu kuta za svaki blob računa kao aritmetičku sredinu promjena kuta svakog piksela u tom blobu. Nazvan je algoritam čitavih blobova.

Algoritam 2. Algoritam čitavih blobova

Ulaz: odresci stanice, optički tok između odrezaka, položaji gornjeg i donjeg pola stanice, razmak između odrezaka stanice

```
centralna_os <- izračunaj parametre centralne osi(pol1, pol2)
```

```
suma_prosječnih_Δφ <- 0
```

```
broj_blobova <- 0
```

```
za i u 0.3*broj odrezaka..0.7*broj odrezaka
```

```
    označena_slika, broj_povezanih_komponenti <- nađi povezane  
        komponente(i-ti odrezak stanice)
```

```
    Δφ_bloba <- polje ispunjeno nulama[broj_povezanih_komponenti]
```

```
    broj_piksela_bloba <- polje ispunjeno  
        nulama[broj_povezanih_komponenti]
```

```
broj_blobova <- broj_blobova + broj_povezanih_komponenti
```

```
za piksel u svi pikseli slike
```

```
    oznaka_povezane_komponente <- označena_slika[piksel]
```

```
    ako oznaka_povezane_komponente == pozadina
```

```
        nastavi sa sljedećom iteracijom
```

```
         $\Delta x$  <- optički_tok[piksel, x_os, i]
```

```
         $\Delta y$  <- optički_tok[piksel, y_os, i]
```

```
         $\Delta\varphi$  <- izračunaj  $\Delta\varphi$ (piksel, i,  $\Delta x$ ,  $\Delta y$ , centralna os)
```

```
         $\Delta\varphi$ _bloba[oznaka_povezane_komponente] <-
```

```
             $\Delta\varphi$ [oznaka_povezane_komponente] +  $\Delta\varphi$ 
```

```
        broj_piksela_bloba[oznaka_povezane_komponente] <-
```

```
            broj_piksela_bloba[oznaka_povezane_komponente] + 1
```

```
    kraj
```

```
za j u 1..broj_povezanih_komponenti
```

```
    suma_prosječnih_ $\Delta\varphi$  <- suma_prosječnih_ $\Delta\varphi$  +  $\Delta\varphi$ _bloba[j] /
```

```
        broj_piksela_bloba[j]
```

```
    kraj
```

```
kraj
```

```
 $\Delta\varphi$ _prosjek <- suma_prosječnih_ $\Delta\varphi$  / broj_blobova
```

```
helicitet <-  $\Delta\varphi$ _prosjek / razmak_između_odrezaka
```

```
vrti helicitet
```

4.4.5. Algoritam svih piksela

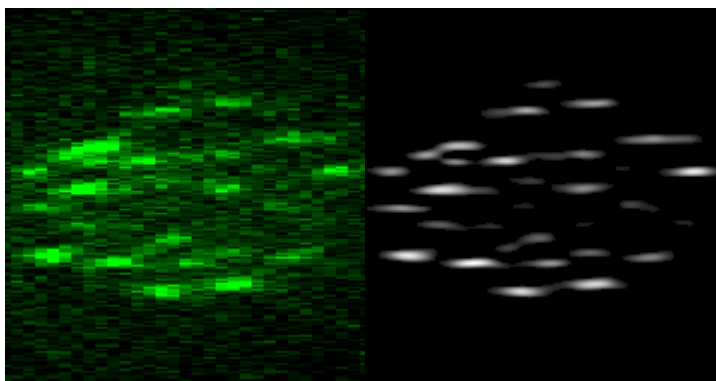
Algoritam svih piksela sličan je algoritmu čitavih blobova, ali ne koristi informaciju o pripadnosti piksela istom mikrotubulu. Ovaj algoritam računa zbroj promjena kuta svakog piksela koji ne pripada pozadini na svakom odresku pa taj broj podijeli s ukupnim brojem piksela koji ne pripadaju pozadini i na taj način dobije se prosječna promjena kuta svih piksela koji nisu pozadina. Helicitet se dobiva dijeljenjem tog broja razmakom između odrezaka.

5. Eksperimentalni rezultati

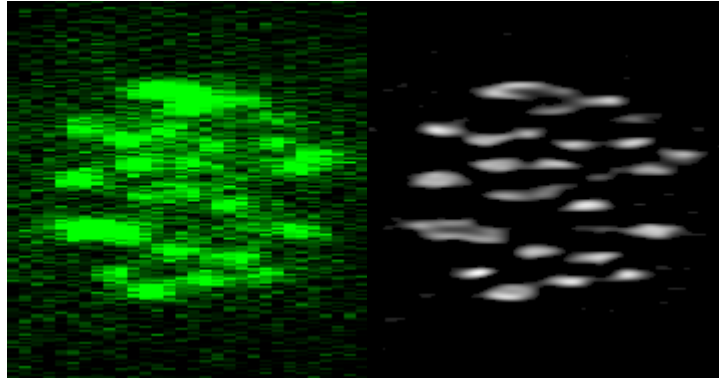
5.1. Pretprocesiranje slike

Izvorne slike stanice pune su šuma koji se eliminira primjenom filtra Meksički šešir te primjenom praga. Isprobane su razne kombinacije parametara, a ovdje je navedena kombinacija koja je dala najbolje rezultate. Primijenjen je filter Meksički šešir uz anizotropno Gaussovo zaglađivanje s parametrom σ jednakim 4 po x osi te 2 po y i z osima. Nakon toga primijenjen je prag koji uklanja sve piksele s intenzitetom manjim od 10% intenziteta piksela s najvećim intenzitetom u slici.

Na sljedećim parovima slika vidi se primjena opisanog postupka. Prvi par uzet je iz stanice na kojoj je helicitet dobro izračunat, a drugi par iz slike na kojoj je relativno loše izračunat. Uočava se visoka razina šuma prije obrade obje izvorne slike, pogotovo na drugoj. Nakon obrade, na desnim slikama dobiveni su blobovi koji odgovaraju mikrotubulima. Pažljivim promatranjem slika nakon obrade moguće je uočiti veći broj sitnih blobova, pogotovo uz rubove druge slike. Ti blobovi ne predstavljaju mikrotubule nego šum koji nismo uspjeli ukloniti. Također se primjećuje da se velik broj blobova međusobno preklapa, a trebali bi biti razdvojeni. To stvara probleme kod nekih algoritama za računanje heliciteta. Ti su problemi kasnije opisani kod odgovarajućih algoritama.



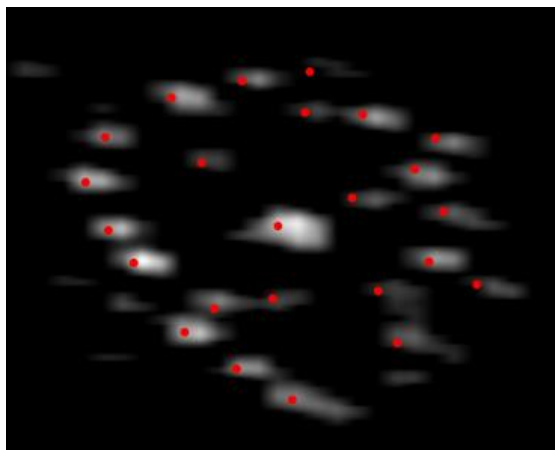
a) Odrezak iz stanice na kojoj je helicitet dobro izračunat.



b) Odrezak iz stanice na kojoj je helicitet loše izračunat.

Slika 12. Primjeri obrade izvornih slika.

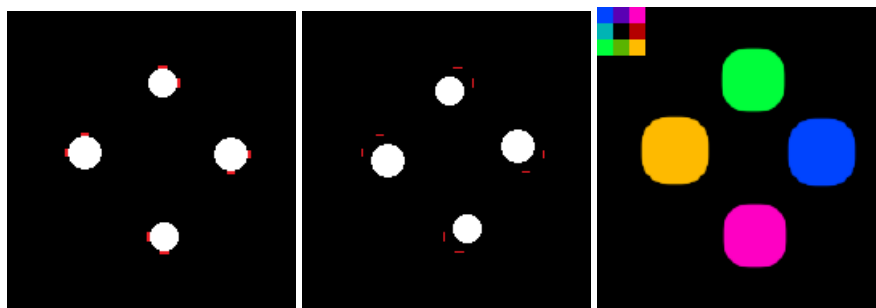
Za provjeru ispravnosti metode koja pronalazi blobove, na slikama s blobovima crvenom su bojom istaknute oznake središta mikrotubula koje su napravljene ručno na izvornim slikama. Primjer takve provjere je na slici 13. Za veliki broj mikrotubula oznake se relativno dobro preklapaju, ali za neke mikrotubule oznake se nalaze na rubu mikrotubula ili čak izvan mikrotubula. To može biti ili posljedica nepreciznosti metode pronalaženja blobova ili zbog nepreciznosti u ručnom označavanju zbog velike količine šuma. Bitno je naglasiti da se kod ručnog određivanja blobova promatrao samo otprilike svaki deseti odrezak jer se pomaci mikrotubula na uzastopnim odrescima često ne primjećuju. Algoritmi za računanje heliciteta iz ovog rada prilikom računanja nisu preskakali odreske. Isprobano je i pristup s preskakanjem odrezaka kao kod ručnog određivanja, ali je dao slabije rezultate nego bez preskakanja.



Slika 13. Provjera poklapanja dobivenih blobova s ručnim oznakama središta mikrotubula.

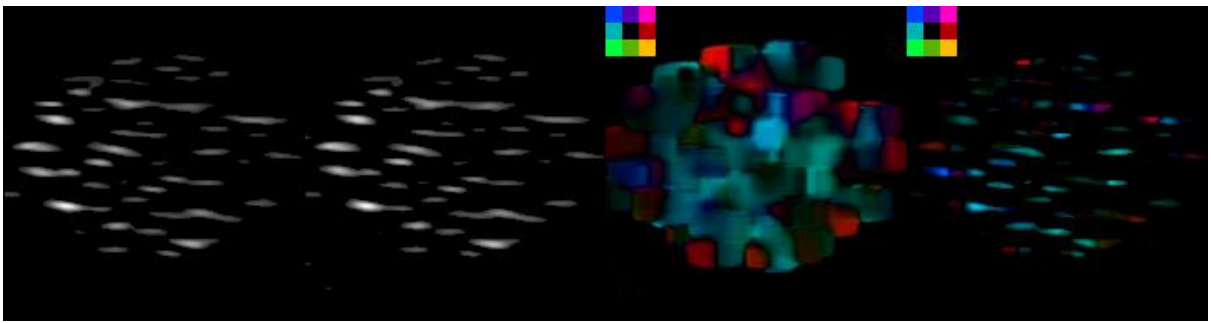
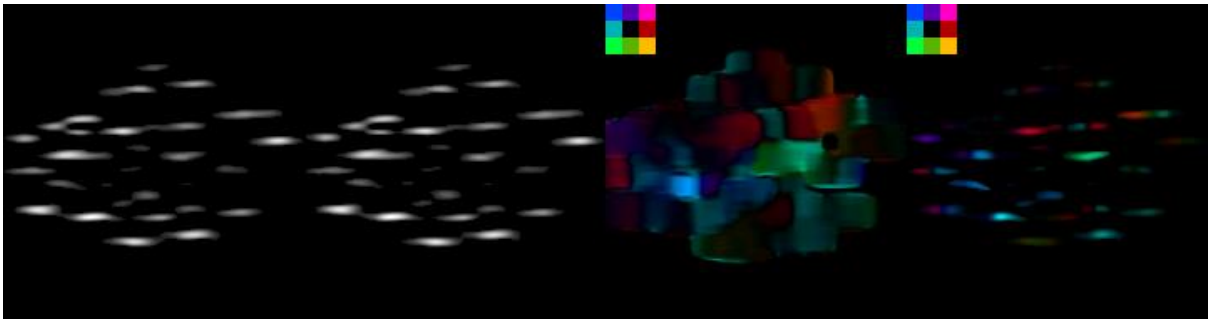
5.2. Optički tok

Sad kad imamo slike s manje šuma, možemo računati optički tok između parova slika. Da bismo se uvjerali u ispravnost računanja optičkog toka, ručno je napravljen par slika na kojem su mikrotubuli predstavljeni krugovima. Crvene linije nalaze se na istim pozicijama na obje slike, a služe da bi se uočio pomak krugova. Nad izračunatim tokom primijenjeno je kodiranje bojama zbog vizualizacije pomaka. Legenda se nalazi u gornjem lijevom kutu, a svaka boja označava smjer pomaka. Iz sljedećih se slika vidi da metoda za računanje toka dobro funkcionira. Funkcionalnost optičkog toka provjerena je i numerički tako da se uspoređivao stvarni pomak u pikselima i pomak kojeg je odredio optički tok, rezultati su ispali identični.

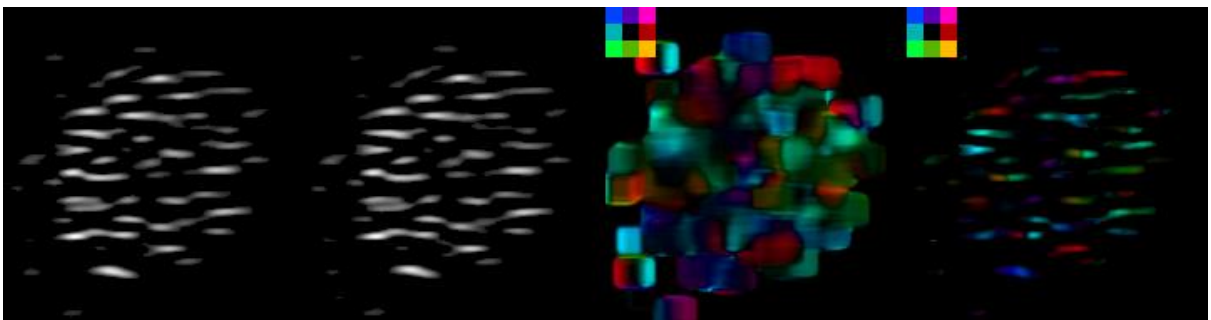
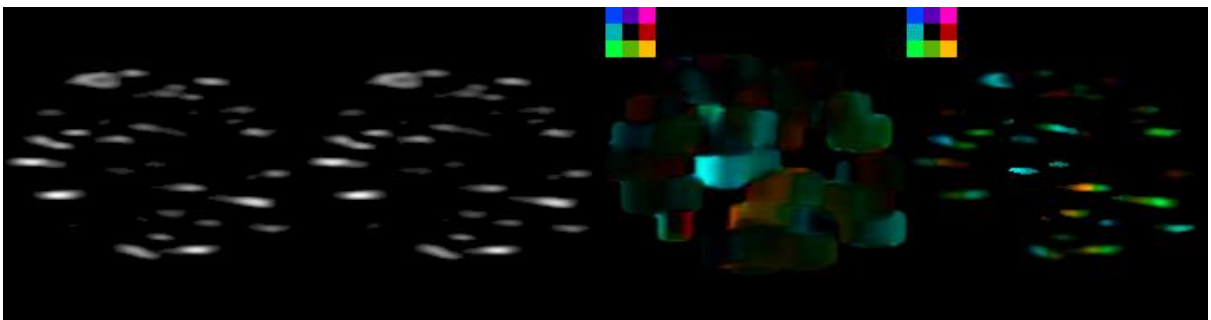


Slika 14. Vizualizacija optičkog toka za pomak ručno nacrtanih mikrotubula.

Rezultati nad stvarnim slikama mikrotubula razlikuju se od gornjeg jednostavnog slučaja. Kod računanja optičkog toka pretpostavlja se da susjedni pikseli imaju slično gibanje pa se zato kod vizualizacije toka vidi da je metoda računanja toka i pikselima izvan ruba mikrotubula pridijelila određeni pomak. To se vidi i na gornjem primjeru. Bitno je uočiti da algoritam čitavih blobova i algoritam svih piksela neće u obzir uzeti pomake koje je metoda za računanje toka pridijelila pikselima izvan mikrotubula. Da bi se vizualiziralo koji pikseli ulaze u proračun kod navedenih algoritama, napravljena je operacija logičko I između slike s mikrotubulima i slike kojom se vizualizira tok. Time su iz vizualizacije toka uklonjeni svi pikseli koji ne pripadaju mikrotubulima i ne uzimaju se u obzir kod algoritma svih piksela. Na sljedećim primjerima prikazano je: početna i konačna slika za računanje toka, vizualizacija izračunatog optičkog toka i vizualizacija toka nakon primjene logičkog I s početnom slikom za računanje toka.



a) Primjeri odrezaka stanica na kojima je helicitet dobro izračunat.



b) Primjeri odrezaka stanica na kojima je helicitet loše izračunat.

Slika 15. Vizualizacija optičkog toka za pomak mikrotubula.

Na svim prikazanim primjerima uočava se da pojedini mikrotubuli imaju istovremeno kretanje u više smjerova. Prepostavka je da do toga dolazi zbog širenja i skupljanja blobova koji predstavljaju mikrotubule. Budući da se svaki mikrotubul mora prvo pojaviti

pa kasnije nestati, pretpostavlja se da se prilikom računanja heliciteta pomaci uzrokovani širenjem i skupljanjem međusobno poništavaju.

5.3. Helicitet

Helicitet je izračunat pomoću četiri prethodno opisana algoritma. Izračuni su provedeni nad nekoliko skupina stanica. Stanice su grupirane prema načinu i sredstvu tretiranja prije snimanja mikroskopom. Ukupno je 5 skupina stanica. Nazivi skupina odgovaraju tretmanu koji su stanice primile (osim 4. skupine jer sadržava stanice koje dolaze iz druge faze diobe – prometafaze, pa je sukladno tomu imenovana):

1. HeLa PRC1-GFP LGN(KD)
2. HeLa PRC1-GFP LGN(control)
3. HeLa Anti-PRC1
4. Prometafazne
5. HeLa PRC1-GFP live

U sljedećoj tablici, za svaku skupinu stanica dani su srednja vrijednost i standardna devijacija heliciteta svih stanica koje ta skupina sadrži. U lijevom stupcu nalaze se vrijednosti koje daju algoritmi iz ovog rada, a u desnom stupcu nalaze se heliciteti izračunati algoritmom iz [1] koristeći ručne oznake. Broj stanica u svakoj od skupina redom iznosi: 10, 5, 5, 10 i 10.

Redni broj skupine stanica	Heliciteti dobiveni algoritmima ovog rada [°/μm]	Heliciteti dobiveni algoritmom iz [1] [°/μm]
1.	Algoritam centra blobova: -1.53 ± 1.53 Algoritam 8-susjedstva: -1.52 ± 1.52 Algoritam čitavih blobova: -1.49 ± 1.43 Algoritam svih piksela: -1.57 ± 1.18	-2.14 ± 1.36
2.	Algoritam centra blobova: -1.00 ± 0.68 Algoritam 8-susjedstva: -1.00 ± 0.66 Algoritam čitavih blobova: -0.71 ± 0.75 Algoritam svih piksela: -1.14 ± 0.84	-2.88 ± 1.69
3.	Algoritam centra blobova: -0.26 ± 0.87 Algoritam 8-susjedstva: -0.26 ± 0.87 Algoritam čitavih blobova: -0.35 ± 0.84 Algoritam svih piksela: -0.69 ± 1.29	-1.83 ± 1.53
4.	Algoritam centra blobova: 0.43 ± 0.75 Algoritam 8-susjedstva: 0.42 ± 0.74 Algoritam čitavih blobova: 0.33 ± 0.62 Algoritam svih piksela: 0.17 ± 0.75	0.09 ± 0.67
5.	Algoritam centra blobova: -1.65 ± 1.24 Algoritam 8-susjedstva: -1.66 ± 1.23 Algoritam čitavih blobova: -1.60 ± 0.96 Algoritam svih piksela: -1.67 ± 0.97	-2.49 ± 1.19

Tablica 1. Usporedba heliciteta izračunatih algoritmima iz rada s helicitetom izračunatim pomoću algoritma iz [1].

Usporedbom s rezultatima dobivenim algoritmom iz [1], uočava se da nad svakom skupinom stanica najbolje rezultate daje algoritam svih piksela. Najbolji rezultati dobiveni su nad prvim skupom stanica. U sljedećoj tablici uspoređuju se rezultati koje daje algoritam svih piksela nad stanicama prve skupine s rezultatima algoritma iz [1]. Dobro poklapanje rezultata vidljivo je na stanicama 1, 3, 5, 6 i 7.

Stanica	Helicitet algoritmom svih piksela [$^{\circ}/\mu\text{m}$]	Helicitet algoritmom iz [1] [$^{\circ}/\mu\text{m}$]
1.	0.20	0.23
2.	-1.81	-3.91
3.	-4.33	-4.01
4.	-0.76	-0.51
5.	-2.35	-2.34
6.	-1.52	-1.78
7.	-1.93	-1.83
8.	-0.46	-2.64
9.	-1.76	-3.03
10.	-0.96	-1.53

Tablica 2. Usporedba rezultata koje daje algoritam svih piksela nad stanicama prve skupine s rezultatima algoritma iz [1].

Na slici 15 prikazane su dvije stanice na kojima je helicitet dobro izračunat i dvije na kojima je loše izračunat. Postoji više mogućih razloga zašto neki algoritmi daju slabije rezultate i zašto su rezultati nad određenim stanicama slabiji. Kod algoritama koji rade nad pojedinim blobovima (svi algoritmi osim algoritma svih piksela), prvi problem je taj što na slikama nije uklonjen sav šum. Zbog toga se ostatci šuma također prepoznaju kao mikrotubuli te što je više takvih lažnih mikrotubula, to je veći njihov utjecaj na rezultat. Drugi velik problem s istim algoritmima je spajanje mikrotubula nakon obrade izvornih slika. Često se događa da se na obrađenim slikama dva ili tri mikrotubula djelomično preklapaju te zbog toga metoda za pronalaženje povezanih komponenti smatra da preklapljeni mikrotubuli zapravo čine jedan mikrotubul.

Algoritam svih piksela ima manje problema sa šumom jer na šum obično otpada mnogo manji broj piksela nego na mikrotubule. Ni preklapanje mikrotubula ne stvara probleme ovom algoritmu jer on u obzir uzima kretanje svakog pojedinog piksela, dakle ne koristi informaciju o povezanosti piksela u mikrotubule. Moguće probleme kod algoritma svih piksela stvaraju različite veličine mikrotubula na pojedinim slikama. Ako na nekom odresku određenom mikrotubulu pripada više piksela nego nekom drugom mikrotubulu

na tom odresku, tada mikrotubul s većim brojem piksela ima veći utjecaj u računanju heliciteta.

Na laptopu sa CPU i5-4210 frekvencije 1.7 GHz, prosječno vrijeme potrebno za obradu izvorne stanice i računanje heliciteta iznosi 27s, što predstavlja značajno poboljšanje u odnosu na vrijeme potrebno za ručno označavanje mikrotubula.

6. Zaključak

U ovom radu razmatra se računanje heliciteta mikrotubula bez potrebe za ručnim oznakama. Slike dobivene konfokalnom mikroskopijom sadrže visoku razinu šuma pa ih je prije računanja heliciteta bilo potrebno prikladno obraditi. Primjenom filtra Meksički šešir na slikama su istaknuti mikrotubuli te je smanjena količina šuma. Pomaci pojedinih piksela između dva odreska stanice izračunati su optičkim tokom. Osmišljena su četiri algoritma za računanje heliciteta. Prvi osmišljeni algoritam pronalazi pojedine mikrotubule na svakom odresku te računa helicitet na temelju promjene kuta središta pojedinih mikrotubula u odnosu na centralnu os diobenog vretena. Drugi algoritam sličan je prvom, uz razliku da u obzir uzima prosječni pomak središnjeg piksela mikrotubula i svih njemu susjednih piksela (njegovo 8-susjedstvo). Prva dva algoritma daju vrlo slične rezultate što je očekivano jer je metoda računanja gotovo ista. Treći algoritam računa promjenu kuta svakog piksela u nekom mikrotubulu pa iz toga računa promjenu kuta pojedinog mikrotubula između uzastopnih odrezaka stanice. Na temelju tih promjena računa helicitet. Posljednji, četvrti algoritam ne uzima u obzir informaciju o povezanosti piksela pojedinog mikrotubula nego računa ukupni helicitet na temelju prosječne promjene kuta svakog piksela u odresku.

Kvaliteta rezultata razlikuje se od stanice do stanice. Na stanicama s većom količinom šuma uglavnom su postignuti slabiji rezultati. Uspoređujući algoritme, najbolji su rezultati postignuti četvrtim algoritmom. To je bilo očekivano jer prva tri algoritma koriste informaciju o pripadnosti pojedinih piksela mikrotubulima, a uz visoku razinu šuma teško je precizno odrediti pojedine mikrotubule. Dva su glavna problema s prva tri algoritma. Prvi problem su lažni mikrotubuli čija prisutnost je posljedica šuma koji nije uklonjen u potpunosti. Drugi problem predstavlja to što se nakon obrade izvornih slika Meksičkim šeširom neki mikrotubuli preklapaju pa se ne može odrediti promjena kuta za svaki pojedini mikrotubul. Problem s četvrtim algoritmom je to što mikrotubuli s većim brojem piksela imaju veći utjecaj na izračunati helicitet.

Rezultati bi vjerojatno bili bolji uz bolju kvalitetu izvornih slika. Način obrade kojim bi se možda mogle poboljšati izvorne slike je primjena postupka dekonvolucije. Kod dekonvolucije pretpostavlja se da je slika koju daje konfokalni mikroskop rezultat

konvolucije stvarne slike i PSF. PSF (engl. *point spread function*) je karakteristika mikroskopa koja se određuje eksperimentalno.

Literatura

- [1] Z. Boban, Model diobenog vretena određen momentima sila i silama, diplomski rad, Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet, Fizički odsjek (2017)
- [2] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, P. Walter, Molecular Biology of the Cell, 6th edition, New York, Garland Science (2014)
- [3] G. M. Cooper, The Cell: A molecular approach, 2nd edition, Boston University Sinauer Associates (2000)
- [4] N. Pavin, I.M. Tolić, Self-Organization and Forces in the Mitotic Spindle, Annu. Rev. Biophys. 45 (2016) 279-298.
- [5] Novak et al., bioRxiv, doi: 10.1101/167437 (2017)
- [6] Grupa Tolić, IRB
- [7] B. Goodman, Y. Zheng, Mitotic spindle morphogenesis: Ran on the microtubule cytoskeleton and beyond, Biochem Soc Trans 34 (2006) 716-721
- [8] A. Ettinger, T. Wittmann, Fluorescence Live Cell Imaging, Methods Cell Biol 123 (2014), 77-94
- [9] D. Semwogerere, E. R. Weeks, Confocal microscopy, Encyclopedia of Biomaterials and Biomedical Engineering, (2005)
- [10] I. Weber, Glasilo IRB (2004)
- [11] Gaussian blur with sigma = 1, URL:
http://fiveko.com/assets/pics/math/gauss1d_shape.jpg
- [12] Gaussian 2D, URL: https://en.wikipedia.org/wiki/File:Gaussian_2d.svg
- [13] Gaussian blur, URL: https://en.wikipedia.org/wiki/Gaussian_blur
- [14] Discrete approximation to Gaussian function with sigma = 1, URL:
<https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>
- [15] The Gaussian kernel, STAT 692 Medical Image Analysis, University of Wisconsin-Madison, Department of Statistics, URL:

<http://www.stat.wisc.edu/~mchung/teaching/MIA/reading/diffusion.gaussian.kernel.pdf>.pdf

[16] Spatial convolution, CS 178 - Digital Photography, Stanford University, Stanford Computer Graphics Laboratory, URL:

<https://graphics.stanford.edu/courses/cs178/applets/convolution.html>

[17] Laplacian/Laplacian of Gaussian, University of Edinburgh, School of Informatics, URL:

<https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>

[19] Optical flow, Wikipedia, URL: https://en.wikipedia.org/wiki/Optical_flow

[20] Optical flow example, URL:

https://upload.wikimedia.org/wikipedia/en/1/10/Optical_flow_example_v2.png

[21] Optical flow, Dense Optical Flow in OpenCV, URL:

https://docs.opencv.org/3.3.1/d7/d8b/tutorial_py_lucas_kanade.html

[22] G. Farnebäck, Two-Frame Motion Estimation Based on Polynomial Expansion,

Linköping University, Computer Vision Laboratory, URL: <http://www.diva-portal.org/smash/get/diva2:273847/FULLTEXT01.pdf>

[23] Labelling connected components of an image, URL: https://www.scipy-lectures.org/packages/scikit-image/auto_examples/plot_labels.htm

[24] Confocal microscopy, URL: https://www.jic.ac.uk/microscopy/more/T5_8.htm

Sažetak

Ovaj rad bavi se računanjem heliciteta mikrotubula bez potrebe za ručnim označavanjem mikrotubula. Helicitet je faktor uvrtnja mikrotubula oko glavne osi diobenog vretena u procesu stanične diobe. Helicitet mikrotubula daje uvid u sile koje sudjeluju u trenutno nepoznatim detaljima stanične diobe. U ovom radu, osmišljena su četiri algoritma za računanje heliciteta. Algoritmi se temelje na algoritmu za računanje heliciteta iz ručnih oznaka mikrotubula iz diplomskog rada Zvonimira Bobana. Izazov predstavlja uklanjanje šuma te isticanje mikrotubula. Za računanje pomaka mikrotubula koristi se gusti optički tok. Osmišljeni algoritmi testiraju se na stanicama dobivenim iz Grupe Tolić s Instituta Ruđer Bošković. Prikazana je usporedba rezultata koje daju algoritmi iz ovog rada s rezultatima algoritma iz diplomskog rada Zvonimira Bobana. Kvaliteta rezultata razlikuje se od stanice do stanice. Postoji još mnogo mjesta za napredak.

Ključne riječi: računalni vid, stanična dioba, mikrotubuli, helicitet, obrada slike, optički tok

Abstract

This paper deals with calculating microtubules' helicity without manual labeling of the microtubules. Helicity is factor of torsion of a microtubule around the central axis of the mitotic spindle in the process of cell division. Microtubules' helicity provides insight in currently unknown details of cell division. Four algorithms for calculating helicity are developed in this paper. The algorithms are based on the algorithm for calculating helicity from manual labels of microtubules from Zvonimir Boban's master thesis. Removal of noise and extraction of microtubules present a challenge. Dense optical flow is used for calculating microtubules' displacement. Developed algorithms are tested on cells provided by the Tolić Group from the Institute Ruđer Bošković. The paper compares the results calculated with developed algorithms with the results calculated with the algorithm from Zvonimir Boban's master thesis. There is still a lot of room for improvement.

Keywords: computer vision, cell division, microtubules, helicity, image processing, optical flow