

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

## DIPLOMOVÁ PRÁCA

Študijný odbor: **Informačné systémy - Spracovanie dát**

**Bc. Júlia Tomášiková**

**Spracovanie a analýza videosekvencií z biologických experimentov  
pomocou špeciálnych algoritmov pre detekciu a trasovanie objektov**

Vedúci: **doc. Mgr. Ivan Cimrák, Dr.**

Reg.č. 152/2016

Máj 2017

### **Prehlásenie**

Prehlasujem, že som túto prácu napísala samostatne pod odborným vedením vedúceho diplomovej práce a že som uviedla všetky použité pramene a literatúru, z ktorých som čerpala.

V Žiline, dňa 26.4.2017

Bc. Júlia Tomášiková

## **PodĎakovanie**

Ďakujem vedúcemu diplomovej práce doc. Mgr. Ivanovi Cimrákovi Dr., za jeho hodnotné rady, odbornú pomoc a usmernenie pri vypracovávaní tejto diplomovej práce. Ďakujem aj Mgr. Ivete Jančigovej PhD. za konzultácie a poskytnuté rady.

Osobitné poďakovanie patrí mojej rodine za podporu počas celého štúdia.

## Abstrakt

BC. JÚLIA TOMÁŠIKOVÁ: *Spracovanie a analýza videosekvencií z biologických experimentov pomocou špeciálnych algoritmov pre detekciu a trasovanie objektov* [Diplomová práca]

Žilinská Univerzita v Žiline, Fakulta riadenia a informatiky, Katedra softvérových technológií.

Vedúci: doc. Mgr. Ivan Cimrák, Dr.

Stupeň odbornej kvalifikácie: inžinier v odbore Informačné systémy - Spracovanie dát, FRI ŽU v Žiline, 2017 – 63 s.

Cieľom diplomovej práce bolo spracovanie a analýza videosekvencií z biologických experimentov pomocou algoritmov pre detekciu a trasovanie objektov. Úvod práce sa zaoberá popisom problematiky trasovania objektov. Obsahom práce bolo vytvorenie nástroja, ktorý by umožňoval trasovanie objektov. Nástroj zaznamenával polohu a počítal aj rýchlosť sledovaných objektov. Výsledky zo spracovaných videosekvencií sa následne porovnali s výsledkami zo simulačného modelu biologického experimentu. Záver práce obsahuje porovnanie výsledkov z videozáznamu a zo simulácie.

**Kľúčové slová:** detekcia objektov, trasovanie objektov, spracovanie videozáznamu, modelovanie, simulácia, Object-in-fluid

## **Abstract**

BC. JÚLIA TOMÁŠIKOVÁ *Processing and analysis of videosequences from biological experiments using precial detection and tracking algorithms*. [Diploma thesis]

University of Žilina, Faculty of Management Science and Informatics, Department of Software Technologies.

Tutor: doc. Mgr. Ivan Cimrák, Dr.

Qualification level: master in field of Information systems, FRI ŽU v Žiline, 2017 – 63 p.

The aim of the diploma thesis was processing and analysis of video sequences from biological experiments using algorithms for object detection and tracing. The introduction of thesis deals with the description of the object tracing. The content of the thesis was to create a tool that would allow tracing objects. The tool recorded the position and also calculated the velocity of the tracked objects. The results from the processed video sequences were then compared with the results from the simulation model of the biological experiment. The conclusion of the thesis contains the comparison of the results from the video sequences and the simulation.

**Key words:** object detection, object tracking, video processing, modeling, simulation, Object-in-fluid

# Obsah

<b>Úvod</b>	<b>10</b>
<b>1 Trasovanie objektov</b>	<b>13</b>
1.1 Detekcia objektov . . . . .	14
<b>2 Trasovanie červených krviniek</b>	<b>18</b>
2.1 Dostupné nástroje . . . . .	19
2.2 Vývoj vlastného nástroja . . . . .	24
<b>3 Implementácia</b>	<b>26</b>
3.1 Reprezentácia detegovaných objektov . . . . .	27
3.2 Vytváranie trás . . . . .	28
3.3 Vytvorenie matice toku . . . . .	32
3.4 Spájanie trás . . . . .	35
3.5 Pripájanie nezaradených bodov . . . . .	38
3.6 Výstup . . . . .	40
3.7 Metódy pre zobrazenie . . . . .	40
3.8 Možné budúce implementácie . . . . .	42
<b>4 Simulácia experimentu</b>	<b>43</b>
4.1 Popis modelu . . . . .	44
4.2 Simulácia experimentu . . . . .	45
4.3 Výsledky simulácie . . . . .	46

<b>5</b>	<b>Vyhodnotenie výsledkov</b>	<b>48</b>
5.1	Spôsob porovnania . . . . .	48
5.2	Vyhodnotenie dát z video záznamu . . . . .	51
5.3	Vyhodnotenie dát zo simulácie . . . . .	53
5.4	Vyhodnotenie . . . . .	55
5.5	Testovanie . . . . .	55
	<b>Záver</b>	<b>59</b>
	<b>Literatúra</b>	<b>61</b>
<b>A</b>	<b>Obsah priloženého CD</b>	<b>63</b>

## Zoznam obrázkov

1.1	Pôvodná snímka . . . . .	15
1.2	Pozadie, ktoré bude zo snímky odstránené . . . . .	15
1.3	Snímka po odstránení pozadia . . . . .	16
1.4	Houghova transformácia . . . . .	16
1.5	Detekcia stredov . . . . .	17
1.6	Detegované stredy červených krviniek . . . . .	17
2.1	Ukážka mikrokanálu . . . . .	18
2.2	Obrázok kanálu . . . . .	20
2.3	Získané trasy pred nastavením penalizácie . . . . .	21
2.4	Získané trasy . . . . .	22
2.5	Detekcia a trasovanie pomocou nástroja TrackMate . . . . .	23
3.1	Okolie s jedným bodom . . . . .	28
3.2	Okolie s dvoma bodmi . . . . .	29
3.3	Nové okolie . . . . .	30
3.4	Vektory a body . . . . .	31
3.5	Prehľadávanie okolia . . . . .	34
3.6	Spájanie trás: vektor . . . . .	36
3.7	Uhly medzi vektormi a ich smer . . . . .	36
3.8	Spájanie trás: očakávaný bod . . . . .	37
3.9	Zobrazenie bodov . . . . .	41
3.10	Zobrazenie trás . . . . .	41



4.1	Elastický objekt . . . . .	44
4.2	Model kanálu . . . . .	45
4.3	Prechod krvinky koncom kanála . . . . .	46
4.4	Prechod krvinky začiatkom kanála . . . . .	47
5.1	Oblasti . . . . .	49
5.2	Výsledky pre prvú oblasť videozáznamu . . . . .	51
5.3	Výsledky pre druhú oblasť videozáznamu . . . . .	52
5.4	Výsledky pre tretiu oblasť videozáznamu . . . . .	52
5.5	Výsledky pre prvú oblasť simulácie . . . . .	53
5.6	Výsledky pre druhú oblasť simulácie . . . . .	54
5.7	Výsledky pre tretiu oblasť simulácie . . . . .	54
5.8	Výsledné trasy pre iný typ kanálu . . . . .	56
5.9	Získané trasy zo súboru $S_1$ . . . . .	57
5.10	Získané trasy zo súboru $S_2$ . . . . .	58

# Úvod

V súčasnosti existuje silné prepojenie výpočtovej techniky s inými vednými odbormi. Počítačová simulácia a modelovanie sa využíva čoraz častejšie. Podstatou modelovania je náhrada skúmaného systému (originálu) jeho modelujúcim systémom (modelom). Cieľom tejto metódy je získať pomocou experimentov s modelom informácie o origináli. Pre využitie modelu namiesto originálu môže existovať viac dôvodov, napríklad časová náročnosť, nedostupnosť skúmaného systému, vysoké náklady na experimentovanie so skúmaným systémom a ďalšie. Simulácia je výskumná technika, ktorej podstata je náhrada skúmaného dynamického systému (originál) jeho simulátorom. Zo simulátora sa pomocou experimentov získajú informácie o pôvodnom skúmanom dynamickom systéme. Modelovanie a simulácia majú svoje uplatnenie v oblasti chémie, fyziky, biológie, ale aj medicíny. [18]

Modelovaniu pre oblasť biomedicíny sa venuje výskumná skupina cell-in-fluid zo Žilinskej univerzity v Žiline. Skupina sa zaoberá projektom mikrofluidického čipu pre izoláciu rakovinových buniek zo vzorky krvi. Úlohou čipu je izolácia cirkulujúcich rakovinových buniek zo vzorky krvi pacienta postihnutého rakovinou. Mikrofluidický čip funguje na jednoduchom princípe. Zariadenie čipu je vybavené úzkymi kanálkami, ktoré majú pravidelne rozmiestnené výbežky (stĺpiky, prekážky). Tie sú zhotovené z magneticky aktívnych polymérov. Kanálkami prúdi krv, ktorá obsahuje rakovinové bunky. Vo väčšine prípadov sú rakovinové bunky väčšie a tuhšie ako ostatné zložky krvi a zachytia sa na výbežkoch. Zariadenie funguje ako miniatúrna pasca, v ktorej sa rakovinová bunka zachytí avšak zdravé bunky prejdú. Práve vďaka použitiu magneticky aktívnych polymérov je možné upravovať veľkosť kanálikov a rozmiestnenie stĺpikov (ich geometriu) a dosiahnuť tak vyššiu účinnosť zariadenia pre špecifické druhy rakoviny. Geometria stĺpikov je viazaná na hlavnú funkciu čipu: napr. miešanie

buniek, ich oddelovanie, zachytávanie. Čipy by mohli v blízkej budúcnosti nahradiť drahé a zložité laboratórne testy. Analýza buniek pomocou mikrofluidických čipov tak otvára nové možnosti pre diagnostiku a personalizovanú liečbu pacientov.

Skupina cell-in-fluid sa zaoberá modelovaním toku krvi na úrovni buniek. Model má dve hlavné časti: tekutinu a elastický objekt. Tekutina je definovaná ako fiktívne body, ktoré sú uložené v pevnej mriežke. Elastický objekt (reprezentuje červenú krvinku) je modelovaný pomocou povrchovej triangulácie. Pod pojmom povrchová triangulácia si predstavíme trojuholníkovú sieť, ktorá pokrýva povrch objektu. Body tvoriace sieť predstavujú hmotné body. Body tvoriace trojuholník sú navzájom prepojené pružinami. Pružiny predstavujú elastické sily. Obe časti spolu interagujú a pôsobia na seba. Model toku krvi je použitý spolu s modelom mikrofluidického čipu a slúži na optimalizáciu zariadenia a zvýšenie jeho účinnosti. Na modeli toku krvi sa stále pracuje. Po validácii modelu, by mohol byť použitý napr. pre rýchle predpovedanie polohy objektu (červenej krvinky) v toku, rýchlosť objektu či získanie informácie o membránovej deformácii jednotlivých buniek v toku. Tieto informácie by sa bez validného modelu získavali iba pomocou drahých a časovo náročných analýz experimentov.[16][17]

Cieľom práce bolo spracovanie a analýza videozáznamov biologických experimentov pomocou špeciálnych algoritmov pre detekciu a trasovanie objektov za účelom získania informácií o polohe a rýchlosti buniek. Spracovanie videozáznamu reálneho experimentu pomocou nami vytvorených metód a porovnanie s dátami získanými zo simulácie s modelom toku krvi, by mohlo pomôcť pri validácii modelu. Rovnako by informácie získané zo spracovaného videozáznamu experimentu, mohli uľahčiť prácu a poskytnúť dodatočné informácie výskumníkom v oblasti medicíny.

V úvode práce sa budeme venovať detekcii červených krviniek, ktorá predchádza procesom trasovania. Budú popísané problémy špecifické pre spracovanie biologických záznamov, aj stručný priebeh detekcie červených krviniek. Keďže detekcia nepatrí medzi ciele práce, je uvedená len pre referenciu. V nasledujúcich kapitolách uvedieme možnosti trasovania objektov, aj príklad použitia konkrétneho nástroja pre trasovanie. Okrem toho predstavíme návrh vlastného nástroja, ktorého implementácia bude podrobne popísaná a vysvetlená. Následne nástroj použijeme pri spracovaní videozáznamov biologických experimentov a výstupy podrobíme meraniam a hodnoteniam. Pre porovnanie nám budú slúžiť výstupy zo simulácie experimentu.

Na základe porovnávacieho kritéria je potrebné zistiť či model toku krvi zodpovedá biologickému toku z videozáznamu.

# Kapitola 1

## Trasovanie objektov

V oblasti počítačového videnia sa v posledných rokoch výskum zameriaval na vývoj efektívnych postupov a algoritmov pre sledovanie pohybu objektov (trasovanie). Využitie je široké od oblasti systémov pre sledovanie dopravy, v armáde pre navádzanie striel, navádzanie satelitov alebo sledovanie mikroorganizmov pod mikroskopom. Tieto systémy sa snažia prebrať väčšiu časť spracovania a vyhodnocovania dát. Z tohto dôvodu je kladený veľký dôraz na vývoj efektívnych algoritmov pre sledovanie pohybu.

Existuje veľké množstvo spôsobov sledovania pohybu, ktoré sa navzájom od seba líšia vlastnosťami sledovaných obrazov, spôsobom snímania snímok, cieľom sledovania pohybu a ďalšími aspektami. Tieto techniky používajú apriórne znalosti pri riešení konkrétnych úloh. Informácie ako predpokladaný tvar, veľkosť, priestorové usporiadanie objektov či spôsob správania sa objektov, slúžia pri spracovaní obrazu ako obmedzujúce podmienky, prípadne sa použijú pri optimalizácii. [1]

Trasovanie objektu môže prebiehať real-time (pohyb objektu je zaznamenávaný a vyhodnocovaný okamžite pri snímaní kamerou) alebo vo forme post-processingu, kedy sa spracúva videozáznam. V diplomovej práci sme sa zaoberali práve post-processingom záznamov biologických experimentov.

V oblasti spracovania videa sa veľmi často používajú pojmy, ktoré môžu byť pre čitateľa, ktorý nie je zbehlý v danej oblasti nejasné. Pre lepšie porozumenie pri čítaní nasledujúceho textu ozrejmieme niektoré pojmy. Video je sekvenca obrázkov, snímok idúcich po sebe. Pri

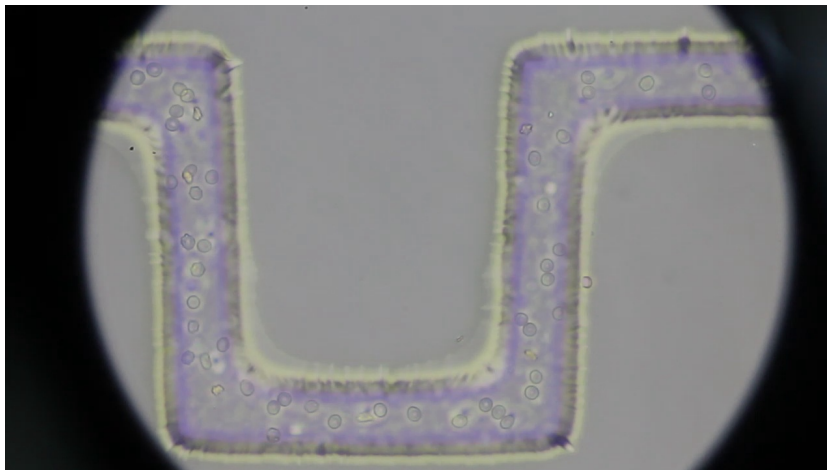
videozázname je dôležité poznať jeho obrazovú frekvenciu. Je to počet celých jednotlivých snímkov zaznamenaných, zobrazených alebo prenesených za sekundu. Jednotkou tejto veličiny sú snímky za sekundu (ang. frames per second), skr. fps. Obrazová frekvencia býva uvedená vo vlastnostiach videozáznamu. Čas medzi jednotlivými snímkami sa rovná obrátenej hodnote obrazovej frekvencie. Tento údaj bude neskôr potrebný pre výpočet rýchlosti (viď. kapitola 5), kde okrem prejdenej vzdialenosti potrebujeme aj čas, za ktorý objekt danú vzdialenosť prešiel. Čas získame súčinom počtu snímok, za ktorý objekt prešiel danú vzdialenosť a obrátenej hodnoty frekvencie.

Ďalší pojem súvisí s trasovaním objektu. Výsledkom trasovania je záznam (trasa) o pohybe (trajektórii) sledovaného objektu. V našom prípade je to zoznam súradníc jeho stredu. Pri spracovaní snímok sú ich rozmery a hodnoty parametrov opisujúcich vzdialenosť alebo rozmer vyjadrené v pixeloch (px). Treba preto aby mal užívateľ informáciu o sledovaných objektoch (približné rozmery) a vedel vhodne zvoliť všetky potrebné parametre. Pokiaľ vieme od poskytovateľa videozáznamu skutočné rozmery snímanej oblasti, je prepočet na px jednoduchý. Ak rozmery nie sú známe, využijeme iné biologické informácie. V našom prípade sme zvolili nasledovný postup: pomocou grafického softvérového nástroja Adobe Photoshop sme odmerali polomer krvinky na snímke videozáznamu. Nameraná hodnota bola približne 12px. Vieme, že polomer červenej krvinky je približne  $4\mu m$ . Po prepočte teda získame odhad veľkosti px ( $4\mu m \approx 3px$ ). Tento prepočet bude taktiež potrebný pre výpočet rýchlosti sledovaných objektov. Rovnako ako odhad veľkosti červenej krvinky sme získali aj odhad rozmerov mikrokanaľu. Presné rozmery kanálu bolo ťažké určiť, pretože zo snímok nebolo jasné kde sú hranice vnútornej steny kanálu.

## 1.1 Detekcia objektov

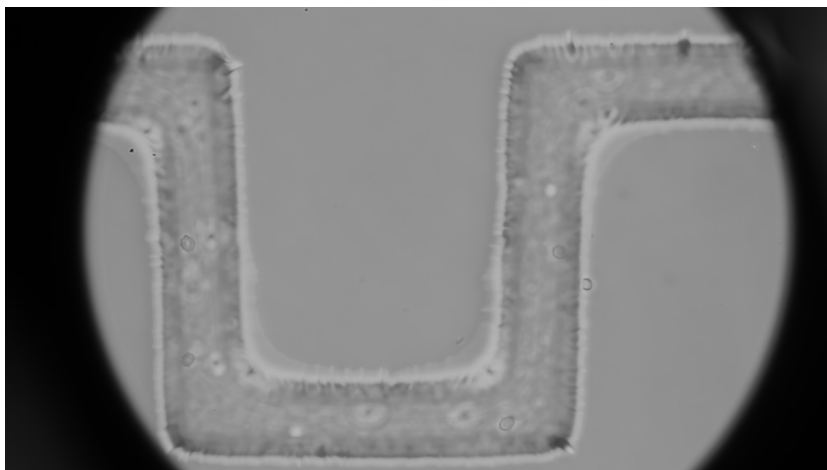
Náš post-processing sa skladal z dvoch hlavných častí: detekcia objektov a ich trasovanie. Pre trasovanie objektov je najprv nutné sledované objekty detegovať. Samotná detekcia objektov je obširná téma, ktorou sa zaoberá práca [2]. Pre predstavu uvedieme veľmi zjednodušený postup práce pri detekcii objektov. Pri spracovaní videozáznamu je nutné najprv video rozdeliť na jednotlivé snímky a následne ich postupne spracovávať.

Zo snímok je ďalej potrebné odstrániť pozadie- štandardný postup pri spracovaní videa.



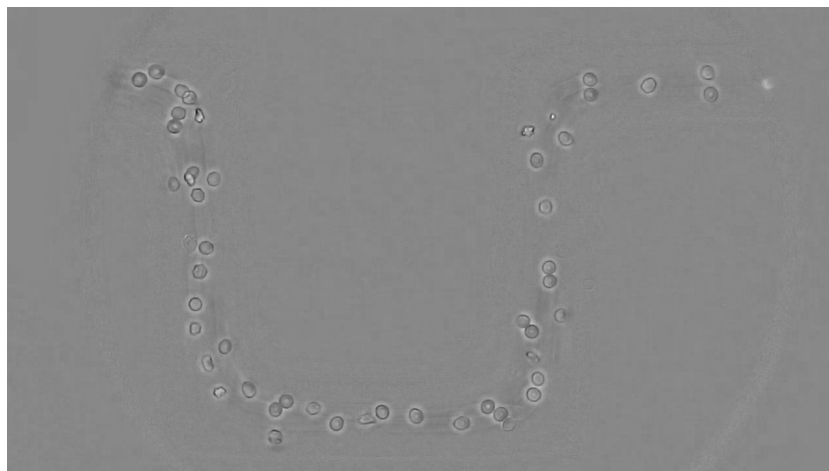
Obr. 1.1: Pôvodná snímka

Pod odstránením pozadia si môžeme predstaviť proces, ktorý odstráni zo snímok časti, ktoré sú statické- počas videa sa nemenia, nepohybujú.



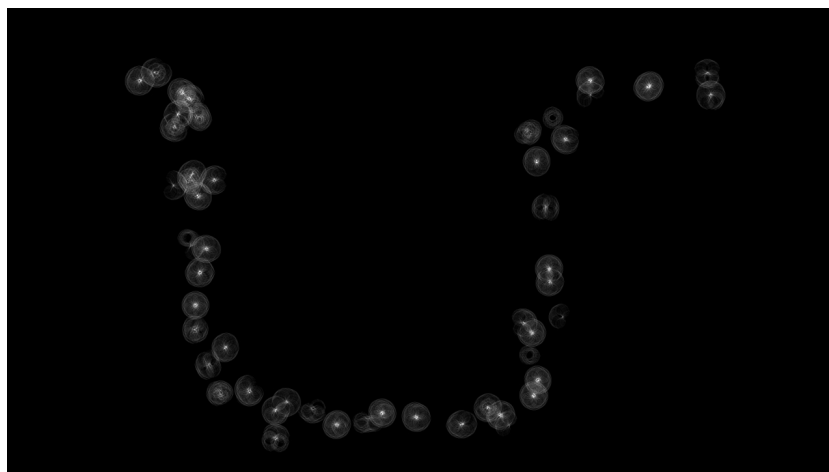
Obr. 1.2: Pozadie, ktoré bude zo snímky odstránené

Týmto sa stanú sledované objekty (objekty, ktoré chceme deketovať) viditeľnejšie. Snímky budú bez iných zbytočných objektov a šumov, ktoré by robili detekciu náročnejšou.



Obr. 1.3: Snímka po odstránení pozadia

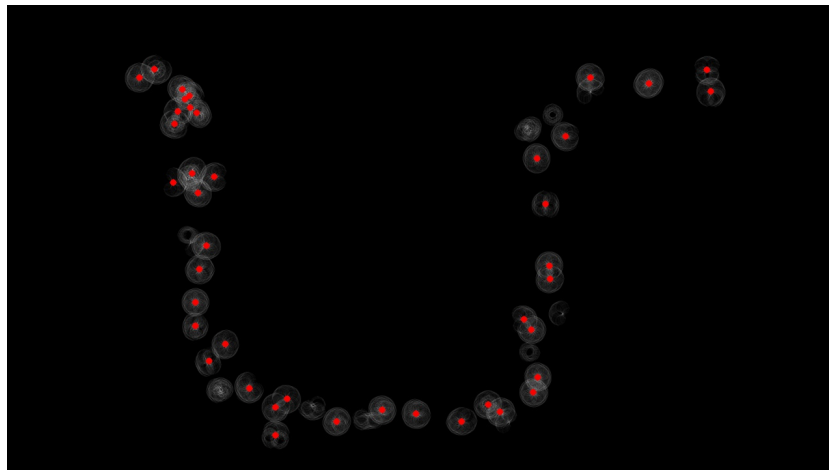
Po takejto úprave sú obrázky pripravené na samotnú detekciu. Súčasťou detekcie je Houghova transformácia, ktorá bola upravená pre naše účely a pomocou vykresľovania kružníc zvýrazní stredy kruhových objektov.



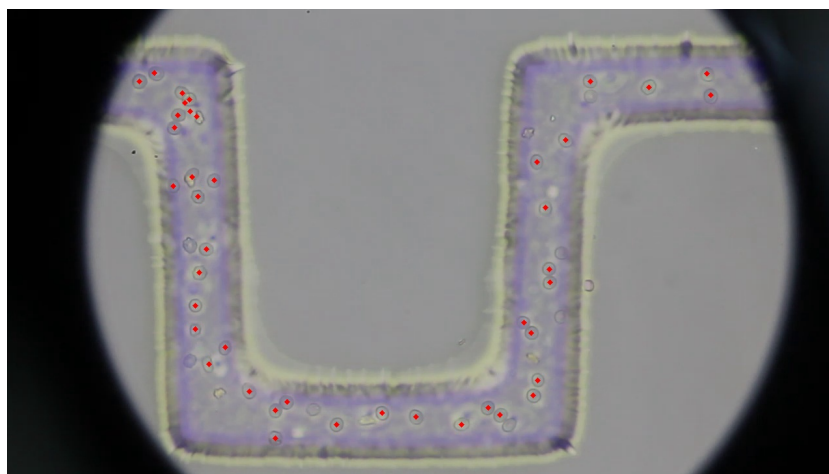
Obr. 1.4: Houghova transformácia



Ďalším krokom je detekcia stredov objektov, ktoré vznikli Houghovou transformáciou.



Obr. 1.5: Detekcia stredov



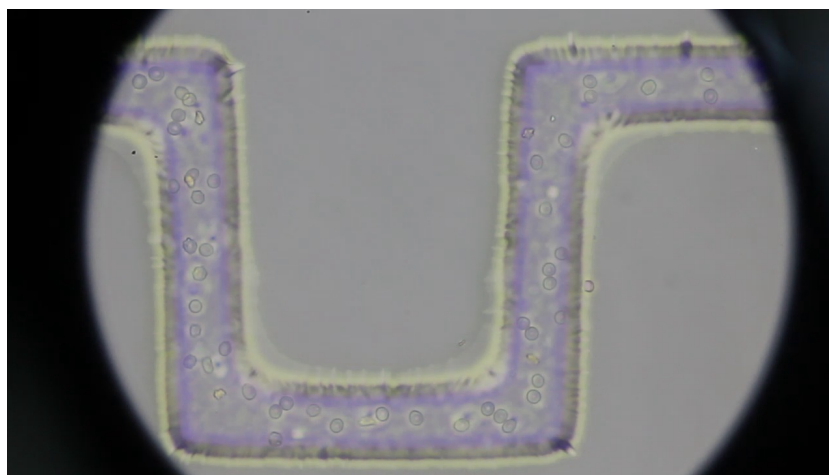
Obr. 1.6: Detegované stredy červených krviniek

Detegované stredy predstavujú stredy červených krviniek. Uvedený postup detekcie červených krviniek je veľmi zjednodušený. Podrobný postup spolu s vysvetlením je popísaný v práci [2]. Pre trasovanie červených krviniek je dôležitý výsledok detekcie, teda zoznam detegovaných bodov pre každú snímku, ktorý tvoril video záznam. Detegovaný bod je definovaný súradnicami jeho stredu na x-ovej a y-ovej osi.

## Kapitola 2

### Trasovanie červených krviniek

Problematikou spracovania videozáznamov biologických experimentov sme sa začali zaoberať v rámci projektovej výučby v štvrtom ročníku. Pri návrhu riešenia trasovania objektov sme museli brať ohľad na charakter spracovávaných videí. Na videu bol zobrazený úsek mikrokanálu a pohybuje sa krvinky (prípadne polystyrénové guľičky) v tekutine. Vzhľadom na to, že hovoríme o mikrokanále, bolo potrebné sa naň dívať pomocou mikroskopu.



Obr. 2.1: Ukážka mikrokanálu

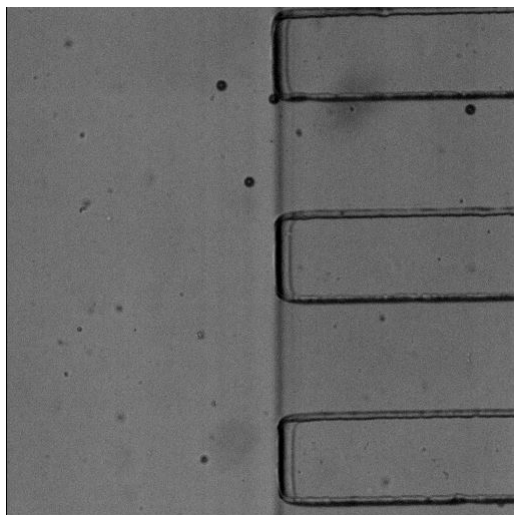
Prevažná väčšina videí bola zhotovená tak, že sa na okulár mikroskopu priložila kamera, ktorá snímala dianie pod mikroskopom. Je preto pochopiteľné, že kvalita záznamu bola nízka.

Pri pokuse o trasovanie červených krviniek sme sa stretli s problémami typickými pre túto oblasť:

- kvalita videozáznamu je asi najväčším problémom. Bohužiaľ, spolupracovníci, ktorí nám videá poskytujú v súčasnosti nemajú možnosť ako získať kvalitnejší záznam. Dôsledkom takéhoto snímania záznamu je, že stačí malý záchvev, napr. buchnutie do stola alebo mikroskopu, a video nebude možné spracovať v celom rozsahu, pretože už základné operácie ako napr. odstránenie pozadia (podrobnejšie popísané v kapitole 1) sa nevykonajú správne.
- šošovka mikroskopu je pri snímaní mikrokanálu zaostrená na určitú hĺbku. Keďže rozmery mikrokanálu umožňujú krvinkám plávať v rôznej hĺbke je prirodzené, že niektoré krvinky nebudú na zázname ostré práve pre rozdielnu hĺbku.
- prekrývanie krviniek počas putovania mikrokanálom je bežný jav, ktorý je však z hľadiska detekcie ťažko riešiteľný.
- vlastnosti červených krviniek neumožňujú využitie existujúcich trasovacích algoritmov. Mnohé algoritmy pre detegovanie a trasovanie objektov, sú založené práve na tom, že sledovaný objekt má stály tvar a ten sa deteguje. Keďže červené krvinky sú pružné a pôsobi na ne kvapalina, počas putovania kanálom menia svoj tvar a deformujú sa. Rovnako sme mohli vylúčiť aj detegovanie a trasovanie na základe farby. Detekovanie pomocou farby nebolo možné, keďže videá sú čierne-biele a okrem toho, krvinka v toku nemá jedinečnú farbu.

## 2.1 Dostupné nástroje

Na videách, ktoré sme v rámci projektovej výučby spracovávali, bol zaznamenaný jednoduchý experiment: polystyrénové guľičky, ktoré unášal tok tekutiny v mikrokanále. Videá zobrazovali vždy výsek kanálu o rozmeroch 819.2 x 819.2 mikrometrov.



Obr. 2.2: Obrázok kanálu

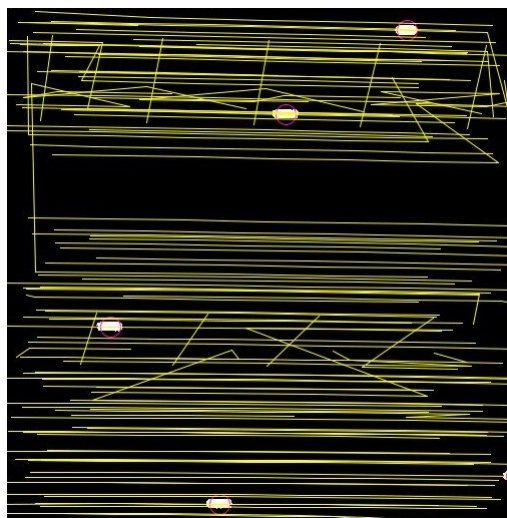
Cieľom bolo získať informácie o jednotlivých guľčkách (neskôr krvinkách). Požadované informácie boli priemerná rýchlosť a trajektória guľčky. Pri analýze riešenia bolo prvým krokom preskúmanie dostupných riešení vo forme hotových algoritmov či aplikácií.

Po analýze nástrojov pre spracovanie obrazu a videa sme zvolili použitie programu ImageJ, ktorý má nástroj TrackMate [3]. ImageJ je voľne dostupný softvér pre spracovanie obrazu a videa, navrhnutý najmä pre oblasť vedy a výskumu. Jeho architektúra umožňuje rozšírenia vo forme pluginov a makier. Okrem zobrazenia, umožňuje aj editáciu, analýzu a spracovanie snímok rôznych typov.

## TrackMate

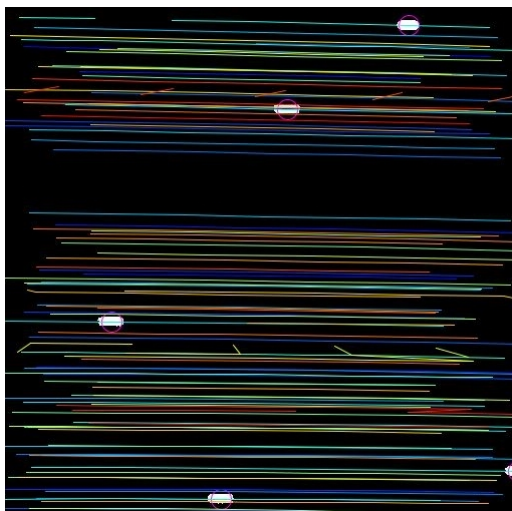
Nástroj TrackMate je určený pre trasovanie objektov. Trasy sú vytvárané v niekoľkých krokoch a na ich výsledný tvar má veľký vplyv voľba parametrov nástroja. V nasledujúcom texte popíšeme niektoré z nich. Jednou z kľúčových je voľba detektora, detegovacieho algoritmu. Nástroj ich ponúka niekoľko, podrobný popis je k dispozícii na [3]. Detektor slúži na detegovanie objektov na snímke. Je možné nastaviť parametre detektora, ako napr. estimated blob diameter (približná veľkosť hľadaného objektu) alebo threshold. Threshold je číselná hodnota, ktorá má za úlohu riešiť situácie, kedy by hrozilo, že bude detegované obrovské množstvo objektov. Každý objekt, ktorého kvalita je pod hodnotou threshold, nebude akceptovaný ako

detegovaný bod. My sme použili LoG detector s hodnotami parametrov, 20 pre estimated blob diameter a 0 pre threshold. Po nastavení parametrov sa pokračuje procesom detekcie. Počas procesu je zobrazený log panel, ktorý informuje o jeho priebehu. Po ukončení detekcie je ešte možné detegované objekty zmeniť (vyfiltrovať, okresať) napr. na základe ich kvality (rovnaká funkcia ako threshold) alebo podľa hodnoty súradnice. V ďalšom kroku sú detegované objekty pospájané a vznikajú tak trasy. Na vytváranie trás slúžia nástroje trackers, opäť ich je k dispozícii niekoľko. Pre naše účely na ukázal vhodný LAP tracker [4]. Parameter nástroja max distance zabezpečí, že body, ktoré sú od seba vzdialené viac ako daná hodnota nebudú nikdy považované za súčasť jednej trasy. Parameter feature penalties umožňuje ladenie získaných trás na základe hodnôt súradníc detegovaných objektov. Tento parameter sa ukázal ako mimoriadne účinný pri eliminácii chybné vytvorených trás.



Obr. 2.3: Získané trasy pred nastavením penalizácie

Na obrázku 2.3 vidíme trasy, ktoré nie sú priamočiare. Keďže z videozáznamu je jasne vidieť, že pohyb sledovaných guľčiek je priamočiary, je evidentné, že detegované trasy nie sú správne. Na základné vlastností pohybu guľčiek, sme teda mohli zaviesť penalizáciu na zmenu y-ovej súradnice (keďže pohyb je priamočiary, neočakávame, že súradnica sa bude výrazne meniť) vo forme nastavenia parametra feature penalties na hodnotu 10. Vďaka tomu, sa nám podarilo eliminovať chybné trasy.



Obr. 2.4: Získané trasy

Nastavenie parametrov bolo vhodné pre konkrétny videozáznam, ktorý sa spracovával. Pre spracovanie iného záznamu by bola potrebná ich kalibrácia, aby sa dosiahli čo najlepšie výsledky. Po príprave videa (rozparsovanie na jednotlivé snímky, odstránenie pozadia) a správnom nastavení potrebných parametrov nástroja TrackMate sme boli schopní získať požadované výstupy- trajektórie guľčiek. Trajektórie popisoval vytvorený xml súbor, ktorý mal nasledovnú podobu:

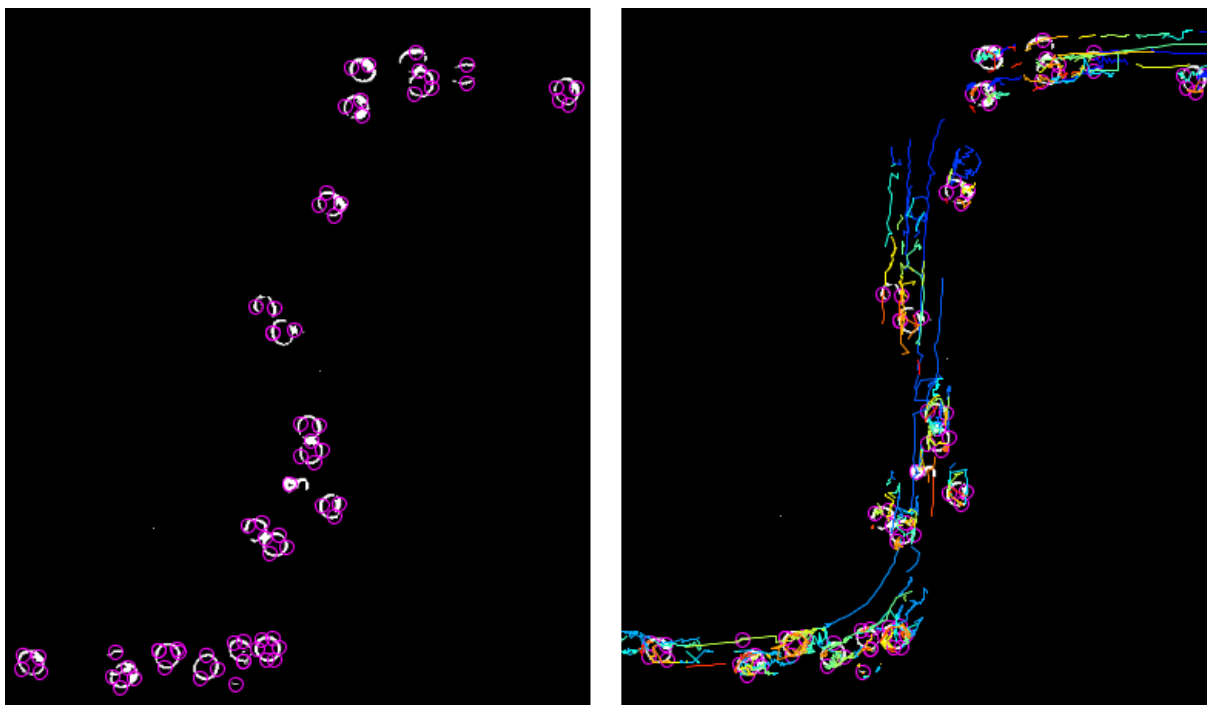
```
<Tracks nTracks="51" spaceUnits="pixel" frameInterval="1.0" timeUnits="frame">
  <particle nSpots="8">
    <detection t="3" x="501.168254" y="109.04619" z="0.0" />
    <detection t="4" x="390.115636" y="106.90949" z="0.0" />
    <detection t="8" x="390.115636" y="106.90949" z="0.0" />
    ...
  </particle>
</Tracks>
```

Pre získanie priemernej rýchlosti bolo potrebné súbor spracovať a rýchlosť vypočítať pomocou vzorca:

$$v = \frac{s}{t},$$

kde  $s$  označovalo prejdenu vzdialenosť a  $t$  čas, za ktorý bola vzdialenosť prejdenu.

Neskôr sme dostali na spracovanie videozáznamy, kde kanál nebol priamočiary ale rôzne ohnutý (viď. obrázok 2.1). Okrem toho sa v kanáli pohybovali červené krvinky a nie silikónové guľičky. S takýmto videozáznamom mal už TrackMate problém. Nedokázal správne vyhodnocovať situáciu a vzniknuté trasy neboli správne. Problém bol hlavne pri detegovaní červených krviniek. Pre predstavu uvádzame obrázok 2.5, ktorý zobrazuje výsledky detekciu a trasovanie červených krviniek pomocou nástroja TrackMate. Biele kruhy sú obrysy krviniek, ktoré chceme detekovať. Vo fialových kruhoch sú výsledky detekcie nástroja TrackMate.



Obr. 2.5: Detekcia a trasovanie pomocou nástroja TrackMate

Absencia nástrojov, ktoré by bolo možné použiť pri trasovaní červených krviniek viedla k vytvoreniu vlastného. Ten by dokázal riešiť problémy typické pre túto oblasť.

## 2.2 Vývoj vlastného nástroja

Po rozhodnutí implementovať vlastný nástroj pre post-processing videozáznamu sme vykonali analýzu dostupných nástrojov a knižníc, ktoré by sme mohli použiť. Okrem nástroja ImageJ sme zvažovali aj použitie knižnice OpenCV a SimpleCV.

### OpenCV

Open Source Computer Vision Library (OpenCV) je knižnica pre prácu s obrazom a videom, pre ich spracovanie. Knižnica je multiplatformová a voľne dostupná pre užívateľov. Má rozhrania pre jazyky C++, C, Java, Python a MATLAB.

Obsahuje množstvo algoritmov, ktoré sa dajú využiť napr. pre detegovanie a rozpoznávanie tvárí, identifikovanie objektov, trasovanie pohybujúcich sa objektov na základe farby alebo tvaru, trasovanie pohybu kamery, vyhľadávanie obrázkov z databázy na základe podobností, sledovanie pohybu očí a mnoho ďalších. Primárne je zameraná na real-time spracovanie obrazu a videa. [5]

### SimpleCV

SimpleCV je voľne dostupný framework pre tvorbu aplikácií zameraných na prácu s obrazom. Vďaka SimpleCV je možné použitie knižníc zameraných na spracovanie obrazu, ako napr. OpenCV, bez toho aby bolo potrebné mať poznatky o súborových formátoch, farebných škálach alebo správe pamäti. SimpleCV používa OpenCV a niektoré jeho funkcie robí jednoduchšími pre použitie. Rovnako ako pri OpenCV existuje množstvo voľne dostupnej dokumentácie a návodov na používanie. [6]

### Python

Python podporuje objektovo orientované, štruktúrované aj funkcionálne programovanie. Je to dynamicky typový jazyk a podporuje veľké množstvo vysokoúrovňových dátových typov.

Dôležitou vlastnosťou Pythonu je, že implementácia je veľmi rýchla a jazyk sa dá jednoducho rozširovať pomocou novo zabudovaných modulov. Tie môžu byť napísané napr. v C



alebo C++. Python môže byť rovnako dobre použitý ako rozširovací jazyk pre aplikácie, ktoré potrebujú programovateľné rozhranie alebo pre už existujúce moduly.

Vďaka rozsiahlej štandardnej knižnici je Python vhodný na množstvo úloh. Existujú knižnice napr. pre vytváranie grafického užívateľského rozhrania, pripájanie sa k relačným databázam, manipulovanie s regulárnymi výrazmi, pre prácu s Internetom a mnoho ďalších. Keďže je väčšia časť štandardnej knižnice kompatibilná medzi platformami, programy môžu pracovať na rôznych platformách bez zmeny. Knižnicu je možné rozširovať.

Ipython je interaktívny shell pre Python, ktorý podporuje okrem iného aj automatické dopĺňanie výrazov a syntax highlighting. Okrem toho Python disponuje aj debuggerom, profilerom a nástrojmi na testovanie.

Algoritmy sú implementované práve v jazyku Python. Spočiatku sme zvažovali aj implementáciu v jazyku Java, pretože s Pythonom sme nemali žiadne predchádzajúce skúsenosti. Napriek tomu, sme rozhodli pre použitie Python-u najmä kvôli jednoduchšej a prehľadnejšej práci s knižnicou OpenCV a väčšej užívateľskej základni (viac ľudí používa pre prácu s knižnicou OpenCV práve Python, vďaka čomu existuje množstvo návodov pre prácu, diskusií a fór atď.).[7] [8] Ďalším dôvodom prečo sme zvolili Python bol modul Object-in-fluid softvérového balíka ESPResSo. Modul sa používa pri simuláciach s elastickými objektami ponorenými do pohybujúcej sa tekutiny (podrobnejšie bude práca s modulom popísaná v kapitole 4). Pôvodne bol modul implementovaný v jazyku TCL, no v súčasnosti sa pracuje na jeho prepise do jazyka Python (keďže aj samotné ESPResSo má nové Python rozhranie). V prípade integrácie nášho nástroja a balíka ESPResSo (prípadne modulu Object-in-fluid) bude proces jednoduchší práve vďaka použitiu jazyka Python.

Pre prácu s Pythonom sme používali vývojové prostredie PyCharm (JetBrains).[9]

Pri implementácii vlastného nástroja sme použili knižnicu OpenCV spolu s jazykom Python. Realizácia je podrobne popísaná v kapitole 3.

## Kapitola 3

### Implementácia

Aby bolo možné trasovať objekty je nutné najprv objekty detegovať. Možnosťami detekcie červených krviniek sa zaoberá práca [2]. Ako bolo spomenuté v kapitole 1, video sa spracovávalo postupne po snímkach, teda aj trasy sa vytvárali postupne. Hlavnou myšlienkou algoritmu pre vytváranie trajektórii sledovaných objektov bolo vytvorenie prvotných trás, pri ktorých sme si istý, že body tvoriace túto trasu sú správne. Vytváranie prvotných trás prebiehalo nasledovne: detegované body sa budú spracovávať postupne po snímkach. Vezmeme prvý detegovaný bod B1 prvej snímky a pozrieme sa na detegované body nasledujúcej snímky. Ak nájdeme taký bod v nasledujúcej snímke, ktorý vieme jednoznačne priradiť k bodu B1, priradíme ho a vznikne tak trasa, ktorá zatiaľ obsahuje dva body. Takto sa postupne prejdú všetky detegované body, všetkých snímkov až vzniknú prvotné trasy. Jednoznačnosť priradenia bude popísaná v kapitole 3.2. Bolo zrejmé, že takto získané trasy nie sú finálne a bolo potrebné ich upraviť pomocou implementovaných metód popísaných v nasledujúcich kapitolách. Napriek tomu sme zvolili takýto postup, najmä z dôvodu uprednostnenia kvality pred kvantitou, máme síce menej a kratších trás, ale považujeme ich za správne.

V nasledujúcich kapitolách bude postupne podrobnejšie popísaný algoritmus vytvárania trás aj všetky implementované metódy. Názvy metód v texte zodpovedajú skutočným názvom metód v priloženom zdrojovom kóde.

### 3.1 Reprezentácia detegovaných objektov

Výstupom detekcie červených krviniek je xml súbor, ktorý obsahuje číslo snímky a súradnice stredov všetkých detegovaných bodov (krviniek). Ukážka xml súboru:

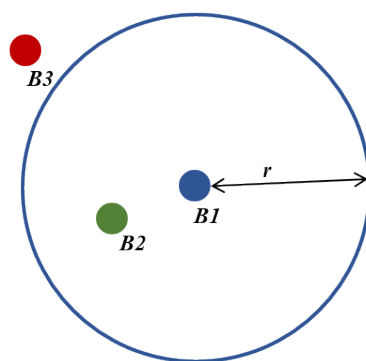
```
<RBC_detection framesCount="120">
<frame number="0" particlesCount="39">
<particle number="0" x="204" y="111"/>
<particle number="1" x="228" y="98"/>
<particle number="2" x="254" y="271"/>
<particle number="3" x="257" y="181"/>
...
```

Tento súbor bol vstupom pre trasovanie objektov.

Prvým krokom bolo teda načítanie hodnôt z xml súboru a ich uloženie do dátovej štruktúry. Ako štruktúru sme zvolili dvojrozmerné pole, maticu (nazveme ju matica detegovaných objektov). Matica je definovaná nasledovne:  $M = (o_{ij})$ , kde  $o_{ij}$  označuje  $j$ -ty detegovaný objekt na snímke  $i$ . Objekt je v matici reprezentovaný usporiadanou štvoricou  $[x, y, u, s]$ , kde  $x, y$  sú súradnice bodu,  $u \in \{0, 1\}$  označuje či je bod použitý v nejakej trase (táto hodnota je inicializačne nastavená na hodnotu 0 a počas vytvárania trás sa môže zmeniť na 1) a  $s$  označuje číslo snímky v ktorej bol objekt detegovaný. Napriek tomu, že sa prvok  $s$  zdá ako zbytočné, duplicitné uchovanie informácie o čísle snímky, na ktorej bol objekt detegovaný, nie je tomu tak. Tento údaj bude využívaný v neskoršom spracovaní vytvorených trás. V prípade, že by detegovaný objekt neobsahoval informáciu o čísle snímky, bolo by potrebné prechádzať maticu detegovaných objektov za účelom zistenia čísla snímky. Takýto postup by bol neefektívny a spomalil by proces spracovania trás. Matica detegovaných objektov je vstupom pre metódu `simple_tracking`, ktorá vytvára prvotné trasy. Naplnenie matice je súčasťou metódy `parse_xml`, ktorá parsuje a číta xml súbor.

## 3.2 Vytváranie trás

Metóda `simple_tracking` prechádza maticu detegovaných objektov a postupne vytvára trasy. Úlohou metódy je vytvorenie prvotných trás na základe jednoduchkej podmienky: ak sú detegované objekty v snímkach idúcich po sebe a nie sú od seba ďalej ako maximálna vzdialenosť (pozeráme sa do okolia objektu), tak ich považuje za súčasť jednej trasy. Ďalším kritériom pre spojenie je, že v okolí objektu sa nachádza práve jeden bod, ktorý vyhovuje daným podmienkam. V prípade, že v okolí objektu sa nachádza viac potenciálnych objektov na spojenie, spojenie sa neuskutoční. Podmienky boli takto nastavené z dôvodu, že sme chceli získať čo najistejšie (vybrané body s najväčšou pravdepodobnosťou tvoria trasu objektu), i keď možno kratšie, trasy. Pre lepšiu predstavu sú uvedené obrázky s popisom:



Obr. 3.1: Okolie s jedným bodom

Obrázok 3.1 znázorňuje ideálnu situáciu. Hľadá sa spojenie pre bod  $B1$  v snímke  $f_1$ .  $B1$  tvorí začiatok vytváranej trasy. Body  $B2$  a  $B3$  sú body v snímke  $f_1 + 1$ , je teda splnená prvá podmienka. Bod  $B2$  je jediným bodom, ktorý zároveň spĺňa aj podmienku vzdialenosti - vzdialenosť od bodu  $B1$  nie je väčšia ako  $r$  a teda dôjde k spojeniu bodov. Vzniká trasa, ktorá je zatiaľ tvorená bodmi  $B1$  a  $B2$ .

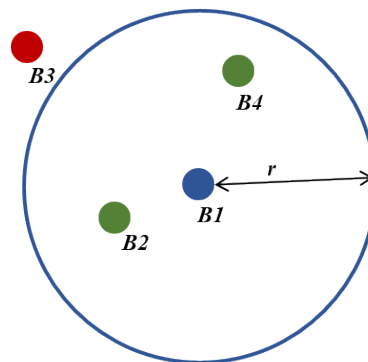
$r$  je parameter metódy, ktorý zadá užívateľ. Očakáva sa, že užívateľ má dostatok informácií o spracovávaných videá a vie preto vhodne odhadnúť hodnotu parametra. Výsledné trasy sú závislé od hodnoty vstupných parametrov a preto by sa mali čo najviac približovať realite. V našom prípade bola hodnota parametra nastavená na 12px čo približne zodpovedá polomeru reálnej červenej krvinky ( $\approx 4\mu m$ ). Prepočet je popísaný v kapitole 1. Hodnotu parametra

sme zvolili po odsledovaní a odmeraní pohybu niekoľkých červených krviniek na snímkach nasledujúcich za sebou.

Vzdialenosť medzi dvoma bodmi sme počítali ako Euklidovskú vzdialenosť:

$$d(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}, \quad (3.1)$$

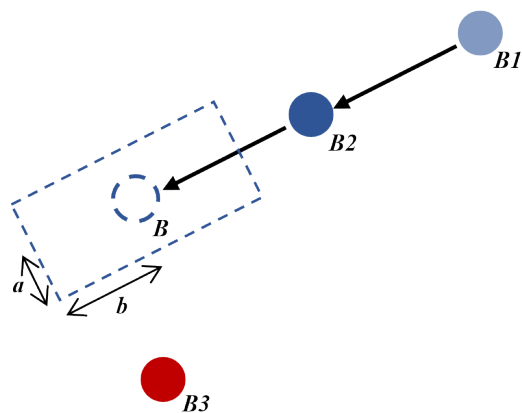
kde  $a = (x_a, y_a)$  a  $b = (x_b, y_b)$  sú dvojrozmerné vektory.



Obr. 3.2: Okolie s dvoma bodmi

Obrázok 3.2 popisuje situáciu, kedy sa hľadá spojenie pre bod  $B1$  v snímke  $f_1$ . Body  $B2$ ,  $B3$  a  $B4$  sú body v snímke  $f_1 + 1$ , je teda splnená prvá podmienka. Body  $B2$  a  $B3$  zároveň spĺňajú aj podmienku vzdialenosti. Keďže však neexistuje jedinečné spojenie pre bod  $B1$ , bod sa nespojí so žiadnym ďalším bodom a ostane označený ako nezaradený bod. Body, ktoré sú označené ako nezaradené, môžu byť ešte neskôr priradené do iných trás.

Takto sa prechádzajú všetky body matice detegovaných objektov a hľadajú sa nové spojenia. Keď sa spracováva bod (hľadá sa bod pre ďalšie spojenie), ktorý už je priradený v trase (v príklade je to bod  $B2$ ) pri hľadaní adeptov sa berie ohľad na doterajší smer už vytvorenej trasy. Prehľadávané okolie teda nebude opisovať tvar kruhu ale obdĺžnika, vid' Obrázok 3. Podľa doterajšieho smeru trasy sa vypočítajú súradnice bodu **B**. Tento bod nám slúži ako akýsi odhad, kde by sa mal nachádzať ďalší bod trasy. Skutočný bod, ktorý pridáme do trasy budeme teda hľadať práve v okolí tohto bodu.



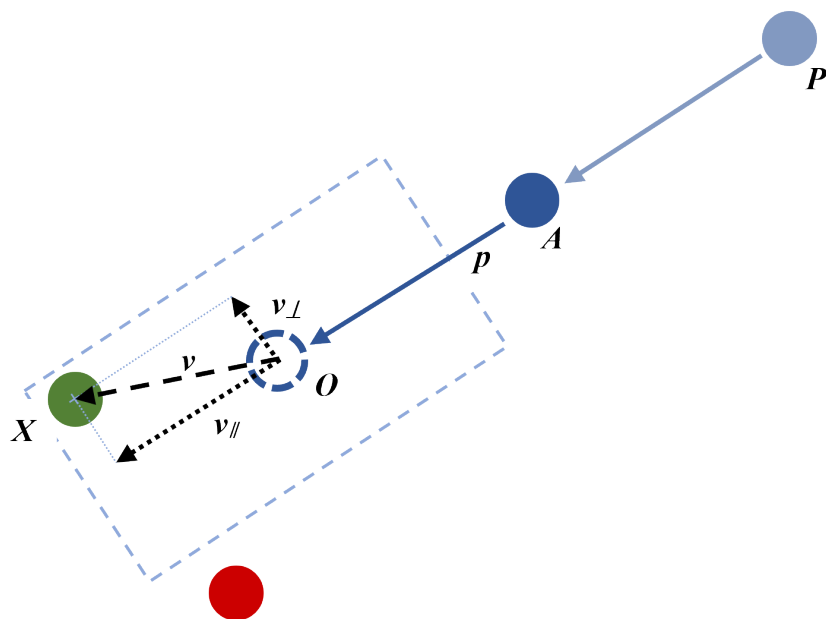
Obr. 3.3: Nové okolie

To či sa bod nachádza v okolí sme overovali výpočtom. Keďže sme poznali doterajší smer trasy, vedeli sme predpokladať, kde by sa mal nachádzať ďalší hľadaný bod. Súradnice očakávaného bodu (ozn.  $O$ ) sme vypočítali ako:

$$x_O = 2 * x_A - x_P$$

$$y_O = 2 * y_A - y_P,$$

kde  $A$  označuje aktuálny bod, pre ktorý hľadáme nasledovníka a  $P$  označuje bod ktorý je v danej trase predchodcom bodu  $A$ . Bod, ktorý predstavuje ďalší možný bod danej trasy je označený ako  $X$ . Pre lepšiu predstavu uvádzame obrázok 3.7, na ktorom vysvetlíme postup výpočtu.



Obr. 3.4: Vektory a body

Pre výpočet boli potrebné dva vektory:  $v_{\perp}$  a  $v_{\parallel}$ . Vektor  $v_{\perp}$  bol kolmý na vektor  $p$  a  $v_{\parallel}$  bol s vektorom  $p$  rovnobežný. Veľkosti vektorov sme vypočítali nasledovne:

$$v_{\parallel} = \frac{\vec{p}}{\|\vec{p}\|} * \|v\| * \cos \alpha = \frac{\vec{p}}{\|\vec{p}\|} * \frac{\|v\| * |(v,p)|}{\|v\| * \|\vec{p}\|} = \frac{|(v,p)|}{\|\vec{p}\|}$$

$$v_{\perp} = \sqrt{\|v\|^2 - \|v_{\parallel}\|^2}$$

Ak je sú teda veľkosti oboch vektorov ( $v_{\perp}$  aj  $v_{\parallel}$ ) pre bod  $X$  menšie, rovné ako hodnoty zadané v pamatery metódy, bod môžeme zaradiť do trasy. Rozmery prehľadávaného okolia sú opäť vstupom od užívateľa. Zmena pri hľadaní adeptov na spájanie pri už zaradených bodoch je v prehľadávanom okolí. Ostatné podmienky ako následnosť snímok a jedinečné priradenie ostali nezmenené.

Metóda končí, keď prejde všetky body matice detegovaných objektov. Výstupom metódy je zoznam vytvorených trás a zoznam bodov, ktoré ostali nezaradené. Oba zoznamy sú vstupom pre ďalšie metódy.

### 3.3 Vytvorenie matice toku

Keďže podmienky pri vytváraní prvotných trás v metóde `simple_tracking` boli pomerne striktné, vytvorené trasy boli v prevažnej väčšine krátke. Ďalším krokom teda logicky bolo spájanie kratších trás s cieľom vytvorenia dlhších trás. Pri analýze riešenia tohto problému sme dospeli k názoru, že by bolo nápomocné získať nejakú informáciu o toku a tým o smere a tvare požadovaných trás. Túto informáciu by sme potom mohli použiť pre správne vyhodnocovanie prípadného spájania dvoch trás do jednej. Naším cieľom bolo vytvorenie matice, v ktorej by boli uložené informácie o toku, resp. o očakávanom smere pohybu detegovaných objektov.

Matica sa naplňala dátami v dvoch krokoch. Najprv sa spracovali objekty, ktoré boli priradené do nejakého tracku. Spracovanie je náplňou metódy `calculate_flow_matrix`. V druhom kroku boli do matice doplnené údaje o objektoch, ktoré neboli zaradené do žiadnej trasy. Tieto údaje boli vypočítané pomocou špeciálnej interpolácie. Doplnenie matice zabezpečuje metóda `resolve_matrix`. Obe metódy sú podrobnejšie popísané v nasledujúcom texte.

Matica mala rozmery obrázkov, ktoré sme spracovávali. Prvok matice na pozícii  $[x][y]$  reprezentoval detegovaný objekt so súradnicami  $x, y$  a obsahoval informácie:  $n, [x_v, y_v]$ .  $n$  nadobúdalo hodnoty  $-1, 0$  alebo  $n \geq 1$  a vyjadrovalo počet bodov s danými súradnicami, ktoré boli zaradené do trasy.  $0$  slúžila pre body, ktoré ostali nezaradené (a nemáme teda informáciu o smere pohybu objektu) a  $-1$  bola inicializačná hodnota.  $[x_v, y_v]$  vyjadrovali hodnotu smerového vektora, teda smer ktorým sme očakávali že sa bude objekt so súradnicami  $[x, y]$  pohybovať (smer, ktorým ho bude tok unášať). Tento smerový vektor nám slúžil ako jedno z kritérií pri spájaní trás.

#### Metóda pre spracovanie zaradených bodov

Úlohou metódy `calculate_flow_matrix` bolo prvotné naplnenie matice toku dátami o objektoch, ktoré boli zaradené v nejakej trase. Postupne sa prechádzali všetky trasy a ich body. Pre názornosť uvádzame príklad: povedzme, že ideme spracovávať trasu  $T$ , ktorý tvoria body  $B1, B2, B3$  so súradnicami  $[x_1, y_1], [x_2, y_2]$  a  $[x_3, y_3]$ . Ako prvý budeme spracovávať bod  $B1$ : pozrieme sa do matice toku na pozícii  $x_1, y_1$ . Ak je hodnota  $n$  rovná  $-1$  vieme, že bod  $B1$  je prvým bodom s týmito súradnicami, ktorý spracovávame. Hodnotu  $n$  teda nastavíme na  $1$  a



súradnice smerového vektora vypočítame, tak že súradnice spracovávaného bodu (aktuálneho, ozn.  $a$ ) ( $B1$ ) odpočítame od súradníc nasledovného (ozn.  $b$ ) bodu v trase ( $B2$ ):

$$x_v = x_n - x_a$$

$$y_v = y_n - y_a$$

Označenie  $x_v, y_v$  sa bude aj ďalej v texte používať pre hodnoty vektora, prípadne súradnice bodu (napr. pre bod  $B1$ )  $x_1, y_1$ .

Získame tak informáciu o smere, ktorým sa sledovaná bunka ubera. V situácii, že hodnota  $n$  je väčšia, rovná 1 (znamená, že existuje viac bodov s danými súradnicami  $x, y$ , čo je jav, ktorý môže nastať) je potrebné vypočítať priemerný vektor. Berieme tak do úvahy všetky smery, ktorými sa detegové body v trasách pohybovali. Výpočet priemerného vektora:

$$x_{vp} = \frac{n * x_{vm} + x_v}{n+1}$$

$$y_{vp} = \frac{n * y_{vm} + y_v}{n+1}$$

$$n = n + 1,$$

kde  $vm$  je hodnota smerového vektora, ktorá je aktuálne uložená v tokovej matici,  $v$  je hodnota smerového vektora spracovávaného bodu a  $vp$  je hodnota priemerného smerového vektora, ktorý bude uložený v matici miesto vektora  $vm$ . Okrem hodnoty smerového vektora sa aktualizovala aj hodnota  $n$ , ktorá sa zvýšila o jedna.

Pre výpočet priemerného vektora sme použili aritmetický priemer všetkých vektorov bodov s danými súradnicami, uvádzame dôkaz: ak

$$p = \frac{\sum_{i=1}^n x_i}{n}$$

je aritmetický priemer štatistického súboru, kde  $n$  je rozsah súboru a  $x_i$  sú prvky súboru, potom aj

$$k = \frac{n * p + x_{n+1}}{n+1}$$

je aritmetický priemer štatistického súboru, kde  $n + 1$  je rozsah súboru, pretože  $k$  môžeme vyjadriť ako

$$k = \frac{n * \frac{\sum_{i=1}^n x_i}{n} + x_{n+1}}{n+1} = \frac{\sum_{i=1}^{n+1} x_i}{n+1}.$$

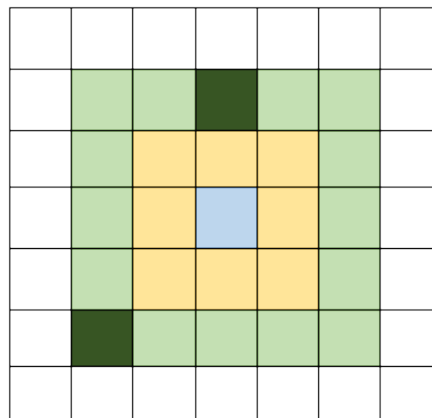
Pri posledných bodoch trás sa hodnota smerového vektora nastavila na rovnakú hodnotu ako pri predposlednom bode, keďže nie je možné vektor vypočítať a očakávali sme, že smer trasy sa nebude výrazne meniť.

## Metóda pre spracovanie nezaradených bodov

Metóda `resolve_matrix` mala za úlohu spracovať nezaradené body. Pri nezaradených bodoch sme zvolili nasledovný postup: ak mal nezaradený bod rovnaké súradnice ako zaradený bod a teda hodnota vektora pre bod s danými súradnicami už bola vypočítaná (hodnota  $n$  v tokovej matici bola rôzna od -1) v predchádzajúcom kroku (pomocou metódy `calculate_flow_matrix`), hodnotu vektora sme neaktualizovali.

V prípade, keď nezaradený bod nemal rovnaké súradnice ako zaradený bod, bolo potrebné určiť smerový vektor. Body sa v kanály pohybujú rovnakým smerom. Ak sa teda isté body nachádzajú v určitom mieste kanála, vieme, že pôjdu rovnakým, alebo aspoň veľmi podobným smerom. Smerový vektor pre nezaradený bod sme teda určili tak, že sme našli susedný zaradený bod a jeho vektor sme potom prehlásili za vektor nezaradeného bodu.

Okolie nezaradeného bodu sa prehľadávalo postupne. Ak sa v jeho najbližšom okolí nenachádzal žiaden zaradený bod, metóda sa rekurzívne zavolala a prehľadalo sa väčšie okolie. Ak sa v okolí zaradených bodov našlo viac, vypočítal sa priemerný vektor, pričom mal väčšiu váhu vektor toho zaradeného bodu, ktorý bol k nezaradenému bodu bližšie.



Obr. 3.5: Prehľadávanie okolia

Ukážka prehľadávania okolia pre nezaradený bod (na obrázku modrá kocka): pri prehľadávaní najbližšieho okolia bodu (zobrazené žltou farbou) sa nenašiel žiaden zaradený bod. Prehľadá sa teda väčšie okolie (zobrazené zelenou farbou), v ktorom sa našli zaradené body a prehľadávanie preto končí. Hodnota priemerného vektora pre nezaradený bod sa počíta ako vážený aritmetický priemer s váhami rovnými obrátenej hodnote vzdialenosti:

$$x_{vp} = \frac{\sum_{i=1}^c \frac{1}{d(u, r_i)} * x_{vr_i}}{\sum_{i=1}^c \frac{1}{d(u, r_i)}}$$

$$y_{vp} = \frac{\sum_{i=1}^c \frac{1}{d(u, r_i)} * y_{vr_i}}{\sum_{i=1}^c \frac{1}{d(u, r_i)}},$$

kde  $c$  je počet nájdených zaradených bodov v okolí,  $u$  označuje nezaradený bod pre ktorý počítame vektor,  $r_i$  je nájdený zaradený bod,  $d(u, r_i)$  je vzdialenosť medzi  $i$ -tým zaradeným a nezaradeným bodom,  $v_{r_i}$  je hodnota vektora  $i$ -teho zaradeného bodu.

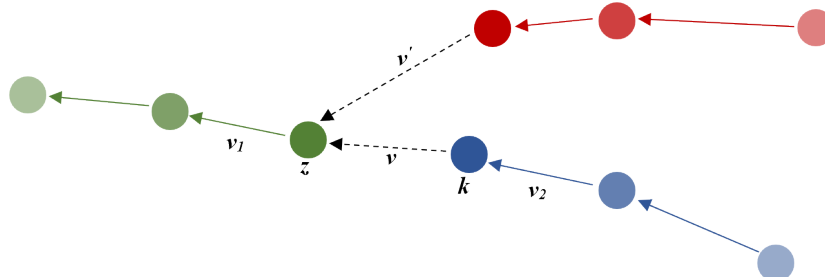
Pre nezaradený bod so súradnicami  $[x, y]$  bude teda mať toková matica na pozícii  $[x][y]$  hodnotu  $[0, [x_{pv}, y_{pv}]]$ . Nastavenie  $n$  na 0 slúži ako príznak toho, že hodnota vektora je dopočítaná z okolia bodu a nie zo smeru trasy. Po spracovaní nezaradených bodov je matica kompletná a pripravená na použitie pri spájaní trás.

### 3.4 Spájanie trás

Spájanie trás zabezpečuje niekoľko metód. Prvou z nich je metóda `simple_joining`. Postupne prechádza trasy a zisťuje, ktoré dve by bolo možné spojiť a vytvoriť tak jednu, dlhšiu trasu. Má dva parametre. Jedným z nich je zoznam doposiaľ vytvorených trás (výstup z metódy `simple_tracking`) a druhým je hraničná vzdialenosť medzi bodmi dvoch trás, pri ktorej ešte dovoľíme ich spojenie.

Na to aby metóda vyhodnotila dve trasy ako adepty na spojenie, musia byť splnené podmienky:

- následnosť framov
- vektor  $v$ , ktorý by vznikol spojením trás, sa musí nachádzať medzi vektormi  $v_1$  a  $v_2$



Obr. 3.6: Spájanie trás: vektor

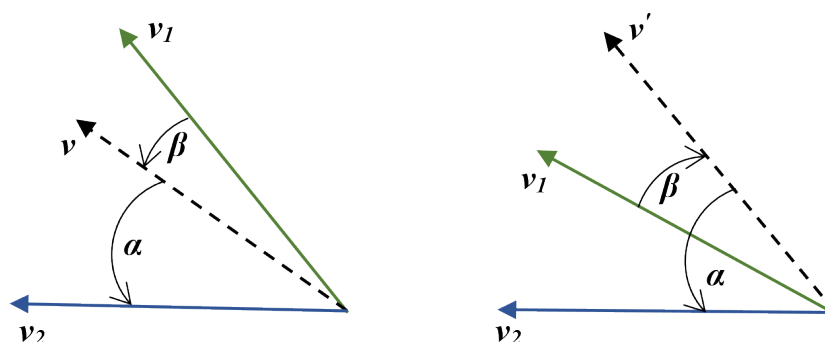
Podmienky vysvetlíme na obrázku 3.6. Majme dve trasy:  $T_1$  (modrú),  $T_2$  (zelenú). Na to, aby sme mohli trasy spojiť musí platiť, že ak trasa  $T_2$ , končí v bode  $k$  na snímke  $s$ , tak potom trasa  $T_1$  musí začínať v bode  $z$  v snímke  $s + 1$ . Takto je zabezpečená následnosť snímok.

Ďalšou podmienkou pre spojenie trás je vektor  $v$ . Aby bolo možné dve trasy spojiť musí platiť, že novovzniknutý vektor sa nachádza medzi vektormi  $v_1$  a  $v_2$ . Na overenie tejto podmienky sme použili vektorový súčin:

$$v_1 \times v = v_{1x} * v_y - v_{1y} * v_x$$

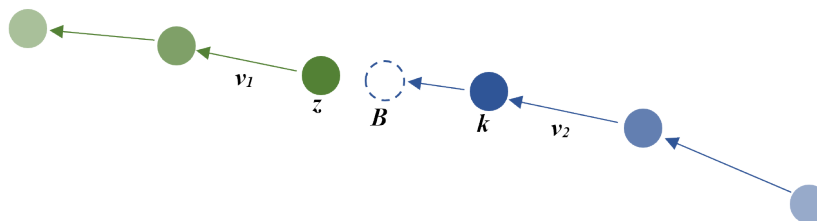
$$v \times v_2 = v_x * v_{2y} - v_y * v_{2x}$$

Ak boli oba výsledné vektory záporné alebo kladné vedeli sme, že orientované uhly medzi vektormi majú rovnaký smer a teda  $v$  sa nachádzal medzi vektormi  $v_1$  a  $v_2$ . V opačnom prípade mali orientované uhly rozdielny smer a  $v$  sa nenachádzal medzi vektormi  $v_1$  a  $v_2$ . Pre ilustráciu uvádzame obrázok 3.7.



Obr. 3.7: Uhly medzi vektormi a ich smer

Ak sa  $v$  nenachádzal medzi vektormi  $v_1$  a  $v_2$ , trasa  $T_1$  bola zaradená do zoznamu kandidátov pre spojenie trasy  $T_2$ . Po prejdení všetkých trás bol zoznam kandidátov pre danú trasu kompletný. Ostávalo vybrať ten, ktorý je pre spojenie najlepší. Kritériom pre výber najlepšieho kandidáta bude vzdialenosť prvého bodu kandidáta a očakávaného bodu trasy. Keďže vďaka tokovej matici sme vedeli, ktorým smerom by sa trasa  $T_2$  mal uberať, vieme vypočítať aký by mal byť nasledujúci, očakávaný bod trasy. Vybraným kandidátom pre spojenie sa stane tá trasa, ktorej bod je najbližšie k očakávanému bodu **B**.



Obr. 3.8: Spájanie trás: očakávaný bod

Metóda umožňuje pomocou parametra zvoliť hranicu tejto vzdialenosti, tzn. kandidát bude vybraný ak je vzdialenosť bodov menšia ako zadaná hranica. Parameter je opäť vstupom od užívateľa a očakáva sa, že ho zvolí s ohľadom na vlastnosti záznamu.

Metóda vráti dva zoznamy: zoznam dvojíc trás na spojenie, spolu s hodnotou vzdialenosti medzi spájanými bodmi a zoznam trás, pre ktoré sa nenašli žiadne vyhovujúce trasy na spojenie.

Ďalšou metódou, ktorá sa zaoberá spájaním trás je `check_duplicity`, ktorá prejde zoznam adeptov na spojenie a skontroluje, či sa v ňom nenachádza duplicita. Napr. vznikla dvojica  $[T_2, T_1]$  ktorých vzdialenosť bodov je 0,85 a zároveň vznikla dvojica  $[T_5, T_1]$  ktorých vzdialenosť je 1,54. Metóda nájde túto duplicitu a vyberie tú dvojicu, ktorá je pre spojenie vhodnejšia, tú s menšou vzdialenosťou. Ak sa v zozname žiadne duplicity alebo viacnásobné výskyty nenašli, zoznam dvojíc ostáva nezmenený.

Takto skontrolovaný zoznam je vstupom pre ďalšiu metódu `find_multiple_merge`. Metóda prechádza zoznam dvojíc na spojenie a hľadá dvojice, ktoré tvoria násobné spájanie trás. Napr v zozname sú dvojice  $d_1 = [T_2, T_1], d_2 = [T_8, T_{10}], d_3 = [T_1, T_6], \dots, d_{15} =$

$[T_6, T_{12}], d_{16} = [T_9, T_5], \dots$  Vidíme, že dvojice  $d_1, d_3$  a  $d_{15}$  obsahujú zhodné trasy. Je preto možné vytvoriť násobné spojenie trás a to tak, že miesto troch dvojíc vznikne jedna štvorica  $s_1 = [T_2, T_1, T_6, T_{12}]$ . Takýmto spôsobom vznikne nový zoznam, kde sa okrem dvojíc nachádzajú aj n-tice, v ktorých sú trasy vhodné na spojenie. Tento zoznam už je konečný a je vstupom pre metódu, ktorá spája trasy.

Metóda sa volá `merge_tracks` a na základe vstupných parametrov (zoznam trás, zoznam trás na spojenie, zoznam trás, ktoré sa nebudú spájať) vykoná spájanie trás. Výstupom je nový zoznam trás, kde sa okrem pôvodných trás pre ktoré sa nenašli adepti na spojenie, nachádzajú aj nové trasy, ktoré vznikli po spojení.

Po tomto kroku môže nasledovať viac činností. Jednou z nich je opätovné spustenie metód pre spájanie trás. Keďže vznikli nové trasy, je možné, že sa nájdu ďalší adepti na spojenie, alebo sa vykoná metóda, ktorá sa snaží o zaradenie dosiaľ nezaradených bodov.

### 3.5 Pripájanie nezaradených bodov

Metóda `try_resolve` má ako vstupné parametre zoznam nezaradených bodov, zoznam trás a hodnotu maximálnej vzdialenosti pri priradovaní bodov do trás. Vzdialenosť je opäť vstupom od užívateľa. Metóda postupne prechádza trasy a nezaradené body a zisťuje, či by sa nezaradený bod mohol pripojiť na začiatok alebo koniec trasy. Robí to nasledovným spôsobom: najskôr je nutné overiť následnosť framov. V prípade, že chceme bod, ktorý sa nachádza v snímke  $s$  pripojiť na začiatok trasy, je nutné aby trasa začínala (jej prvý bod sa nachádzal) v snímke  $s + 1$ . Ak naopak chceme bod pripojiť na koniec trasy je potrebné aby trasa končila (jej posledný bod) v snímke  $s - 1$ .

Ak toto bod spĺňa, overí sa ešte predpokladaný smer pohybu bodu voči smeru trasy. Pri pripájaní na začiatok sa skontroluje či smer, ktorým ide trasa (z prvého do druhého bodu) je rovnaký ako očakávaný smer pohybu nezaradeného bodu so súradnicami  $[x, y]$  zapísaného v matici toku na pozícií  $[x][y]$ . Pri pripájaní na koniec trasy sa kontroluje opäť smer nezaradeného bodu voči smeru konca trasy (z predposledného na posledný bod). Ak nezaradený bod svojim očakávaným smerom vyhovuje podmienke, uloží sa do zoznamu kandidátov pre pripojenie na začiatok, alebo do zoznamu kandidátov pre pripojenie na koniec trasy. Následne

sa prejdú oba zoznamy a vyberú sa najvhodnejší kandidáti. Pri výbere najvhodnejšieho kandidáta sa uplatňuje podobný postup ako pri spájaní trás v kapitole 3.4. Vypočíta sa očakávaný bod a ako najvhodnejší bod pre pripojenie sa vyberie ten, ktorý je najbližšie k očakávanému bodu. Ak je bod vybraný ako najvhodnejší pre spojenie, je označený ako použitý (zmena z 0 na 1 v usporiadnanej štvorici popisujúcej bod, vid' kapitolu 3.1) a pripojený k danej trase. Výstupom metódy je zoznam nových trás a aktualizovaný zoznam nezaradených bodov.

Aj túto metódu je možné opätovne spustiť a pokúsiť sa priradiť ďalšie nezaradené body. Prípadne spustiť metódu na spájanie trás (doplnené body mohli vytvoriť nové možnosti spojenia) a následne sa opäť pokúsiť priradiť nezaradené body. Je na užívateľovi, aký postup zvolí.

## 3.6 Výstup

Keď užívateľ dosiahne konečnú podobu trás, je možné vykonať export do xml súboru. Súbor obsahuje informáciu o počte trás ako aj informácie o jednotlivých trasách: počet bodov, popis trajektórie. Ukážka xml súboru:

```
<RBC_tracking tracksCount="337">
  <track number="0" particlesCount="46">
    <particle frame="0" x="293" y="288"/>
    <particle frame="1" x="293" y="286"/>
    <particle frame="2" x="293" y="286"/>
    <particle frame="3" x="294" y="284"/>
    ...
  </track>
</RBC_tracking>
```

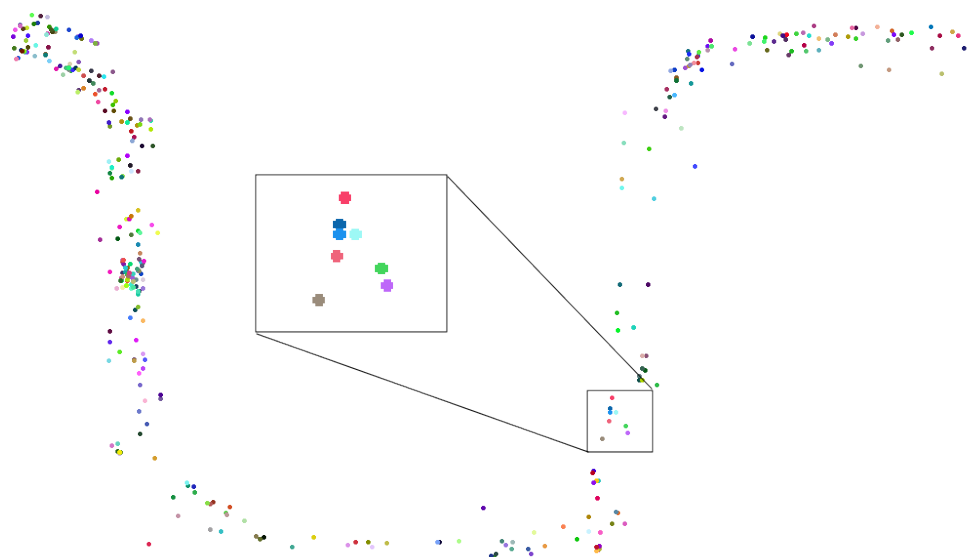
Vytvorenie a zápis do súboru zabezpečuje metóda `generate_track_xml`, ktorej parametrami sú zoznam vytvorených trás a názov súboru.

## 3.7 Metódy pre zobrazenie

Okrem vyššie popísaných metód boli vytvorené aj jednoduché metódy pre zobrazenie detegovaných bodov alebo vzniknutých trás.

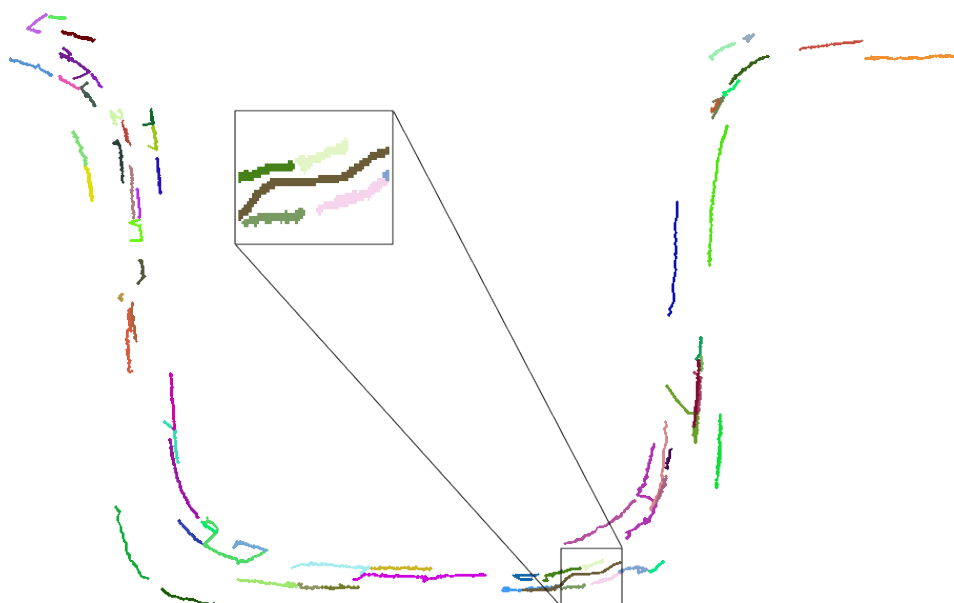
Metóda `draw_points` vykreslí detegované body. Vstupnými parametrami sú matica detegovaných objektov, obrázok, na ktorý sa majú body vykresliť a polomer kružnice, ktorá bude zobrazovať bod. Okrem toho je možné zadať aj rozsah snímok (od, do), ktoré chceme vykresliť. Ak rozsah nie je zadaný, vykreslia sa všetky body. Aj je zadaná iba jedna hodnota, vykreslia sa body na zadanej snímke. Pre vykreslenie kružnice, ktorá reprezentuje bod sme použili funkciu knižnice OpenCV `circle`.





Obr. 3.9: Zobrazenie bodov

Metóda `draw_tracks` pracuje podobne ako metóda `draw_points`. Vstupnými parametrami sú zoznam trás, obrázok, na ktorý sa trasy vykreslia a opäť rozsah v tvare `od, do`. Ak rozsah nie je zadáný, vykreslia sa všetky trasy. Ak je zadaná iba jedna hodnota, vykreslí sa daná trasa. Pre vykreslenie trasy sme použili funkciu knižnice OpenCV `line` [10].



Obr. 3.10: Zobrazenie trás

### 3.8 Možné budúce implementácie

V budúcnosti by bolo možné pokúsiť sa implementovať aj trasovanie, ktoré berie viac do úvahy charakter detekcie. Pri bližšom skúmaní výstupov z procesu detekcie bolo viditeľné, že veľa objektov bolo detegovaných prerušovane, tzn. na snímke číslo  $s - 1$  daný objekt bol detegovaný, na snímke číslo  $s$  nie a na snímke číslo  $s + 1$  sa objekt opäť podarilo zachytiť. Takého výpadky vznikali počas detekcie bežne. Preto by bolo vhodné algoritmus trasovania v budúcnosti rozšíriť a doplniť jeho funkcie, tak aby sa vedeli vysporiadať aj s touto situáciou (napr. by sa neprehľadávala vždy iba jedna nasledujúca snímka ale niekoľko. Ich počet by bol vstupom od užívateľa.).

## Kapitola 4

### Simulácia experimentu

Vytvorenie simulačného modelu a simulácie experimentu by nebolo možné bez spolupráce s výskumnou skupinou cell-in-fluid. Skupina nám poskytla simulačné skripty, uviedla nás do problematiky simulácie v danej oblasti a poskytla cenné rady pri vytváraní modelu a simulácií experimentu. Skupina cell-in-fluid sa zaoberá modelovaním toku krvi na úrovni buniek a úzko spolupracuje s vývojármi nástroja ESPResSo na univerzite v Štutgarte. Okrem toho skupina spolupracuje s Krems University, ktorá sa podieľa na vývoji modelov. Spracovávané videozáznamy boli poskytnuté práve spolupracovníkmi z Krems University. Videozáznam zachytával experiment, v ktorom prúdila krv mikrokánalom. Krv bola riedená, aby sa docielil požadovaný hematokrit. Mikrokánal bol vyrobený spojpomocne pracovníkmi univerzity, ktorí sa podieľajú na výskume. [16]

### Simulačné nástroje

Pri modelovaní a simulácii sa využíval softvérový balík ESPResSo, najmä modul Object-in-fluid. Balík sa využíva pre vykonávanie vedeckých simulácií molekulovej dynamiky v oblasti molekulárnej biológie, chémie či fyziky a ich analýzu. Je to voľne dostupný, open-source softvér, používaný výskumnými skupinami po celom svete. Medzi používateľov ESPResSo-a patrí aj výskumná skupina cell-in-fluid.

Pre prácu s ESPResSo-om sa používa skriptovací jazyk TCL. Medzi výhody Tool Com-

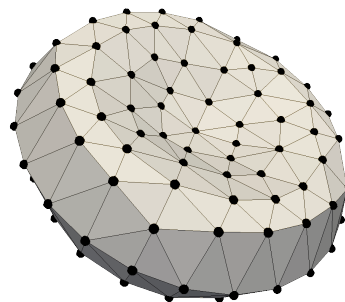
mand Language patrí jednoduchosť tvorby programov a ich jednoduché ladenie. Jazyk má jednoduchú syntax, ktorá však môže byť v niektorých prípadoch spočiatku neprehľadná a nepohodlná, najmä pri vyhodnocovaní premenných či matematických výrazoch. [11] [13] [14]

Pre zobrazenie výsledkov simulácie sme použili softvér ParaView. Nástroj je vhodný pre vytváranie vizualizácií dát a ich analýze. ParaView je voľne dostupná, multiplatformová aplikácia postavená na knižniciach VTK (Vizualization Tool Kit). Výstupom simulácie boli okrem trajektórií elastických objektov reprezentujúcich červené krvinky (podrobnejšie bude súbor popísaný v podkapitole 4.3) aj vtk súbory, ktoré popisovali elastický objekt. ParaView sme použili práve na prezeranie vtk súborov, čo nám umožnilo sledovať priebeh simulácie.

## 4.1 Popis modelu

Pre lepšiu predstavu čitateľa o modeli, s ktorým prebiehajú simulácie, uvedieme veľmi zjednodušený popis. Model je tvorený dvoma časťami, elastickým objektom a tekutinou.

Objekt je modelovaný pomocou povrchovej triangulácie. Povrchová triangulácia predstavuje trojuholníkovú sieť, ktorá pokrýva povrch objektu. Body tvoriace sieť predstavujú hmotné body, ktoré tvoria trojuholník. Body sú navzájom prepojené pružinami- elastickými silami.

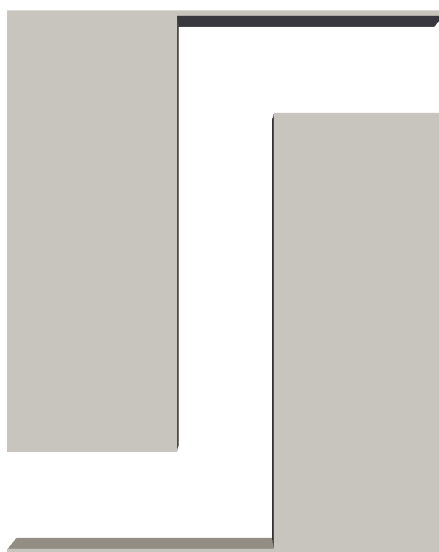


Obr. 4.1: Elastický objekt

Pre modelovanie tekutiny bola použitá lattice-Boltzmannova metóda. Vychádza z predpokladu, že kvapalina je zložená z fiktívnych bodov, ktoré sú uložené v pevnej mriežke (kocka s ôsmimi vrcholmi).

Obe časti modelu na seba navzájom pôsobia: objekt pôsobí na tekutinu, tekutina pôsobí na objekt. Interakcia bola simulovaná použitím metódy vnorenej hranice (immersed boundary method).[15]

Okrem elastického objektu reprezentujúceho červenú krvinku a tekutiny sa v simulačnom modeli nachádzal aj model samotného kanálu. Ten bolo potrebné navrhnuť a vytvoriť. Približné rozmery reálneho kanálu sme získali z videozáznamu. Použiť model, ktorý by rozmermi zodpovedal skutočnému kanálu však nebolo možné. Model celého kanálu spolu s prúdiacou tekutinou a pohybujúcimi sa elastickými objektami by obsahoval veľké množstvo diskretizovaných bodov buniek ako aj tekutiny, a simulácia by bola časovo a výpočtovo náročná. Preto sme sa rozhodli, že budeme simulovať iba časť kanálu. Modelovaná časť kanálu zodpovedá svojimi rozmermi kanálu z experimentu.



Obr. 4.2: Model kanálu

## 4.2 Simulácia experimentu

Pre simuláciu experimentu popísaného v kapitole 2, sme použili skript `simulation.tcl`, ktorý nám bol poskytnutý skupinou `cell-in-fluid`. Skript sme upravili pre účely simulácie, ktorej cieľom bolo sledovať pohyb elastických objektov (krviniek) v prúdiacej kvapaline.

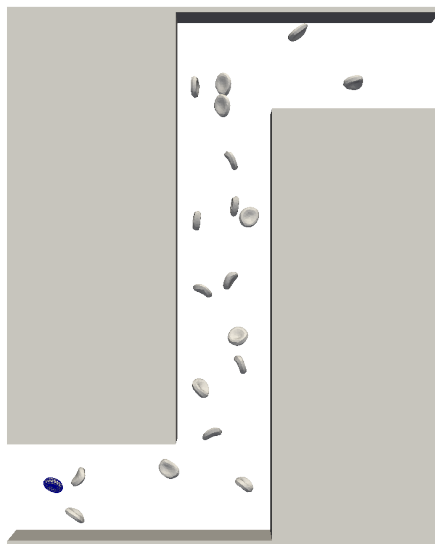
Simulácia sa skladala z niekoľkých krokov: najprv bolo potrebné umiestniť červené krvinky, čiže určiť ich počiatočnú polohu, z ktorej sa začnú vplyvom tekutiny pohybovať. Poloha červenej krvinky, presnejšie poloha jej stredu sa určovala náhodne. Bolo však potrebné zabezpečiť, aby bola v bezpečnej vzdialenosti od stien kanálu a nevznikali tak kolízie.

Ďalším krokom bolo rozprúdenie tekutiny, čo malo za dôsledok rozpohybovanie elastických objektov. [12]

### 4.3 Výsledky simulácie

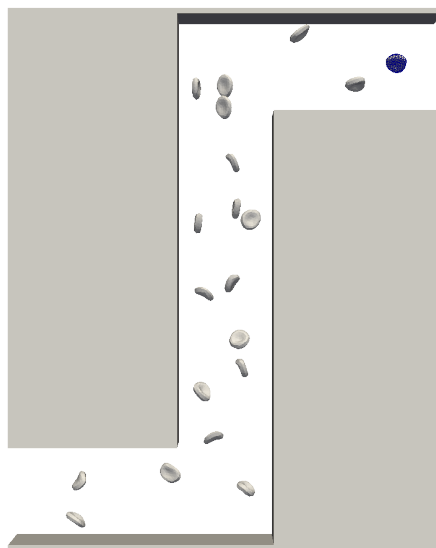
Počas simulácie sme zaznamenávali trajektóriu elastických objektov. Simulácia prebiehala s 24 objektami, ktoré sa cyklicky opakujú. Počet objektov je porovnateľný s objektami zachytenými na jednej snímke z videozáznamu experimentu.

Po prejení kanálom objekt nezaniká ale vráti sa na začiatok kanála a pokračuje vo svojej ceste. Z hľadiska simulácie sa však po prechode na objekt dívame ako na nový objekt; novú červenú krvinku, ktorá prechádza kanálom.



Obr. 4.3: Prechod krvinky koncom kanála

Na obrázku 4.3 je zobrazená krvinka, ktorá prechádza koncom kanálu. Následne sa vráti na jeho začiatok (obrázok 4.4). Je zachovaná jej rýchlosť a poloha vzhľadom na šírku kanálu.



Obr. 4.4: Prechod krvinky začiatkom kanála

V priebehu simulácie sa zapisujú aj trajektórie jednotlivých červených krviniek (trasy). Po prechode z konca kanálu na jeho začiatok je trajektória krvinky zapísaná v novej trase pretože, ako sme uviedli v predchádzajúcom texte, po prechode koncom sa na objekt dívame ako na nový objekt, ktorý prechádza kanálom. Dáta sa zapisujú priebežne najmä preto, aby boli k dispozícii aj keď bude potrebné simuláciu ukončiť skôr, alebo v prípade výskytu chyby, v dôsledku ktorej sa simulácia ukončí.

Súbor popisujúci pohyb elastických objektov obsahuje informácie o počte trás a samotné trasy. Ukážka súboru:

```
<RBC_tracking tracksCount="379">
  <track number="0" particlesCount="25">
    <particle time="0.0" x="89.33333" y="43.0"/>
    <particle time="0.03333" x="89.33333" y="43.0"/>
    <particle time="0.06666" x="90.66666" y="43.33333"/>
    <particle time="0.1" x="90.33333" y="43.66666"/>
    <particle time="0.13333" x="90.0" y="43.0"/>
```

# Kapitola 5

## Vyhodnotenie výsledkov

Na to aby výsledky simulácie mohli nahradiť výsledky reálneho experimentu je potrebné, aby model čo najviac zodpovedal realite. Práve na validáciu (prípadne kalibráciu a optimalizáciu) modelu bude možné použiť výsledky zo spracovania videozáznamov. Ak by výsledky porovnania ukázali, že model krvného toku sa správa rovnako ako reálny tok z experimentu zachyteného na videozázname, môžeme model považovať za validný.

### 5.1 Spôsob porovnania

Pre porovnanie sme mali k dispozícii výstupy z trasovania, trasy detegovaných objektov prechádzajúcich kanálom. Pri hľadaní vhodného spôsobu porovnania výstupov sa ponúkalo niekoľko možností:

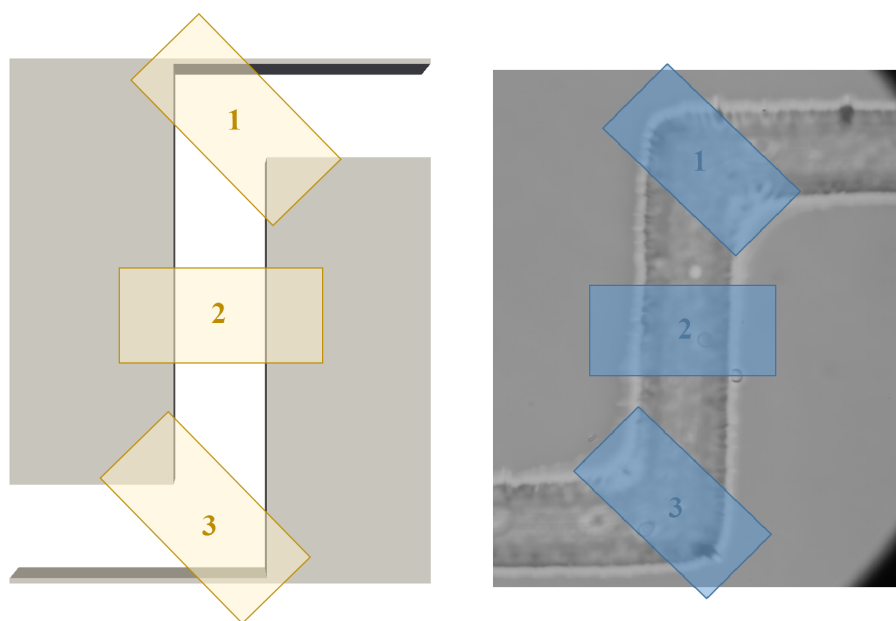
- porovnanie tvarov získaných trás: nielen, že by bolo náročné exaktne vyhodnotiť, či majú dve trasy rovnaký tvar, ale v prípade, že by sme mali vhodný nástroj pre vyhodnotenie tvaru, existovala by veľmi malá pravdepodobnosť, že trasy z experimentu a simulácie budú zhodné. Štartovacia pozícia krvinky v simulácii je náhodná a bolo by teda nesprávne očakávať rovnaké výsledné trasy ako v experimente.
- priemerné rýchlosti krviniek na jednotlivých trasách: toto by mohlo byť vhodným ukazovateľom zhody, keby bola detekcia a trasovanie objektov na vyššej úrovni. Pri pohľade na výsledné trasy z trasovania je viditeľné, že trasy sú kratšie a zachytávajú iba úsek z



celej cesty krvinky kanálom. Je to v dôsledku vzniknutých chýb pri detekcii a trasovaní ale aj z dôvodu nízkej kvality poskytnutých videozáznamov. Okrem toho, trasovanie červených krviniek je oblasť so špecifickými problémami pre spracovanie videozáznamu (popísané v 2). Keďže trasy nie sú rovnako dlhé, porovnávanie početností priemerných rýchlostí rovnakej hodnoty by bolo zavádzajúce.

- rýchlosť objektu v danom mieste kanála: keďže priemerná rýchlosť nebola pre porovnanie vhodná, zvolili sme okamžitú rýchlosť. V rôznych miestach kanálu sme sledovali objekty prechádzajúce cez pomyselné priesečníky a zaznamenávali ich rýchlosť. Rýchlosť sme sledovali v troch oblastiach. Oblasti boli tvorené niekoľkými priesečníkmi, ktoré ležali tesne vedľa seba. Pri prechode trasy jedným z priesečníkov oblasti sa vypočítala okamžitá rýchlosť. Priemerná rýchlosť trasy v danej oblasti sa počítala práve z okamžitých rýchlostí. Tento ukazovateľ sme považovali, za vhodný, pretože na rozdiel od priemernej rýchlosti objektu, okamžitá rýchlosť v danom bode nie je závislá od dĺžky danej trasy.

Oblasti pre simuláciu a videozáznam sú znázornené na obrázku 5.1.



Obr. 5.1: Oblasti

Priesečníky boli určené rovnicou priamky, ktorej tvar sme získali výpočtom. Následne bolo potrebné nájsť tie trasy, ktoré v určitých bodoch pretli priesečníky. Hľadanie trás prebiehalo nasledovne: postupne sa prechádzali všetky trasy a ich body. Body tvoriace danú trasu sa dopĺňali do rovnice prvého priesečníka. Porovnávali sa výsledné hodnoty dvoch po sebe idúcich bodov. Uvedieme príklad: bod  $B1$  a  $B2$  so súradnicami  $(x_1, y_1)$  a  $(x_2, y_2)$  sú dva po sebe idúce body v trase  $T$ . Bod  $B1$  sa nachádza v čase  $t_1$ , bod  $B2$  sa nachádza v čase  $t_2$ . Daná je rovnica priamky (priesečníka):

$$a * x + b * y + c = 0.$$

Po dosadení súradníc bodu, vieme určiť jeho polohu voči priamke. Ak je výsledok

$$\begin{cases} < 0 & \text{bod leží nad priamkou} \\ = 0 & \text{bod leží na priamke} \\ > 0 & \text{bod leží pod priamkou} \end{cases}$$

V prípade, že bod  $B1$  ležal nad priamkou a bod  $B2$  ležal pod priamkou, trasa  $T$  prešla priesečníkom. Okamžitú rýchlosť v pri prechode priesečníkom sme vypočítali ako:

$$\begin{aligned} s &= d(B1, B2) = \sqrt{(x_{B1} - x_{B2})^2 + (y_{B1} - y_{B2})^2} \\ t &= |t_1 - t_2| \\ v &= \frac{s}{t}. \end{aligned}$$

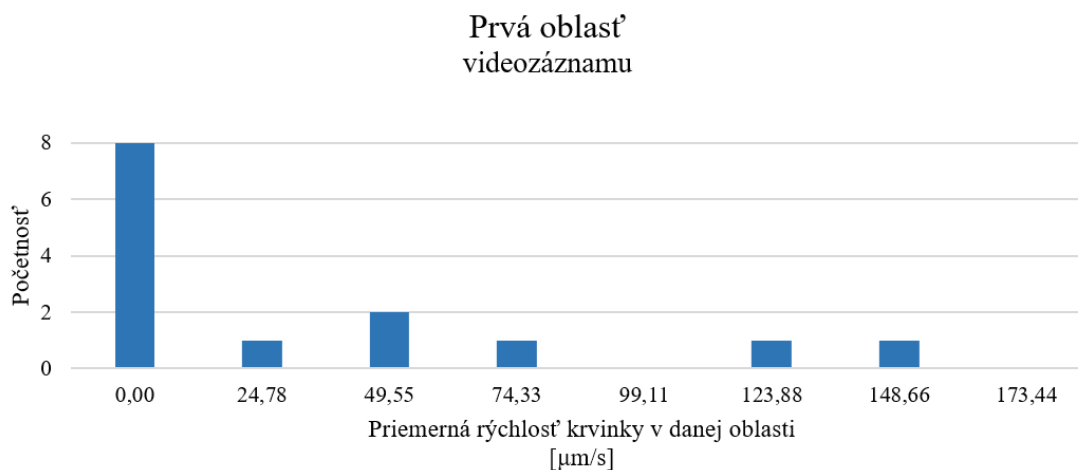
Okamžité rýchlosti pri prechode priesečníkov nám slúžili pre získanie priemernej rýchlosti danej trasy v danej oblasti. Tieto hodnoty slúžili na porovnanie dát získaných zo spracovania videozáznamu a dát získaných zo simulácie. V prípade, že hodnoty priemerných rýchlostí získaných z videozáznamu budú na rovnakých miestach kanálu (v rovnakých oblastiach) zhodné s hodnotami zo simulácie, model dostatočne kopíruje správanie reálneho krvného toku a považujeme ho za validný.

## 5.2 Vyhodnotenie dát z video záznamu

V nasledujúcom texte budú uvedené výsledky pre videozáznam. Výsledky sú vo forme histogramu. Histogram bol vypracovaný pre každú oblasť a zobrazuje priemerné rýchlosti trás v oblasti a ich početnosť. Rýchlosti sú uvedené v mikrometroch za sekundu a vypočítané boli spôsobom opísaným v kapitole 5.1.

### Prvá oblasť

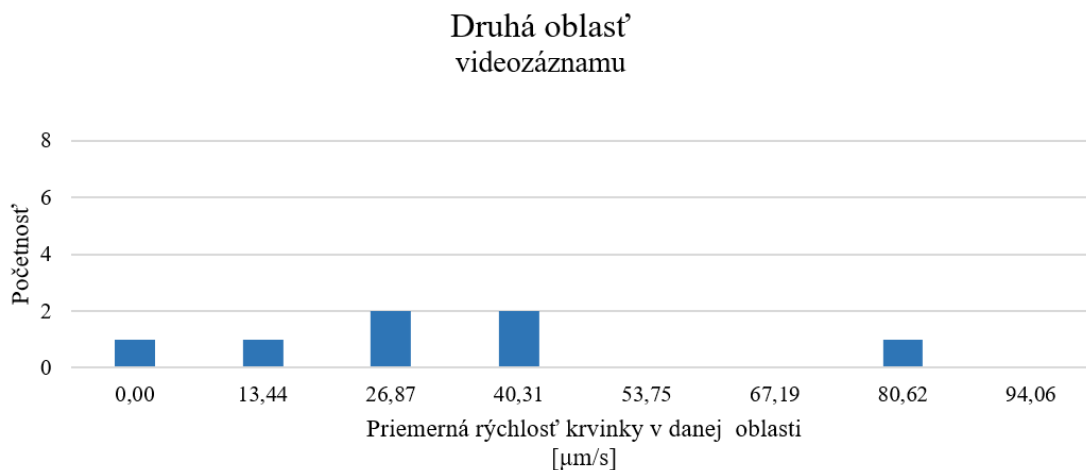
Z celkového počtu trás 644 prešlo danou oblasťou 14. Minimálna priemerná rýchlosť v oblasti bola  $\approx 9,99\mu\text{m/s}$ , maximálna priemerná rýchlosť bola  $\approx 148,66\mu\text{m/s}$ .



Obr. 5.2: Výsledky pre prvú oblasť videozáznamu

## Druhá oblasť

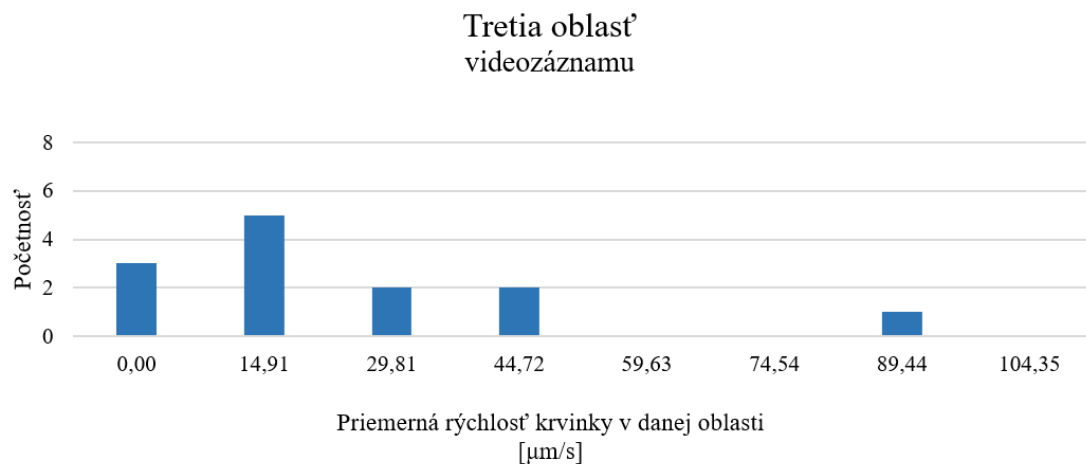
Z celkového počtu trás 644 prešlo danou oblasťou 7. Minimálna priemerná rýchlosť v oblasti bola  $\approx 12,07 \mu\text{m/s}$ , maximálna priemerná rýchlosť bola  $\approx 80,62 \mu\text{m/s}$ .



Obr. 5.3: Výsledky pre druhú oblasť videozáznamu

## Tretia oblasť

Z celkového počtu trás 644 prešlo danou oblasťou 13. Minimálna priemerná rýchlosť v oblasti bola  $\approx 9,99 \mu\text{m/s}$ , maximálna priemerná rýchlosť bola  $\approx 89,44 \mu\text{m/s}$ .



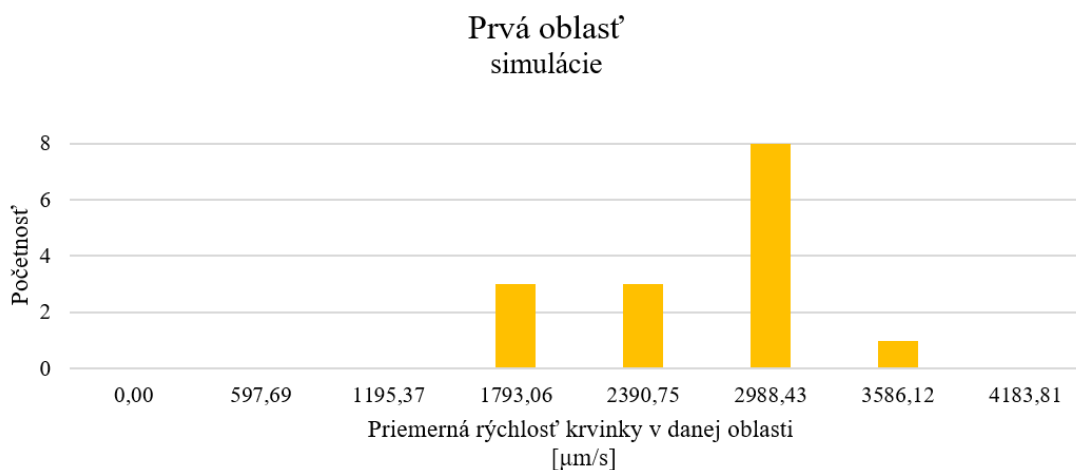
Obr. 5.4: Výsledky pre tretiu oblasť videozáznamu

## 5.3 Vyhodnotenie dát zo simulácie

V nasledujúcom texte budú uvedené výsledky pre simuláciu. Výsledky sú vo forme histogramu. Histogram bol vypracovaný pre každú oblasť a zobrazuje priemerné rýchlosti trás v oblasti a ich početnosť. Rýchlosti sú uvedené v mikrometroch za sekundu a vypočítané boli spôsobom opísaným v kapitole 5.1.

### Prvá oblasť

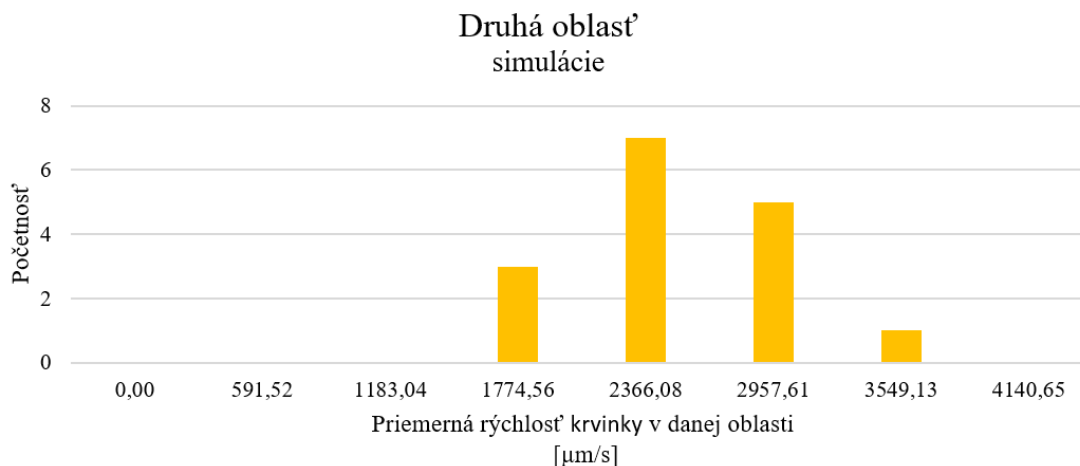
Z celkového počtu trás 37 prešlo danou oblasťou 15. Minimálna priemerná rýchlosť v oblasti bola  $\approx 1873,26 \mu\text{m}/\text{s}$ , maximálna priemerná rýchlosť bola  $\approx 3586,12 \mu\text{m}/\text{s}$ .



Obr. 5.5: Výsledky pre prvú oblasť simulácie

## Druhá oblasť

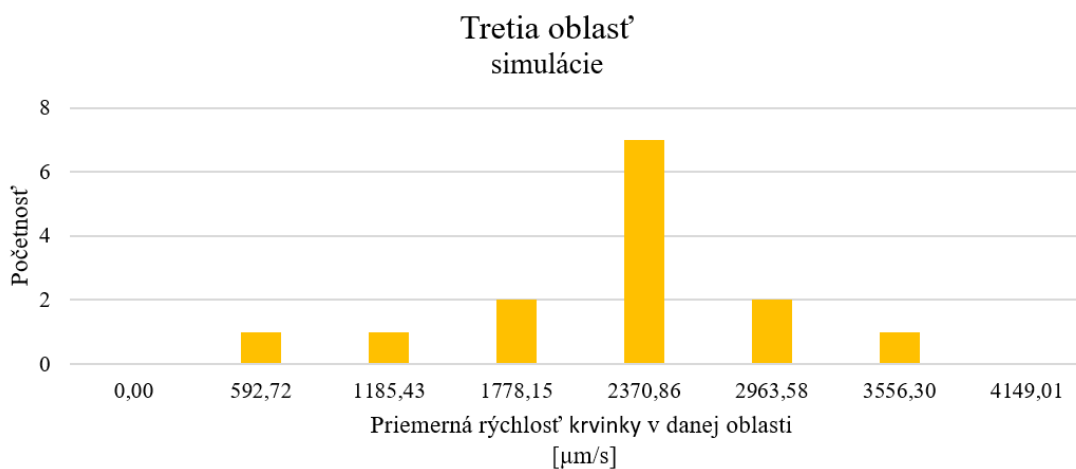
Z celkového počtu trás 37 prešlo danou oblasťou 16. Minimálna priemerná rýchlosť v oblasti bola  $\approx 2089,30 \mu\text{m/s}$ , maximálna priemerná rýchlosť bola  $\approx 3549,12 \mu\text{m/s}$ .



Obr. 5.6: Výsledky pre druhú oblasť simulácie

## Tretia oblasť

Z celkového počtu trás 37 prešlo danou oblasťou 14. Minimálna priemerná rýchlosť v oblasti bola  $\approx 925,68 \mu\text{m/s}$ , maximálna priemerná rýchlosť bola  $\approx 3556,29 \mu\text{m/s}$ .



Obr. 5.7: Výsledky pre tretiu oblasť simulácie

## 5.4 Vyhodnotenie

Pri porovnávaní výsledkov z videozáznamu a simulácie sa rýchlosti rádovo líšia. Nesúlad v rýchlostiach mohol byť zapríčinený viacerými faktormi, ktoré vplývali na výsledky meraní:

- rýchlosť simulačného toku: prvým a najvýraznejším faktorom bolo nastavenie rýchlosti toku simulácie. Tá je v porovnaní s rýchlosťou toku experimentu výrazne vyššia. Pre správne nastavenie rýchlosti toku v simulácii, by bola potrebná kalibrácia modelu. Kalibrácia modelu však nebola cieľom práce.
- experiment: pri vykonávaní experimentu sa objavujú problémy, ktoré ovplyvňujú tok (počas experimentu je náročné regulovať rýchlosť toku, v kanáli vznikajú upchaté miesta).
- chyby trasovania: získané trasy z videozáznamu nie sú úplné. Pre získanie dlhších trás, ktoré by boli porovnateľné s trasami zo simulácie by bolo vhodné rozšíriť algoritmus trasovania tak, aby dokázal riešiť výpadky detekcie. Viď kapitolu 3.8.
- chyby detekcie: výsledky trasovania sú závislé od výsledkov detekcie. V prípade, že sú detegované ako krvinky chybné objekty, alebo krvinky nie je možné detegovať (napr. kvôli nízkej kvalite záznamu), je logické, že ani vzniknuté trasy nemôžu byť správne. Viac informácií o vplyve detekcie na výsledky trasovania bude popísaných v kapitole 5.5.

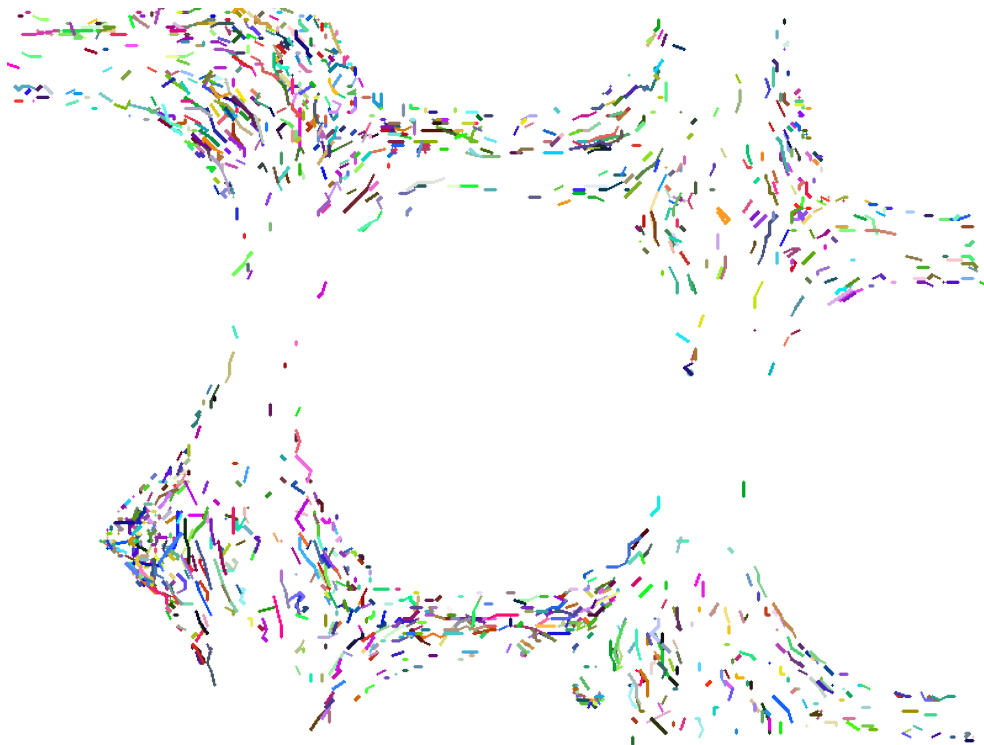
Aj keď sa nám nepodarilo získať rovnaké rýchlosti pre dané oblasti, považujeme trasovanie za úspešné. Pre lepšie výsledky bude potrebná kalibrácia simulačného modelu, aby bolo možné vytvoriť simuláciu zodpovedajúcu biologickému experimentu. Zároveň bude nutné rozšíriť trasovací algoritmus tak, aby riešil výpadky detekcie a bol schopný pospájať trasy zodpovedajúce jednej bunke.

## 5.5 Testovanie

Pri tvorbe a vývoji trasovacieho algoritmu sme pracovali s rôznymi typmi kanálov a viacerými súbormi detegovaných objektov. V nasledujúcom texte budú jednoducho popísané výsledky.

## Rôzne typy kanálov

Okrem videozáznamu popísaného v predchádzajúcich kapitolách (obrázok mikrokanálu 2.1) sme spracovávali aj videozáznam iného experimentu. V experimente bol použitý kanál, v ktorom boli pravidelne usporiadané prekážky. Kvalita tohto videozáznamu však bola veľmi nízka a preto ani výsledky trasovania nie sú najlepšie. Napriek tomu, sa však podarilo záznam spracovať a získať výsledné trasy. Podrobnejšie informácie o tomto type kanálu, spolu s výsledkami meraní sú uvedené v [2]. Trasy sú zobrazené na obrázku 5.8.



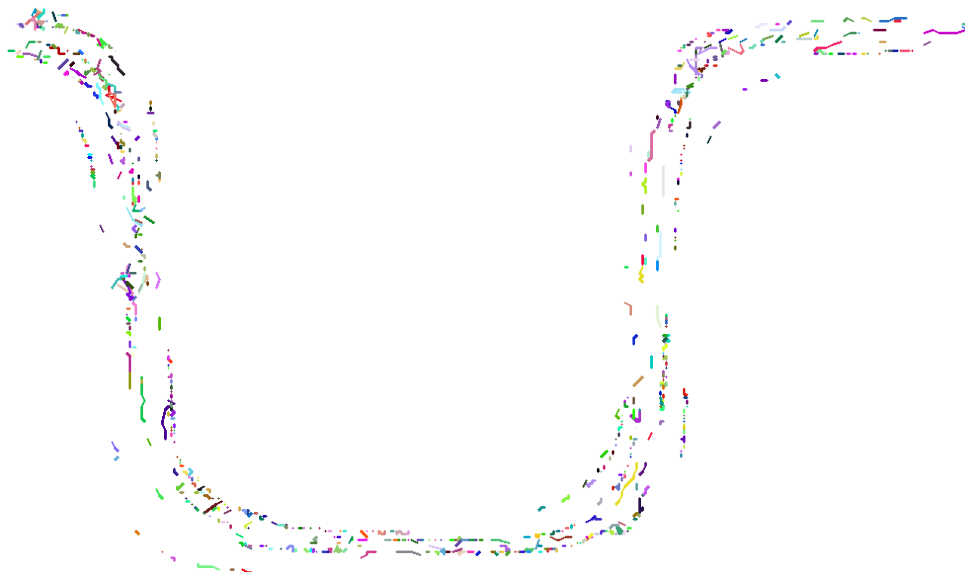
Obr. 5.8: Výsledné trasy pre iný typ kanálu



## Rôzne súbory detegovaných objektov

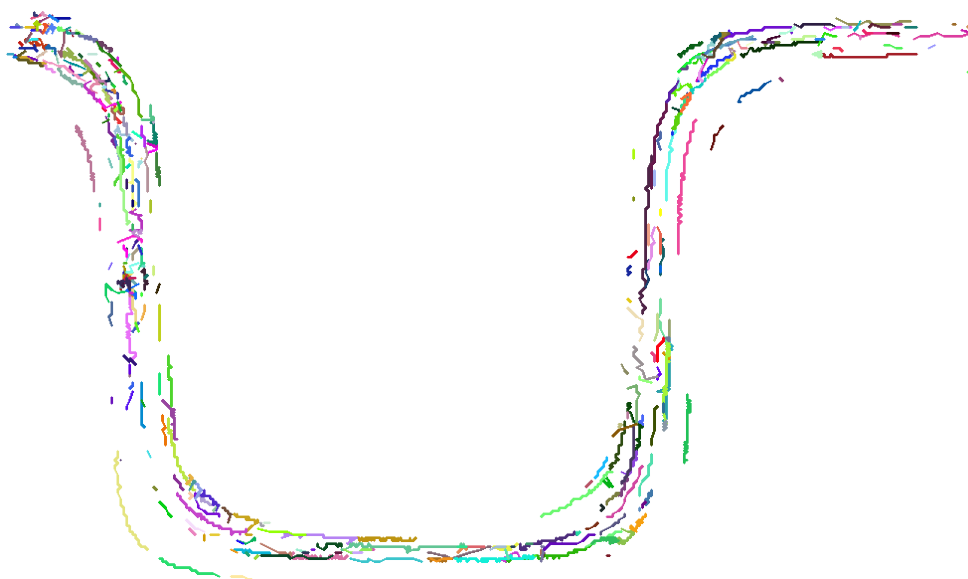
Pri vytváraní výsledných trás objektov je veľmi dôležitá ich detekcia. Výsledné trasy budú pri nastavení rovnakých hodnôt parametrov v metódach trasovania rôzne pre dva odlišné súbory detegovaných objektov.

Na základe súboru detegovaných objektov  $S_1$  prebiehalo trasovanie, výsledky ktorého sú popísané v kapitole 5.2. Na obrázku 5.9 sú zobrazené výsledné trasy. Súbor  $S_1$  tvorili v prevažnej väčšine správne detegované body. Pod pojmom správne detegované body, sú myslené body ktoré detegovali stredy červených krviniek. Nesprávne detegované body nedetegovali stredy červených krviniek a vznikli v dôsledku chýb detekcie. Príčiny chýb detekcie sú popísané v [2]. Celkový počet detegovaných bodov bol 2849.



Obr. 5.9: Získané trasy zo súboru  $S_1$

V súbore  $S_2$  sa nachádzal veľký počet nesprávne detegovaných bodov. Hoci je nejaký bod  $B$  nesprávny, pretože nedeteguje skutočný stred červenej krvinky, od skutočných stredov nie je veľmi vzdialený. V prípade, že bod  $B$  bude priradený do nejakej trasy, trasa už nemusí presne kopírovať pohyb detegovaného objektu ale odchýlka bude minimálna. Výsledné trasy, ktoré vznikli s použitím súboru  $S_2$  sú zobrazené na obrázku 5.10. Celkový počet detegovaných bodov bol 5213.



Obr. 5.10: Získané trasy zo súboru  $S_2$

Je preto potrebné zvážiť, či neexistujú situácie, kedy je výhodnejšie pracovať radšej s väčším počtom bodov (i keď časť z nich nie je presná) s cieľom získania dlhších trás. Získané trasy síce nebudú presné trasy sledovaných objektov, ale budú im svojimi vlastnosťami podobné, čo môže byť v niektorých prípadoch dostačujúce.

# Záver

Cieľom práce bolo spracovať a analyzovať videosekvencie zo záznamov biologických experimentov obsahujúce pohyb buniek. Sledovala sa poloha buniek a ich rýchlosť. Následne sa dáta použili na porovnanie experimentov s počítačovou simuláciou v Object-in-fluid module vedeckého softvérového balíka ESPResSo.

V úvode práce sme opísali význam trasovania objektov, aj vysvetlili pojmy súviace so spracovaním videozáznamov a trasovaním. Následne bol opísaný priebeh biologického experimentu a spôsob akým poskytnuté videozáznamy vznikajú. Rovnako boli opísané aj špecifické problémy pri spracovaní videozáznamov biologických experimentov (nízka kvalita záznamu, vlastnosti červených krviniek).

Pri analýze v súčasnosti dostupných nástrojov pre detekciu a trasovanie objektov sme dospeli k názoru, že je potrebný vývoj vlastného nástroja. Ten by dokázal riešiť problémy typické pre túto oblasť. Nástroj je implementovaný v jazyku Python a pre prácu s obrazom využíva knižnicu OpenCV.

Nástroj vytvára trasy sledovaných objektov v niekoľkých krokoch. Najprv sú vytvorené prvotné trasy, ktoré síce nie sú dlhé, ale považujeme ich za správnu reprezentáciu pohybu sledovaných objektov. Následne sa spájajú prvotné trasy s cieľom vytvorenia nových, dlhších trás. Po spájaní trás je možné vzniknuté trasy rozšíriť o body, ktoré neboli použité pri vytváraní prvotných trás. Oba spôsoby rozširovania trás (doplňanie bodov či spájanie trás) je možné viacnásobne opakovať. Parametre používané pri vytváraní trás sú vždy vstupom od užívateľa. Je preto nutné aby mal užívateľ čo najviac informácií o spôsobe akým videozáznam vznikol aj o sledovaných objektoch. Výsledné trasy sú totiž závislé od hodnôt parametrov.

Rovnako sú výsledky trasovania závislé od výsledkov detekcie, ktoré sú vstupom pre proces

trasovania. Detekcia je v práci opísaná len nárožne pre lepšiu predstavu čitateľa.

Po získaní trás z videozáznamu sme mohli pristúpiť k meraniu priemerných rýchlostí krviniek v daných oblastiach kanálu. Výsledky sme porovnali s výsledkami meraní zo simulácie. Pre simuláciu biologického experimentu bolo potrebné vytvoriť model kanálu a následne zaznamenávať pohyb elastických objektov prúdiacich v modeli krvného toku. Elastické objekty predstavovali červené krvinky. Vytvorenie simulačného modelu a simulácie experimentu by nebolo možné bez spolupráce s výskumnou skupinou cell-in-fluid. Výsledky meraní sú uvedené v poslednej kapitole.

Na základe výsledkov meraní sme zistili, že rýchlosti sa výrazne líšia. Odchýlky však pripisujeme výpočtovému (simulačnému) experimentu, ktorý sa ukázal ako nie zhodný s biologickým experimentom. Pre lepšie výsledky je potrebná jeho kalibrácia, čo však nebolo cieľom práce.

Proces trasovania považujeme za správny, vytvorené trasy kopírovali tvar kanálu a pohyb sledovaných objektov. Je možné, že výsledky trasovania by boli lepšie v prípade videozáznamov s vyššou kvalitou. Kvalita poskytnutých záznamov bolo nízka a neumožňovala kvalitnú detekciu. Keďže sa nepodarilo vždy detegovať všetky objekty na snímke, trasovanie nie je optimálne. Vzniknuté trasy sú pomerne krátke a neúplné. V prípade, že sa v budúcnosti zlepší kvalita videozáznamov, alebo sa bude implementovať rozšírenie trasovacieho algoritmu opísané v 3.8, je veľmi pravdepodobné že výsledky trasovania sa budú viac blížiť k reálnemu pohybu sledovaných objektov.

# Literatúra

- [1] POTÚČEK I.; Diplomová práca, Vysoké učení technické v Brně, Brno, 2002.
- [2] MUČKA F.; *Algoritmy a ich implementácia pre analýzu a získavanie dát zo záznamov biologických experimentov*, Diplomová práca, Žilinská univerzita v Žiline, Žilina, 2017.
- [3] Nástroj Track Mate;  
[http://imagej.net/Getting\\_started\\_with\\_TrackMate](http://imagej.net/Getting_started_with_TrackMate); [cit. 1.4.2017]
- [4] LAP tracker;  
[http://imagej.net/TrackMate\\_Algorithms#Cost\\_calculation\\_.26\\_Brownian\\_motion](http://imagej.net/TrackMate_Algorithms#Cost_calculation_.26_Brownian_motion); [cit. 4s.4.2017]
- [5] Knižnica OpenCV;  
<http://opencv.org/>; [cit. 1.4.2017]
- [6] SimpleCV;  
<http://simplecv.org/>; [cit. 7.4.2017]
- [7] Jazyk Python;  
<https://www.python.org/>; [cit. 1.4.2017]
- [8] Tutorialy pre prácu s OpenCV;  
[http://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_tutorials.html](http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html);  
[cit. 1.4.2017]
- [9] Vývojové prostredie PyCharm  
<https://www.jetbrains.com/pycharm/>; [cit. 1.4.2017]

- [10] Funkcie knižnice OpenCV pre vykresľovanie;  
[http://docs.opencv.org/3.0-beta/modules/imgproc/doc/drawing\\_functions.html](http://docs.opencv.org/3.0-beta/modules/imgproc/doc/drawing_functions.html); [cit. 1.4.2017]
- [11] Domovská stránka výskumnej skupiny Cell-in-fluid;  
<http://cell-in-fluid.fri.uniza.sk/en/home>; [cit. 1.4.2017]
- [12] Object-in-fluid Documentation, September 2014.
- [13] Jazyk TCL;  
<https://www.tcl.tk/>; [cit. 1.4.2017]
- [14] ESPResSo,  
<http://espressomd.org/wordpress/>; [cit. 17.3.2017]
- [15] Tóthová R.; *Modelovanie pohybu červených krviniek v mikrofluidických zariadeniach*, 8th Workshop on Intelligent and Knowledge oriented Technologies, Technical University in Košice, 2013, ISBN 978-80-8143-128-9.
- [16] Gusenbauer M.; Mazza G.; Brandl M.; Schrefl T.; Tóthová R.; Jančigová I.; Cimrák I.; *Cell rheology in microfluidic perfusion: computational and experimental approach*, Micro and nano engineering, 42nd international conference : September 19-23, 2016, Vienna, Austria. 2016.
- [17] TOMÁŠIKOVÁ J.; *Vplyv triangulácie na vznikmodelovanie vzniku červených krviniek pomocou softvéru ESPResSo*, Bakalárska práca, Žilinská univerzita v Žiline, Žilina, 2015.
- [18] MÁRTON P.; ADAMKO N.; *Praktický úvod do modelovania a simulácie*, Edis, Žilina, 2011, ISBN 978-80-554-0387-8.

# **Dodatok A**

## **Obsah priloženého CD**

**Zdrojové súbory** Zdrojové súbory vytvorenej aplikácie

**Obrázky** Elektronická podoba všetkých použitých ilustrácií

**Práca** Elektronická podoba diplomovej práce