



ИНСТИТУТ ЗА МАТЕМАТИКУ И ИНФОРМАТИКУ  
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ  
УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ

МАСТЕР РАД

---

**Примена алгоритама препознавања облика у  
неутронској дозиметрији**

---

*Студент:*  
Никола Николић

*Професор:*  
др Милош Ивановић

Септембар 2016.

# Садржај

<b>1 Увод у неутронску дозимертију</b>	<b>3</b>
1.1 Атоми и зрачење . . . . .	3
1.1.1 Типови зрачења . . . . .	4
1.1.2 Извори, примена и ефекти зрачења . . . . .	5
1.2 Неутронско зрачење . . . . .	5
1.2.1 Примена неутронског зрачења . . . . .	6
1.2.2 Дозиметрија . . . . .	6
1.3 Дефиниција проблема . . . . .	7
<b>2 Обрада слика употребом библиотеке <i>OpenCV</i></b>	<b>9</b>
2.1 Сиво скалирана слика . . . . .	10
2.2 Сегментација слике . . . . .	11
2.2.1 Раздвајање трагова од позадине . . . . .	12
2.2.2 Проналажење трагова . . . . .	13
2.2.3 Сегментација трагова . . . . .	15
2.3 Исправљање ротације слике у односу на претходну слику из секвенце . . . . .	16
2.3.1 Детекција интересних тачака . . . . .	16
2.3.2 Рачунање описног вектора . . . . .	17
2.3.3 Поређење описних вектора и одређивање трансформације . . . . .	17
<b>3 Развој корисничког интерфејса употребом <i>Qt</i> оквира</b>	<b>20</b>
3.1 Развојни алати . . . . .	21
3.2 Модули . . . . .	21
3.3 Сигнали и слотови . . . . .	22
<b>4 Софтвер за детекцију трагова</b>	<b>24</b>
4.1 Структура пројекта . . . . .	25
4.2 Начин употребе . . . . .	26
4.2.1 Управљање пројектима . . . . .	27
4.2.2 Приказ и опције приказа детектованих трагова . . . . .	28
4.2.3 Приказ и извоз резултата обраде . . . . .	30
<b>5 Анализа</b>	<b>34</b>
5.1 Резултати . . . . .	35
5.2 Најчешће грешке . . . . .	37
5.3 Дискусија . . . . .	40
<b>6 Закључак</b>	<b>41</b>

# Сажетак

Дозиметрија се бави одређивањем апсорбоване дозе јонизујућег зрачења у разним материјалима, укључујући и људско ткиво. Карактеристике тренутно доступних личних неутронских дозиметара су далеко од идеалних, а траг детектори се данас сматрају једном од најперспективнијих техника. Међу њима је најпознатији *CR-39* детектор. При интеракцији упадних неутрона са детектором настају трагови који нагризањем материјала базом постaju видљиви. На основу тих трагова врши се процена дозе зрачења.

Мерење густине трагова врши се визуелном проценом броја трагова по јединици површине помоћу микроскопа. Тада процес може да траје и по више сати. У овом раду ће се описати примена алгоритама обраде слика (*image processing*) која овај процес може делимично аутоматизовати и значајно убрзати.

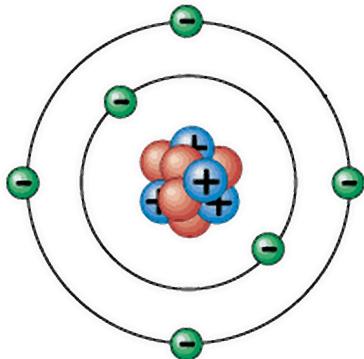
У првом поглављу дат је увод у неутронску дозиметрију, њену примену и дефинисан је проблем којим ће се бавити овај рад. Друго поглавље представља *OpenCV* библиотеку и описује коришћене алгоритме за обраду слика. Треће поглавље укратко представља *Qt* платформу за развој корисничког интерфејса и његове главне предности. У четвртом поглављу описан је софтвер за детекцију трагова неутронске дозиметрије. Дат је опис структуре пројекта као и начин употребе развијеног софтвера. Даље, у петом поглављу дата је анализа софтвера, процена тачности детекције трагова, као и анализа најчешћих грешака детекције. У последњем поглављу дат је предлог могућих унапређења апликације.

# Глава 1

## Увод у неутронску дозимерију

### 1.1 Атоми и зрачење

Атом је основна јединица материје. Атоми су изграђени од позитивно наелектрисаног атомског језгра и одређеног броја електрона. Електрони носе негативно наелектрисање и крећу се у електронском омотачу око језгра. Атомско језгро садржи протоне, који имају позитивно наелектрисање, једнако негативном наелектрисању електрона и неутроне који немају наелектрисање.



Слика 1.1: Шематски приказ структуре атома

Хемијске особине атома одређује број протона у језгру. Сваки атом садржи једнак број протона и електрона и зато је електрично неутралан. Атом постаје наелектрисан тако што прими или отпушти један или више електрона и постаје јон. Такав процес назива се јонизација.

Како хемијске особине атома не зависе од броја неутрона, постоје атоми истог елемента са различитим бројем неутрона - тзв. изотопи. Атомска језгра са неповољним односом броја протона и броја неутрона су нестабилна и путем радиоактивног распада прелазе у стабилније стање.

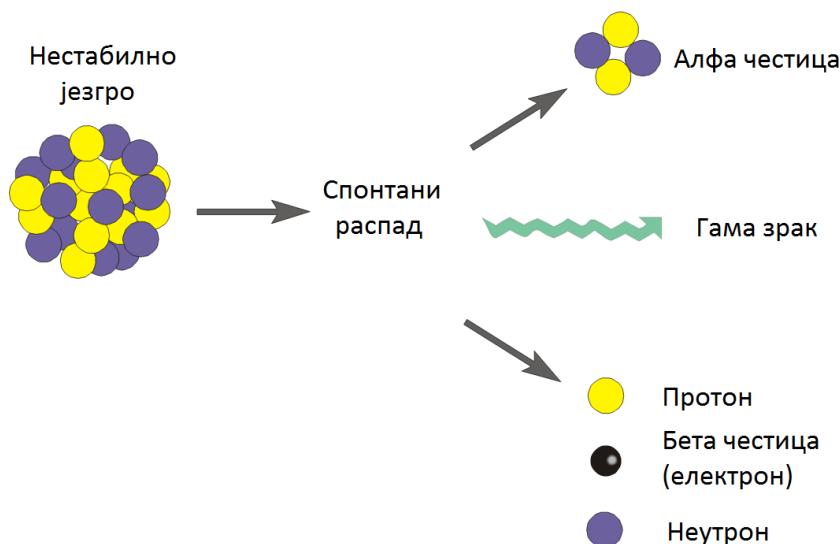
Радиоактивност је спонтани процес у којем се атомско језгро, емитујући једну или више честица преображава у друго језгро. При трансформацији језгра, мења се састав или енергетско стање. Радиоактивно зрачење продире кроз различите материјале, а такође може и да јонизује средину кроз коју пролази. На тај начин може да изазове оштећење ћелија живих организама. Настали јони нарушују биохемијске процесе у ћелијама, што може довести до разних поремећаја у њиховом функционисању и деоби, па и до настанка озбиљних болести.

### 1.1.1 Типови зрачења

Према ефектима које производи на материју, зрачење се може класификовати на јонизујуће и нејонизујуће зрачење.

Јонизујуће зрачење укључује космичке зраке,  $X$  зраке и зрачење од радиоактивних материјала. Заједничко својство је висока енергија која омогућава јонизацију атома средине кроз коју зрачење пролази. Како долази до јонизације атома и молекула који су до тада били неутрални, настају оштећења материјала. Јонизирајуће зрачење спада у најопасније, оно је за око невидљиво, не осећа се код контаминације, тешко се детектује и постоји јако узак број терапија које могу помоћи. Три основна типа јонизујућег зрачења су  $\alpha$ ,  $\beta$  и  $\gamma$  зрачење.

Нејонизујуће зрачење укључује ултраљубично светло, топлотно зрачење, радио таласе и слично. Не поседује енергију као јонизујуће зрачење и обично се сматра безопасним при малим дозама које не изазивају пораст температуре.



Слика 1.2: Честице које могу настати при радиоактивном распаду

- **Алфа зрачење ( $\alpha$ ).** Алфа честица је позитивно наелектрисано језгро хелијума (језгро са два протона и два неутрона) емитовано од стране нестабилног атомског језгра. Алфа честица је релативно масивна, али има кратак домет у ваздуху, свега 3-5 cm. Алфа честице могу апсорбовати кожа или папир.
- **Бета зрачење ( $\beta$ ).** Бета честице су високоенергетски електрони или позитрони. Електрони настају распадом језгра са вишком неутрона, а позитрони настају при распаду језгра са вишком протона у односу на њихов оптимални број за стабилност језгра. Бета честице су много мање од  $\alpha$  честица и могу продрети даље у материјал или ткиво. Бета зрачење се може апсорбовати материјалима од пластике, стакла или метала.
- **Гама зрачење ( $\gamma$ )** је врло високи енергетски фотон (форма електромагнетне радијације врло кратке таласне дужине), емитован од стране нестабилног језгра које често у исто време емитује и  $\alpha$  и  $\beta$  честице. Гама зрачење узрокује јонизацију у атомима када пролази кроз материјал, првенствено због интеракције са електронима. У одсуству густих материјала,  $\gamma$  зрачење може оштетити материјал на који делује.

- **Неутронско зрачење** ( $n$ ) је емитовање неутрона из нестабилног језгра. Ван језгра неутрони су нестабилни и имају време полураспада око 11,5 минута. Када су у реакцији са материјалом изазивају емисију  $\beta$  и  $\gamma$  зрачења. Неутронско зрачење захтева значајну заштиту да би се редуковало излагање.

### 1.1.2 Извори, примена и ефекти зрачења

Изворе зрачења делимо на **природне** и **вештачке**. Природном зрачењу изложени смо услед зрачење Сунца и космоса, радиоактивних елемената у тлу, објектима у којима живимо, преко хране и пића које уносимо у тело. Количина овог неизбежног природног зрачења се разликује од места до места на Земљи.

Ефекти зрачења на људско здравље од природних извора нису изразито негативни, јер се оно никад не сакупља у телу. Природно зрачење је стално и веома слабо. Биолошки механизам је еволутивно прилагођен на ову врсту зрачења.

Вештачки извори радијације на Земљи премашују 5% укупног зрачења. Ови извори зрачења су они које је човек изградио:

- **Нуклеарне електране** имају за циљ да произведу електричну енергију. Највећу и сталну потенцијалну опасност у раду нуклеарних електрана чине евентуални инциденти при којима се емитују велике количине радиоактивног материјала у атмосферу и земљиште.
- **Медицински извори** зрачења настају при лечењу неких болести услед употребе апарате који емитују радиоактивно зрачење. Такође, истрошени апарати и опрема остају као проблем у виду радиоактивног отпада.
- **Технички извори** зрачења се односе на разне апарате који функционишу на бази радионуклеида, те представљају опасност за животну средину (радиоактивни громобрани, разни апарати у научно-истраживачким институцијама).

Јонизујуће зрачење делује штетно на биолошке системе и може да доведе до појаве функционалних, морфолошких и генетских промена, а уколико су дозе којима је особа изложена високе може да дође и до смрти. Нису сви људи подједнако осетљиви на зрачење, као ни сва ткива и органи у организму.

Акутни ефекти се јављају непосредно после озрачивања особе код изложености високим дозама радијације. Јачина ефекта зависи од примљене дозе, врсте зрачења и осетљивости озрачених ткива. Код акутне означености јавља се радијациона болест.

Код хроничне изложености радиоактивном зрачењу, обично код професионално изложеног особа, долази до оштећења ткива, и то најчешће крви и хематопоетских органа. Уколико се реагује на време, најчешће долази до опоравка.

## 1.2 Неутронско зрачење

Неутрони су ненаелектрисани и не интереагују са електронима, па могу да прођу знатно растојање у материји без икакве интеракције. Понашање неутрона је слично  $\gamma$  зрачењу, и оно припада групи индиректно јонизујућих зрачења. Међутим, неутрони не могу директно да јонизују материју, већ то чине искључиво преко секундарно произведених честица.

Електромагнетска интеракција неутрона са електронима је занемарљива. Са друге стране, пролазећи кроз материју, неутрони се могу расејати еластично или нееластично на атомским језгрима. Расејање је еластично ако се укупна кинетичка енергија у судару очува, тј. када је енергија коју неутрон изгуби једнака кинетичкој енергији узмака језгра. Расејање је нееластично када језгро апсорбује извесну количину енергије и пређе у више енергетско стање. Такође, неутрон може бити захваћен или апсорбован од стране језгра,

што доводи до промене броја неутрона захваћеног језгра, па самим тим и његових физичких и хемијских својстава.

За брзе неутроне је типично да највише губе енергију у материји преко еластичних расејања. Овај процес се назива модерација неутрона и један је од најважнијих процеса у нуклеарним реакторима. Како енергија неутрона опада, расејање се наставља, али генерално, расте вероватноћа захвата на језгру. Дифузија термалних неутрона је такође узрокована низом еластичних расејања.

Еластично расејање неутрона се доста користи за регистровање брзих неутрона, проучавајући узмакнута језгра (углавном узмакнуте протоне) различитим инструментима. Овај тип расејања се такође користи за регистровање узмакнутих језгара помоћу јонизационих метода.

### 1.2.1 Примена неутронског зрачења

Најважније примене неутронских извора су заступљене у индустрији, у науци и истраживању, као и у медицини:

- **Геофизичка мерења:** мапирање и анализа рудника минерала, нафтна истраживања, мапирање и анализа каменолома, истраживање урана.
- **Контрола индустријских процеса:** контрола цементног процеса, испитивање квалитета угља, испитивање дебљине зида.
- **Медицина:** мерење састава тела, терапија канцера, студије дијете и исхране.
- **Безбедност:** детекција и идентификација експлозива, детекција и идентификација хемијског оружја, детекција и идентификација специјалних нуклеарних материјала, детекција нагазних мина.
- **Општа истраживања:** референтни извор брзих неутрона за инструментацију, калибрациони извор за инструментацију за посматрање неутрина, проучавање осетљивости електронских компоненти на радијацију, испитивање нуклеарног реактора, неутронска радиографија.
- **Еколођија:** испитивање нуклеарног отпада, испитивање отпада за очување ресурса и поступак опоравка, квантификација угљеника у земљишту.

### 1.2.2 Дозиметрија

Дозиметрија [2] се бави одређивањем дозе јонизирајућег зрачења, првенствено у људском ткиву, али и у разним другим материјалима. Неутронска дозиметрија је од велике важности за заштиту од зрачења у близини акцелератора честица и нуклеарних реактора, за квантификацију ефекта излагања космичком зрачењу, као и код терапије брзим и спорим неутронима.

Иако је развој нуклеарне физике и физике елементарних честица унео у употребу много врста детектора, они се заснивају на истом принципу: предаја дела или целокупне енергије зрачења детекторској маси, где се она преводи у неку другу форму енергије која је погодна за људско опажање.

За детекцију, неутралне честице најпре морају проћи кроз неку врсту реакције у детектору, чији ће производ бити наелектрисане честице које јонизују или ексцитују атоме материјала детектора. Облик у коме се појављује енергија зависи од детектора и његове конструкције. На пример, гасни детектори директно прикупљају јонизационе електроне за формирање струјног сигнала, док у сцинтилаторима ексцитација и јонизација доприносе појави молекуларних прелаза, чији је крајњи резултат емисија светlostи. У фотографским

емулзијама, јонизација изазива хемијске реакције које омогућују формирање латентне слике трага.

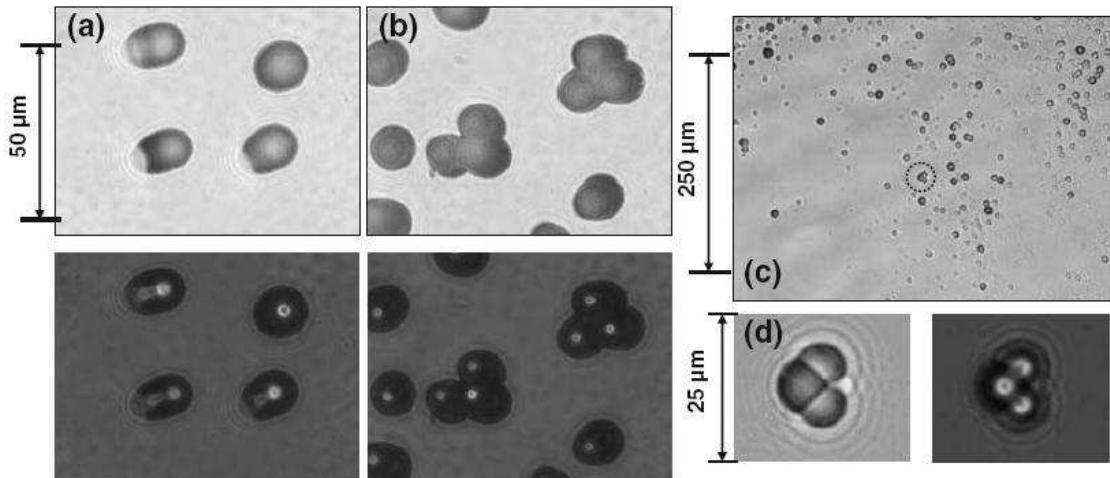
Савремени детектори су претежно електричне природе. У неком тренутку, информација из детектора се претвара у електрични сигнал који је погодан за електронску обраду. Такви детектори користе активне методе за детекцију и дозиметрију неутрона.

Поред активних, користе се и пасивне методе, односно методе које не захтевају директно напајање енергијом у току излагања. Међу данас коришћеним пасивним личним неутронским дозиметрима, траг детектори се сматрају једном од најперспективнијих техника. Посебно место у свету траг детектора заузима *CR-39* детектор.

### ***CR-39* детектор**

Међу бројним предностима *CR-39* детектора у односу на друге мерне технике су велика осетљивост на протоне, нискоенергетски праг за неутроне, као и неосетљивост на Гама зрачење. Због тога је *CR-39* најпогоднији кандидат за примену у личној неутронској дозиметрији.

При расејању неутрона на атомима *CR-39* детектора ( $C_{12}H_{18}O_7$ ) производе се узмакнута језгра H, C и O, док се у нуклеарним реакцијама стварају секундарне наелектрисане честице ( $\alpha$  честице, протони, деутерони...). Секундарне наелектрисане честице и узмакнута језгра остављају оштећења при проласку кроз детекторски материјал, која се називају "латентни трагови". Попречне димензије латентних трагова су до 10 nm, тако да се могу видети само под електронским микроскопом. Траг се може визуелизовати (учинити видљивим) под оптичким микроскопом, ако се делује агресивним хемијским агенсом, као што је на пример, водени раствор *NaOH* или *KOH*. Услед веће хемијске активности, раствор нагриза оштећени део више него неоштећени, тако да се латентни траг знатно увећава и може се посматрати обичним оптичким микроскопом. Примери нагрижених трагова дати су на слици 1.3.



Слика 1.3: Трагови настали озрачивањем *CR-39* детектора

### **1.3 Дефиниција проблема**

Мерење густине трагова насталих траг детектором *CR-39* визуелном проценом броја трагова по јединици површине помоћу оптичког микроскопа траје незанемарљиво дugo.

Још један од изазова је и опис развоја трага, односно раст трагова. Ови процеси могу да трају и по више сати.

У складу са поменутим, основна хипотеза Мастер рада је да се овај посао може аутоматизовати и значајно убрзати применом алгоритама за обраду слика. Као основни програмерски алати користе се решења отвореног кода: језик *C++* спретнат са библиотеком *OpenCV* (*Open Source Computer Vision Library*), као и *Qt* платформа за израду корисничког интерфејса.

## Глава 2

# Обрада слика употребом библиотеке *OpenCV*

*OpenCV (Open Source Computer Vision Library)* [3] је библиотека отвореног кода, која садржи имплементације више стотина алгоритама рачунарског вида (*computer vision*). Библиотека је написана у програмском језику *C++*, али подржава и интерфејсе ка програмским језицима *C*, *Python* и *MATLAB*, а у развоју су и интерфејси за *CUDA* и *OpenCL* платформе. Подржана је на свим водећим оперативним системима: *Windows*, *Linux*, *Android* и *Mac OS X*. Библиотека је објављена под *BSD* лиценцом, те је стога погодна и за академску и за комерцијалну употребу.

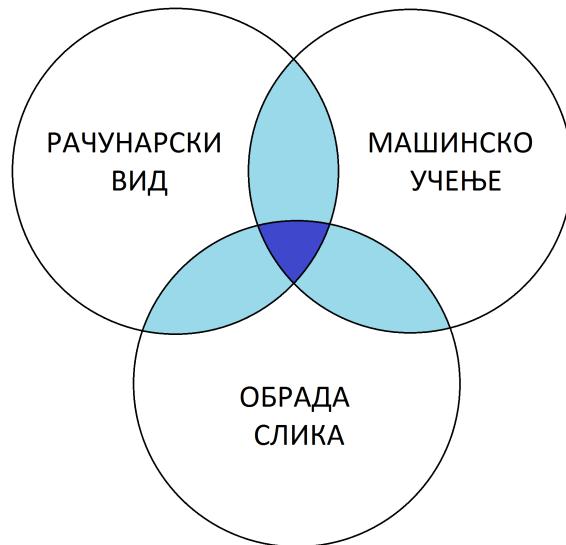
Имплементирани алгоритми се могу користити за препознавање облика, детекцију и препознавање лица, праћење покрета при видео снимку, спајање више слика у једну, препознавање маркера за проширену стварност и слично. Такви алгоритми су примењени у бројним реалним апликацијама, попут програма за видео надзор, навигацију и аутоматизацију рада робота, проверу призвода у фабрикама, асистенцију при вожњи аутомобила итд.

Рачунарски вид помаже у прикупљању релевантних информација са слика и доношењу одлука базираних на тим подацима. Циљ рачунарског вида је да омогући да рачунар посматра ствари на исти начин као и људи. Основни кораци система базираног на рачунарском виду су:

- прикупање слика,
- манипулација slikama,
- извлачење релевантних информација и
- доношење одлука.

Као што се из наведеног види, за један такав систем веома су важни и алгоритми машинсог учења (*machine learning*) као и алгоритми обраде слика (*image processing*). Библиотека *OpenCV* садржи такве алгоритме. Компонента која нас занима је управо обрада слика. Обрада слика је процес манипулације податима слике у сврху прикупљања релевантних информација.

Софтвер, који је резултат овога рада, има задатак да са слике материјала неутронске дозиметрије изврши процену броја трагова зрачења. Кораци обраде слике и процене броја трагова дати су у наставку.



Слика 2.1: Однос рачунарског вида са машинским учењем и обрадом слика

## 2.1 Сиво скалирана слика

Први корак за већину алгоритама обрада слика је рачунање сиво скалиране слике на основу оригиналне слике, тј. слике која садржи црвену, зелену и плаву компоненту. У даљем процесу користи се само сиво скалирана слика. На тај начин постиже се значајна уштеда у меморији, јер слика која се обрађује може бити интерно копирана виш пута. Самим тим постижу се боље перформансе, а добија се и на једноставности алгоритама.

Листинг 2.1: Рачунање сиво скалиране слике

```

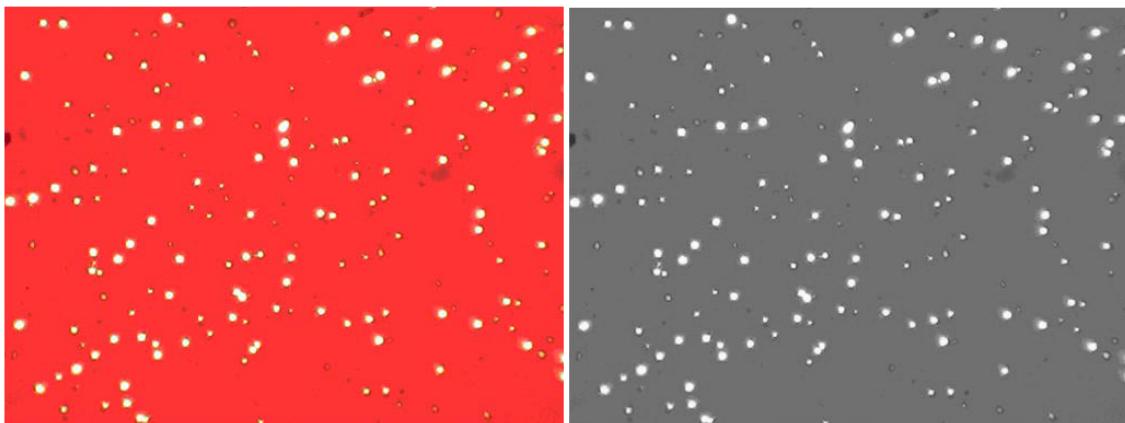
1 // Load BGR image.
2 Mat bgr = imread(path, CV_LOAD_IMAGE_COLOR);
3
4 // Convert image to grayscale.
5 Mat grayscale;
6 cvtColor(bgr, grayscale, CV_BGR2GRAY);

```

Сиво скалирана слика рачуна се као:

$$Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \quad (2.1)$$

где су  $Y$  - сиво скалирана слика,  $R$  - црвена компонента оригиналне слике,  $G$  - зелена компонента оригиналне слике и  $B$  - плава компонента оригиналне слике. Коефицијенти представљају измерену перцепцију интензитета код трохроматских људи. Конкретно, људски вид је најосетљивији на зелену, а најмање на плаву боју.



Слика 2.2: Оригинална (лево) и сиво скалирана (десно) слика

## 2.2 Сегментација слике

Један од основних циљева обраде слика је идентификација објеката који се појављују на дигиталној слици. Задатак сегментације слике је да произведе поједностављене информације за анализу. Сегментација пружа информације о позицији и изгледу објекта.

Постоје различите методе сегментације, а избор конкретне методе се врши на основу примене. Неке од најчешћих метода сегментације су:

- **Сегментација прагом** (*Thresholding*) - подразумева раздвајање објеката на основу боје пиксела. У зависности од боје, пиксел припада одређеној групи објеката. Уколико се врши сегментација са једним прагом, метода се назива бинаризација. Бинаризација се најчешће користи за раздвајање објеката од позадине. Ово је најједноставнији и најбржи метод сегментације. Сегментација прагом се примењује у алгоритмима препознавања текста за раздвајање текста од позадине.
- **Сегментација кластерирањем** (*Clustering*) - подразумева груписање пиксела сличних карактеристика у групе/објекте. Поступак сегментације се састоји од рачунања вектора одлика за сваки пиксел, проналажења карактеристичних вектора пиксела (центара кластера који свакако припадају различитим објектима) и придрживање сваког пиксела једном од кластера. Ова метода се примењује са великим успехом у обради сателитских и авионских снимака, а највећа мана је велика рачунарска сложеност.
- **Детекција ивица** (*Edge detection*) - подразумева проналажење тачака слике око којих се значајно мења интензитет боје. На овај начин добијају се линије које представљају ивице објеката. Пошто добијене линије нису увек спојене, додатно се користе алгоритми за реконструкцију недостајућих сегмената. Детекција ивица је моћнији, али и спорији метод сегментације.
- **Слив** (*Watershed*) - свака слика се може посматрати као топографска површина, где низак интензитет боје означава брда и врхове, а висок интензитет означава долине. Идеја алгоритма је да се свака изолована долина (локални минимум интензитета боје) пуни различито обложеном водом. Са порастом нивоа, вода из различитих долина почиње да се спаја. Да би се избегло спајање, постављају се баријере на месту спајања. Ниво воде наставља да расте све док сви врхови не буду под водом. Постављене баријере представљају резултат сегментације.

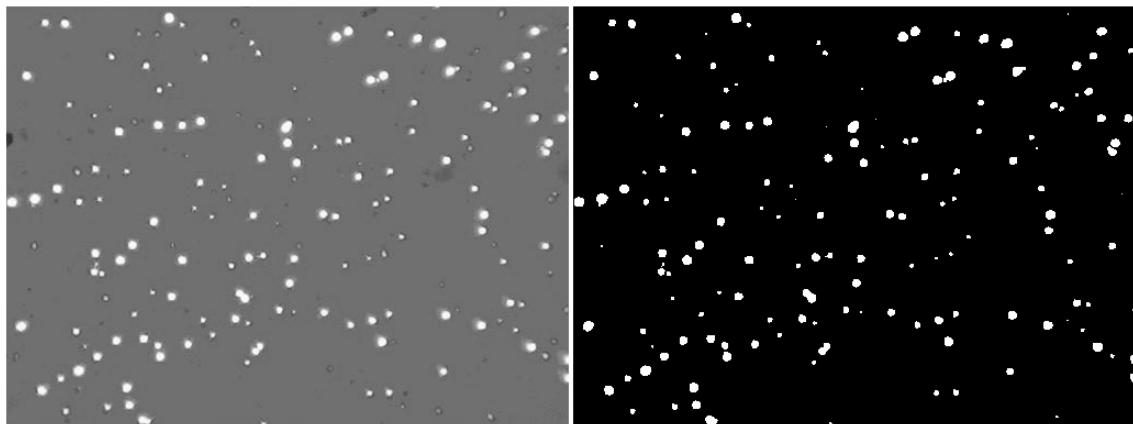
У сличају детекције трагова неутронске дозиметрије, трагови су јасно представљени и различитог су интензитета у односу на позадину. Могу бити представљени светлом бојом на тамној позадини или тамном бојом на светлој позадини. У сваком случају, издавање трагова од позадине је релативно једноставан задатак. Компликованији случај је раздвајање трагова који се преклапају без јасне границе између преклопљених трагова. За решавање овог проблема користи се модификована верзија *watershed* алгоритма. Основни кораци детекције трагова су:

1. раздвајање трагова од позадине,
2. проналажење трагова и
3. сегментација трагова.

У случају да су трагови превише преклопљени, сегметација трагова не може да их детектује као засебне трагове. Из тог разлога, оставља се могућност корекције трагова у каснијој процедуре.

### 2.2.1 Раздвајање трагова од позадине

За раздвајање трагова од позадине користи се сегментација са једним прагом, односно бинаризација. Бинаризација подразумева раздвајање светлих и тамних делова слике, односно, у представљеном случају, идентификовање светлих објеката на тамној позадини.



Слика 2.3: Сиво скалирана (лево) и бинарна/црно-бела (десно) слика

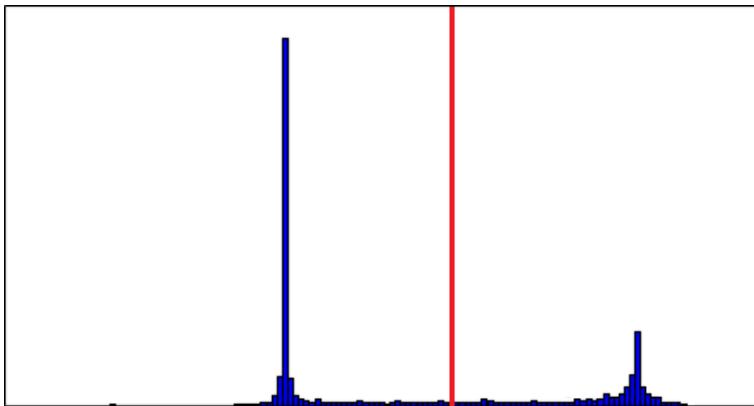
Улазни параметар за бинаризацију је сиво скалирана слика. Резултат бинаризације је црно-бела слика истих димензија као и сиво скалирана слика. Црна боја представља позадину, а белом бојом се означавају пиксели који припадају објектма. Вредност сваког пиксела резултујуће слике рачуна се као:

$$binary(x, y) = \begin{cases} white, & grayscale(x, y) > thresh, \\ black, & \text{иначе.} \end{cases} \quad (2.2)$$

где је *binary* - бинарна слика, *grayscale* - сиво скалирана слика, *thresh* - праг, *white* - вредност белог пиксела, *black* - вредност црног пиксела, а *x* и *y* одговарајући индекси пиксела слика.

**Листинг 2.2: Рачунање бинарне слике са унапред задатим прагом**

```
1 // Calculate binary image.
2 threshold(
```



Слика 2.4: Хистограм интензитета боје на сиво скалираној слици и *Otsu* prag

```

3     grayscale, // Grayscale image.
4     binary,    // Binary image.
5     thresh,    // Threshold value.
6     255,       // Max pixel value, i.e. white pixel value.
7     CV_THRESH_BINARY);

```

Праг према коме се врши класификација може бити и аутоматски одређена вредност помоћу *Otsu* метода. Употребом овог метода креира се хистограм вредности пиксела. Према очекивању, постоје две класе вредности хистограма, и то једна за тамне пикселе и једна за светле пикселе. Као оптимална граница узима се средња вредност између те две класе.

#### Листинг 2.3: Рачунање бинарне слике са *Otsu* pragом

```

1 // Calculate binary image.
2 threshold(
3     grayscale, // Grayscale image.
4     binary,    // Binary image.
5     0,         // Threshold, ignored value.
6     255,       // Max pixel value, i.e. white pixel value.
7     CV_THRESH_BINARY | CV_THRESH_OTSU);

```

### 2.2.2 Проналажење трагова

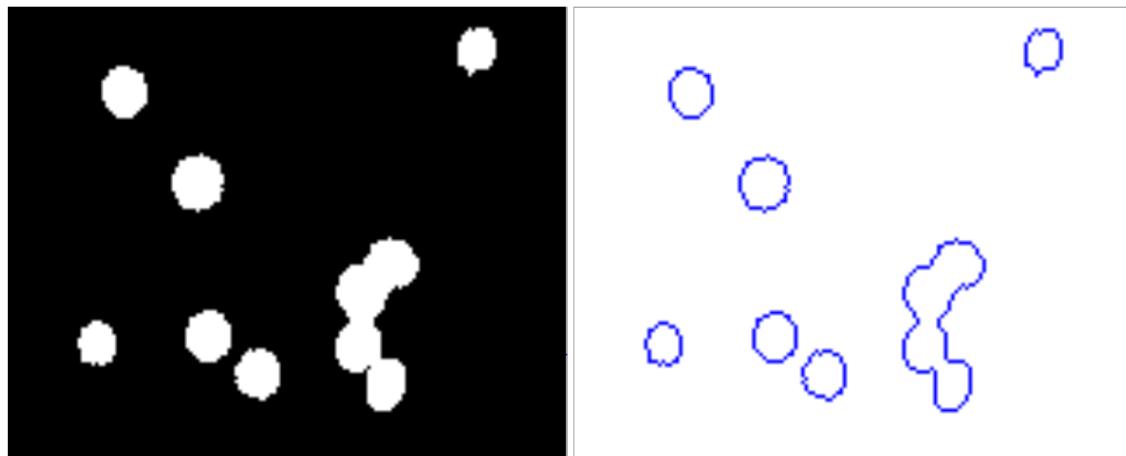
У овом тренутку имамо раздвојене трагове од позадине. На црно белој слици трагови су означени вредношћу 255 (бела боја), а позадина вредношћу 0 (црна боја). Са такве слике проналазимо контуре (контура - скуп тачака који описује објекат) како бисмо могли да испитамо сваку белу површину у циљу сазнавања да ли се ради о појединачном трагу или два или више спојених трагова.

#### Листинг 2.4: Детекција контура трагова

```

1 // Get contours.
2 std::vector<Contour> initialContours;
3 cv::findContours(
4     binary,           // Black-white image
5     initialContours, // Detected contours.
6     CV_RETR_EXTERNAL, // Check only top contours.
7     CV_CHAIN_APPROX_NONE); // Find exact contour points.

```



Слика 2.5: Бинарна/прно-бела слика (лево) и означене контуре са слике (десно)

Како контуре трагова могу бити спојене потребно је да даље процесирајмо такве контуре како бисмо их раздвојили на контуре појединачних трагова. Даље процесирање, односно сегментација трагова, не врши се над свим контурима, већ само над оним чије карактеристике указују на то да би могле бити изграђене од више спојених трагова. Користе се карактеристике попут величине контуре, конвексности контуре, броју одступања од конвексности, највећем одступању од конвексности. Уколико наведене карактеристике контуре нису дале јасну назнаку да се ради о контури једног трага, испитују се тачке које су потенцијални центри спојених трагова.

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0

→

0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0
0	1	2	2	2	2	2	1	0	
0	1	2	3	3	3	2	1	0	
0	1	2	3	3	3	2	1	0	
0	1	2	2	2	2	2	1	0	
0	1	1	1	1	1	1	1	0	
0	0	0	0	0	0	0	0	0	

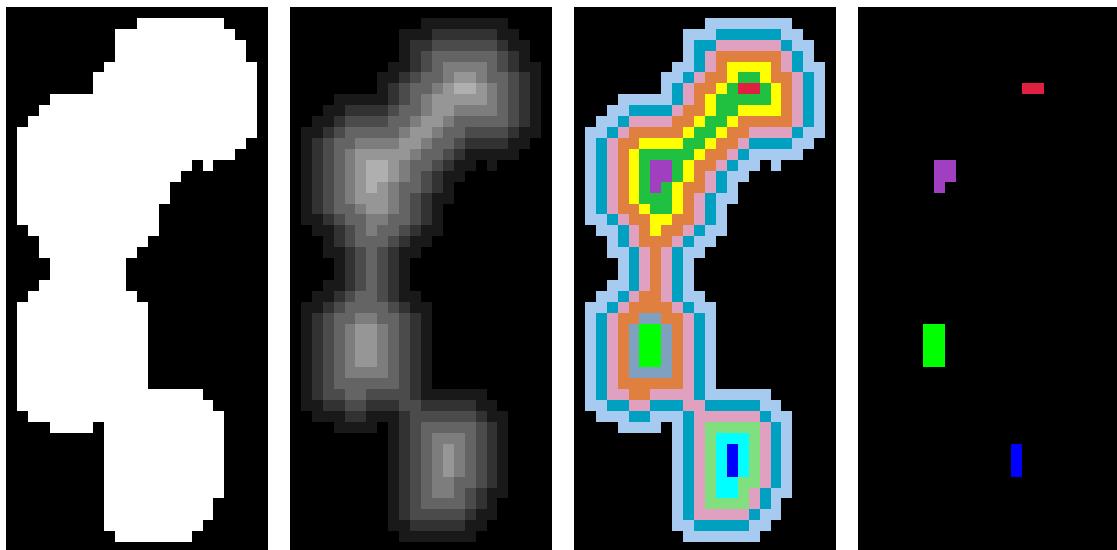
Слика 2.6: Матрица удаљености од позадине

Потенцијалне центре спојених трагова тражимо на основу матрице удаљености од позадине. Матрица удаљености од позадине се обично рачуна само за бинарне слике, тако да најпре извлачимо маску са бинарне слике трага који испитујемо, осноносно најмањи квадрат са слике који садржи траг, а чије су ивице паралелне са ивицама слике. Над маском трага рачунамо матрицу удаљености (*distance transform*). Резултат је сиво скалирана слика која личи на улазну слику, с тим да интензитети боја тачака унутар објеката показују удаљеност до најближе границе са позадином.

Када израчунамо матрицу удаљености, користимо *connectedComponents* функцију *OpenCV*-ја да означимо засебне компоненте. Једна компонента представља све пикселе који се додирују, а притом имају исту вредност. Компоненте које представљају локалне максимуме користе се као потенцијални центри трагова.

#### Листинг 2.5: Поналажење централних региона трагова

```
1 // Calculate distance matrix.
```



Слика 2.7: (i) Бинарна маска трага; (ii) Матрица удаљености; (iii) Означене компоненте са матрице удаљености; (iv) Компоненте са матрице удаљености које представљају локалне максимуме вредности на матрици удаљености;

```

2 cv::Mat dist;
3 cv::distanceTransform(framedMask, dist, CV_DIST_L2, 5);
4
5 // Converts distance to integers.
6 cv::Mat dist_8u;
7 dist.convertTo(dist_8u, CV_8U);
8
9 // Mark local maximums and use it as hint for traces.
10 number0fMarkers = MarkLocalMaximums(dist_8u, markers);

```

Уколико је детектован само један локални максимум, траг се ипак сматра засебним. У случају да је детектовано више локалних трагова, ради се сегментација, при чему се локални максимуми користе као полазни региони за *watershed* алгоритам.

### 2.2.3 Сегментација трагова

*OpenCV* садржи модификовану верзију *watershed* алгоритма. Имплементација не проналази аутоматски регионе из којих креће да означава сличну околину, већ те регионе тражи као улазни параметар. Ту се користе локални максимуми одређени на основу матрице удаљености. Добијене компоненте представљају детектоване трагове.

#### Листинг 2.6: Сегментација трагова

```

1 // Markers are previously obtained local maximas.
2 // Mark background.
3 for (int r = 0; r < mask.rows; r++)
4 {
5     for (int c = 0; c < mask.cols; c++)
6     {
7         int index = (int) mask.at<char>(r, c);
8         if (index == 0)
9         {
10             markers.at<int>(r, c) = 255;
11         }
12     }

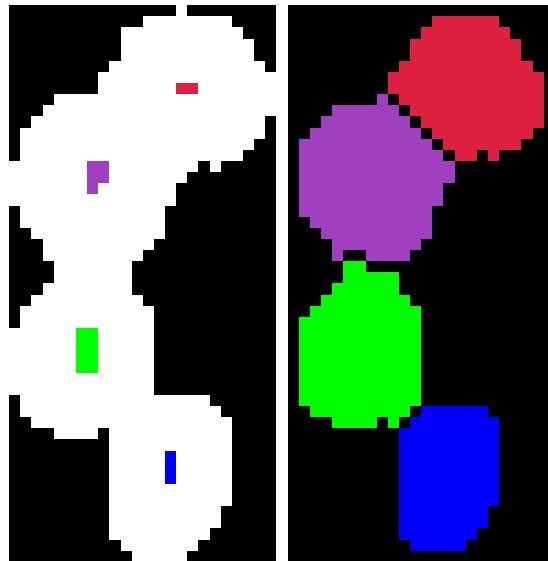
```

```

13 }
14
15 // Run watershed alghoritm to segment connected traces.
16 Mat result;
17 cvtColor(mask, result, CV_GRAY2BGR);
18 cv::watershed(result, markers);

```

---



Слика 2.8: Полазна и крајња матрица за *watershed* алгоритам

## 2.3 Исправљање ротације слике у односу на претходну слику из секвенце

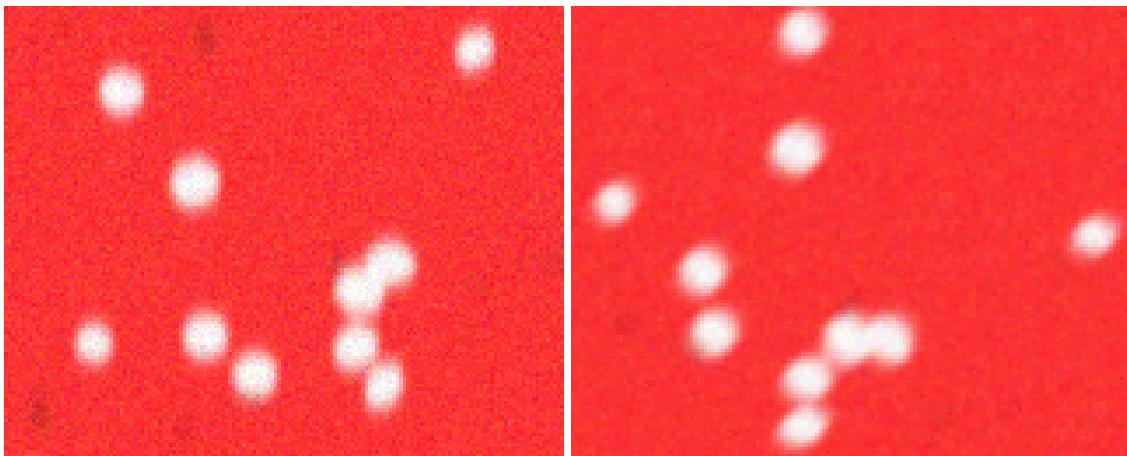
При прикупљању секвенце слика у различитим временским тренуцима дешава се да материјал детектора није усликан под истим углом. У том случају, позиција истих трагова на различитим сликама није иста и није могуће пратити раст трагова. Зато приступамо отклаљању ротације (*deskew*) слике у односу на предходну слику из секвенце.

Пошто је могуће да се трагови не шире у истом правцу и на исти начин, може да се деси и да се контуре трагова значајно разликују. Проналажење сличности између две сличне слике је део многих апликација рачунарског вида: калибрација камере, 3Д реконструкција, препознавање објекта и тако даље. Ови алгоритми се углавном сastoје од три главна корака:

1. детекција интересних тачака,
2. рачунање описног вектора у интересним тачкама и
3. поређење описних вектора и одређивање трасформације.

### 2.3.1 Детекција интересних тачака

Интересне тачке су препознатљиве тачке објекта које се могу пронаћи при различитим погледима на објекат. Описивањем ових тачака на такав начин да су независне од величине



Слика 2.9: Претходна слика из секвенце (лево) и нова слика (десно)

слике, ротације и осветљења, постиже се циљ уочавања сличности између различитих слика. Поређењем описних вектора може се донети закључак о трансформацији слике. Ове тачке се бирају на карактеристичним местима, као што су оштре ивице, мрље, Т-спојеви. У посматраном случају, алгоритам се ослања на чињеницу да се исти узорак слика у различитим времеским тренуцима. Односно, да ће на сликама бити присутни исти трагови и да ће однос између суседних трагова остати исти. Због тога се за интересне тачке бирају центри детектованих трагова.

### 2.3.2 Рачунање описног вектора

Описни вектор представља вектор који описује околину интересне тачке. Овај вектор мора бити карактеристичан, отпоран на шум и дефекте који настају при сликању. Како се алгоритам ослања на претпоставку да у новој слици из секвенце нема превише нових трагова и новонасталог шума, описни вектор за сваку тачку се гради од сортираних удаљености центра дате тачке од центара пет најближих трагова.

### 2.3.3 Поређење описних вектора и одређивање трансформације

На крају се врши упоређивање описних вектора, односно рачуна се еуклидско растојање између описног вектора интересне тачке са описним векторима свих интересних тачака друге слике. Када се добије списак парова са најмањим растојањем, бира се најмање три, а највише десет тачака на основу којих се тражи матрица хомографије. Хомографија је пројекцијска трансформација којом се описују пресликовања тачке из једне равни у другу:

$$x' = Hx, \quad (2.3)$$

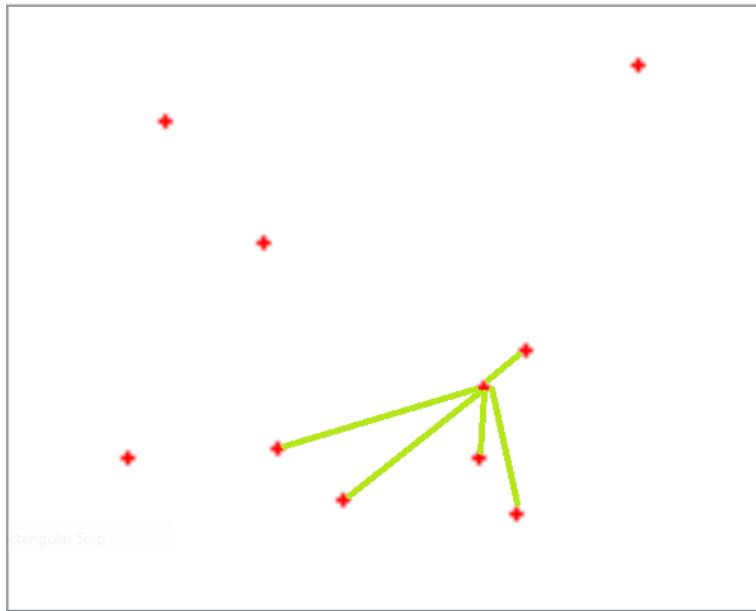
где су  $x'$  и  $x$  тачке у различitim равнима, а  $H$  матрица трансформације која описује транслацију, ротацију, скалирање и перспективу. У датом случају, очекује се афина хомографија, односно да нема перспективе.

Листинг 2.7: Поређење описних вектора и одређивање ротације нове слике

```

1 Mat Document::CalcTransform(Document* prev, Document* curr)
2 {
3     // Get traces for previous and current image.
4     vector<Trace>& prevTraces = prev->GetTraces();

```

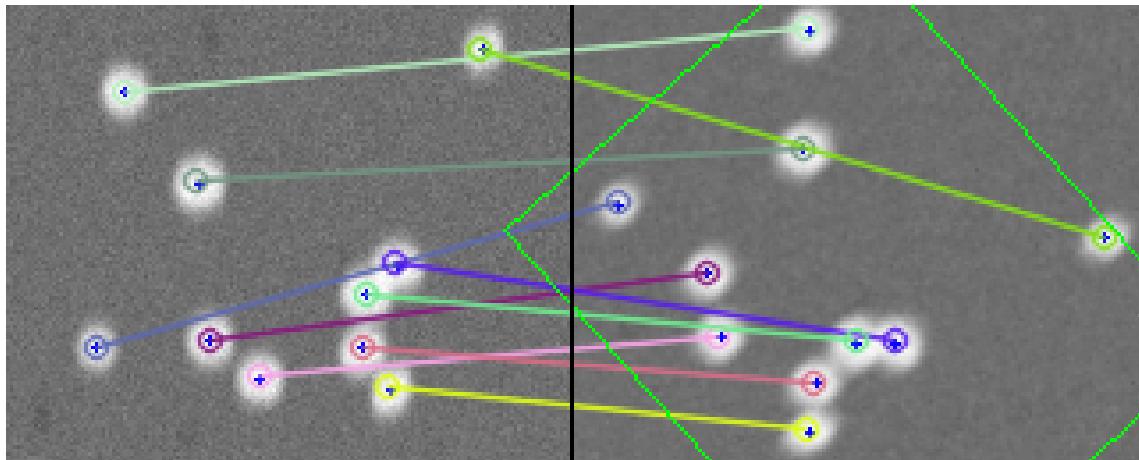


Слика 2.10: Вредности описаног вектора једног трага

```

5     vector<Trace>& currTraces = curr->GetTraces();
6
7 // Get distance from 5 closest traces for both, previous and current
8 // image.
9 Mat prevN5 = GetDist(prevTraces);
10 Mat currN5 = GetDist(currTraces);
11
12 // Match desctiption vectors.
13 BFMatcher matcher;
14 std::vector<DMatch> matches;
15 matcher.match( prevN5, currN5, matches );
16
17 // Pack matches in vector and sort them by distance,
18 std::vector<DMatch*> refMatches = SortMatches(matches);
19
20 // Get matching points witht least distance.
21 std::vector<Point2f> pPrev;
22 std::vector<Point2f> pCurr;
23 for(size_t i = 0; i < matches.size() && i < 10; i++)
24 {
25     DMatch* m = refMatches[i];
26
27     pPrev.push_back(getTraceCenter(prevTraces[m->queryIdx]));
28     pCurr.push_back(getTraceCenter(currTraces[m->trainIdx]));
29 }
30
31 Mat transform;
32 // Minimum 3 points to detect transformation.
33 if (pCurr.size() >= 3 && pPrev.size() >= 3)
34 {
35     Mat temp = findHomography(pCurr, pPrev, CV_RANSAC);
36
37     // Check is it affine homography.
38     // h_{31}=h_{32}=0, \; h_{33}=1.
39     double h31 = temp.at<double>(2,0);
40     double h32 = temp.at<double>(2,1);
41     double h33 = temp.at<double>(2,2);

```



Слика 2.11: Препознате тачке са најмањим еуклидским растојањем описних вектора и детектована трансформација

```

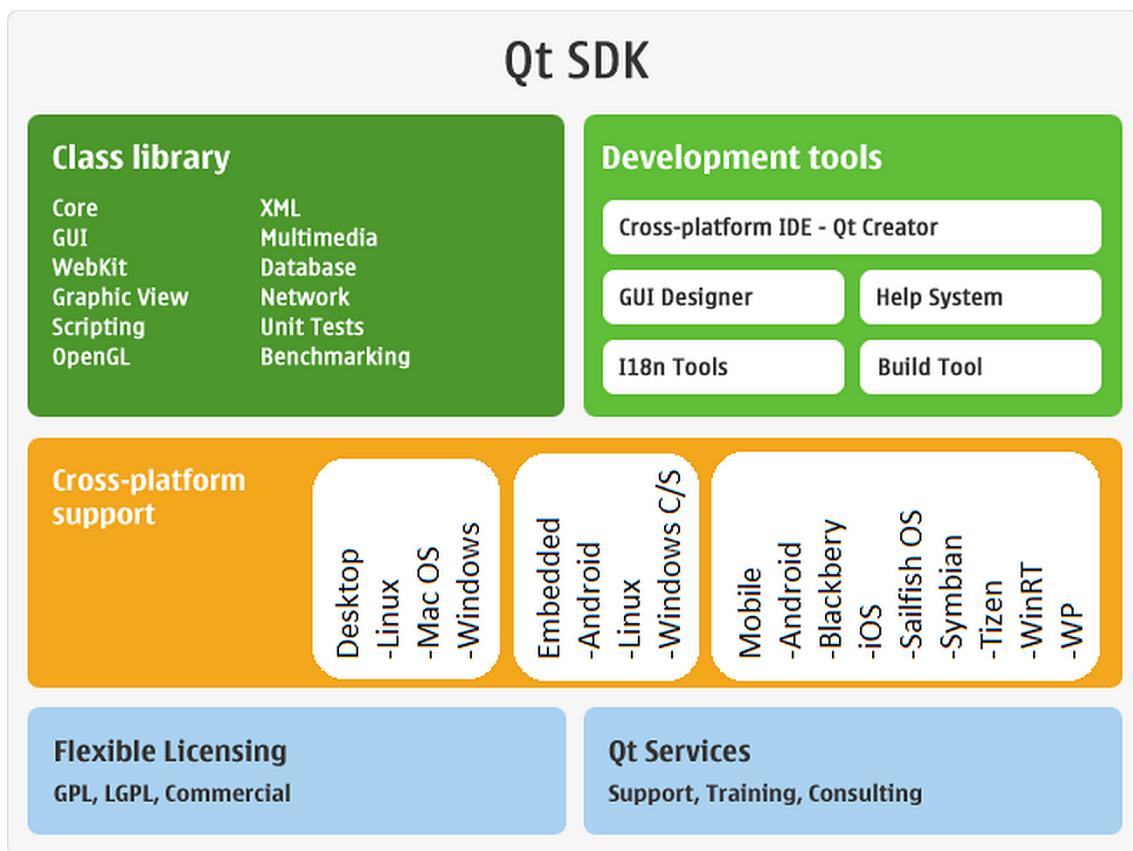
41     if (abs(h31) < cAlmostZero && abs(h32) < cAlmostZero && abs(h33 - 1)
42         < cAlmostZero)
43     {
44         transform = temp;
45     }
46
47     return transform;
48 }
```

Дати алгоритам функционише готово без грешке ако се примени на вештачки генеријаним примерима. Међутим, у случају да су трагови превише густи, односно детекција не разреши спојене трагове, или у случају да на слици има веома мало трагова, може се очекивати да трансформација не буде успешно детектована. Алгоритам се може побољшати увођењем нових вредности у описни вектор интересних тачака, на пример: однос дужег и краћег дијаметра, углови који се формирају са најближим тачкама, а чије је теме у посматраној тачки итд.

## Глава 3

# Развој корисничког интерфејса употребом *Qt* оквира

*Qt* [6] је оквир за развој корисничког интерфејса. Омогућује развој апликација и за десктоп рачунаре и за мобилне телефоне, без потребе да се поново пише изворни код. *Qt* је надскуп стандардног *C++* програмског језика, тако да програмери могу да користе или *Qt* или стандардне *C++* библиотеке, а могу да се користе и у комбинацији.



Слика 3.1: *Qt SDK (Software Development Kit)*

*Qt* већ користе стотине хиљада програмера. Значајани делови *Linux KDE* графичког окружења, *Google Earth* и *Skype* програма су написани коришћењем овог оквира. Обухвата богат скуп графичких контрола *Widgets* које пружају стандардне *GUI* функционалности као што су менији, контекстни мени, *drag and drop*, и слично. Уводи иновативну и безбеднију

алтернативу за комуникацију између објеката названу *Signals and Slots*, али подржава и конвенционални модел за реаговање на клик миша, притисак дугмета, и друге улазне догађаје.

Од самог почетка *Qt* је био доступан за слободну комерцијалну употребу. То је омогућило развој апликација без икаквих ограничења лиценцирања. Постојало је више различитих лиценци. У почетку изворни код је био под лиценцом *FreeQT* која је дозвољавала преузимање и употребу кода, али је објављивање модификација било забрањено. Тренутно, код је доступан под *GNU Lesser General Public 2.1 (LGPL)*, која га чини доступним за употребу у пројектима и отвореног и затвоеног кода.

## 3.1 Развојни алати

*Qt* нуди широк скуп алата за развој софтвера:

- **Qt Assistant** је алат који графички презентује *HTML* документацију пројекта. Садржи примере кода и пуно корисних информација о томе како се користите класе оквира. Такође, пружа могућност једноставне претраге документације.
- **Qt Designer** је алат за пројектовање и изградњу корисничког интерфејса. Кориснички интерфејс се развија на *what you see what you get (WYSWYG)* начин. Могуће је тестирање понашања интерфејса при различитим резолуцијама, што олакшава развој на различитим платформама. Визуелне контроле креирање са овим дизајнером интегришу се са програмским кодом употребом модела сигнала и слотова што олакшава раздавање функционалности од корисничког интерфејса. Све карактеристике визуелних контрола се могу мењати и динамички у коду. Информације о корисничком интерфејсу се чувају у фајловима са екстензијом *.ui*. То су *XML* фајлови на основу којих се аутоматски генерише *C++* код који се даље компајлира.
- **Qt Linguist** је алат који омућава локализацију апликације.
- **Qt Creator** је развојно окружење (*IDE - Integrated Development Environment*) које интегрише претходно наведене алате. Такође, пружа алате за извршавање задатака током целог циклуса развоја апликације. Аутоматизује и убрзава задатке као што су креирање пројекта, додавање нових фајлова, нуди семантичко означавање, проверу синтаксе кода, комплетирање наредбе, рефакторисање, као и многе друге корисне функције.

## 3.2 Модули

*Qt* фрејмворк је подељен у неколико засебних модула. Да би се користиле функционалности одређеног модула, потребно је додати одговарајућу променљиву у пројектни фајл. На пример, за употребу класа за рад са *XML* фајловима потребно је додати:

Листинг 3.1: Укључивање модула за рад са *XML* фајловима

```
1 QT += xml;
```

Коришћени модули за приказ корисничког интерфејса у софтверу за детекцију трагова неутронске дозиметрије обухватају:

- **QtCore** садржи основне класе које користи свака *Qt* апликација, као и други модули. Овај модул омогућава рад са објектима (**QObject**), стринговима (**QString**), фајловима (**QFile**), фолдерима ( **QDir**), локализацијом (**QLocale**) и тако даље. Овај модул је подразумеван, тако да га није потребно експлицитно укључивати.

- **QtGui** модул садржи компоненте графичког корисничког интерфејса (*GUI*). Садржи специјализоване контроле **Widgets** (**QCalendarWidget**, **QDockWidget**), контејнере (**QGroupBox**, **QStackedWidget**) и дијалоге (**QFileDialog**, **QPrintDialog**).
- **QtXml** садржи класе за рад са **Xml** фајловима. Омогућава читање, писање и форматирање **Xml** структура.

### 3.3 Сигнали и слотови

Сигнали и слотови (*Signals and Slots*) су основни механизам свих *Qt* програма. Омогућавају комуникацију између објеката који не знају ништа једни о другима. Типичан пример је када желимо да промена у једној од контрола утиче на промене у другој контроли. Само наслеђене класе из *QObject* могу применити механизам сигнала и слотова. *Widget* контроле имају пуно предефинисаних сигнала и слотова, али је могуће додефинисати нове по потреби.

Идеја је да један објекат може да пошаље сигнал са информацијама насталој промени без знања да ли ће ту информацију неко прихватити. Други објекти се могу повезати на ту врсту сигнала, и при емитовању сигнала од стране првог објекта могу извести одређену акцију. Могуће је повезати само сигнале и слотове са истим параметрима. Слотови се користе за пријем сигнала. У суштини, то су обичне *C++* функције, тако да се могу извршити у било које време.

Листинг 3.2: Дефиниција и емитовање сигнала

```

1 class Sender : public QObject
2 {
3     Q_OBJECT
4
5 public:
6     void sendSignal()
7     {
8         // Emit signal.
9         emit newMessage("Message text!");
10    }
11
12 signals:
13     void newMessage(QString message);
14 };

```

Листинг 3.3: Дефиниција слота

```

1 class Receiver : public QObject
2 {
3     Q_OBJECT
4
5 public slots:
6     receiveMessage(QString message)
7     {
8         // Print received message to debug console.
9         qDebug() << message;
10    }
11 };

```

Сигнали и слотови се могу повезати на следећи начин:

Листинг 3.4: Повезивање сигнала и слотова

```
1 QObject::connect(
```

```
2     sender,
3     SIGNAL(newMessage),
4     receiver,
5     SLOT(receiveMessage));
```

---

На један сигнал могуће је прикачiti више слотова. Чекање на сигнал се може и укинути:

**Листинг 3.5: Укидање везе сигнала и слота**

```
1 QObject::disconnect(
2     sender,
3     SIGNAL(newMessage),
4     receiver,
5     SLOT(receiveMessage));
```

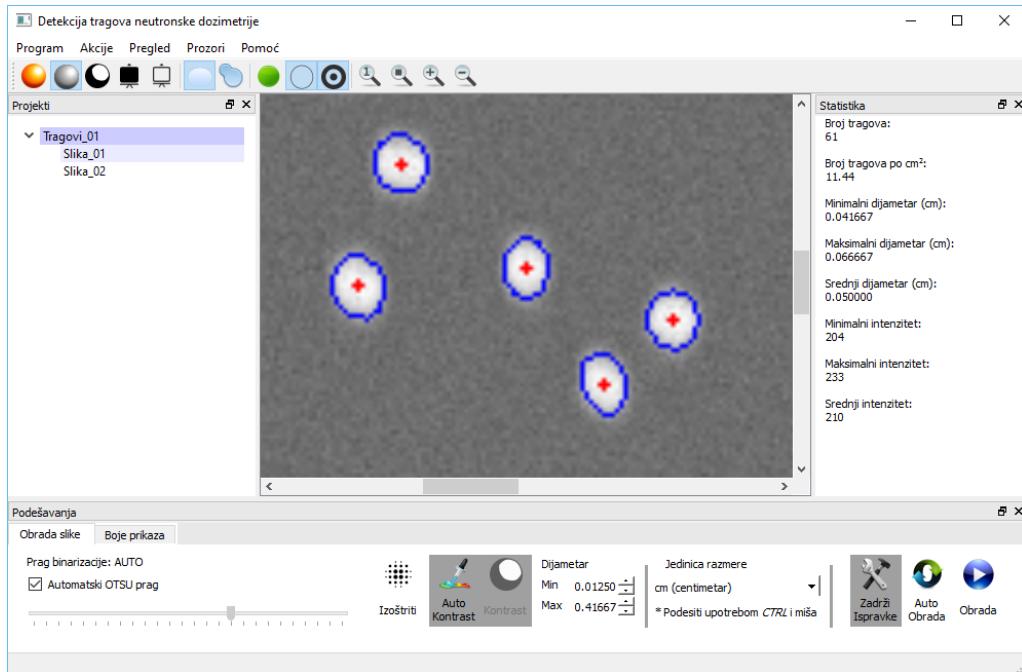
---

## Глава 4

# Софтвер за детекцију трагова

Софтвер за детекцију трагова неутронске дозиметрије омогућава обраду слика неутронске дозиметрије добијених употребом детектора *CR-39*. Резултат употребе софтвера је добијање информација о броју трагова неутрона, њиховој позицији и величини. Омогућено је подешавање параметара обраде слика, подешавање приказа, као и чување резултата обраде. Софтвер је намењен истраживачима са Института за физику Природно-математичког факултета Универзитета у Крагујевцу, који се баве мерењем дозе апсорбованог неутронског зрачења.

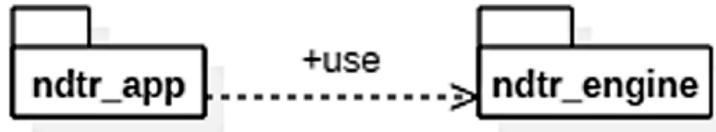
Обрада слике је веома сложена и захтевна по питању перформанси. Услед спредности у брзини у односу на неке друге платформе, софтвер је комплетно развијен у програмском језику *C++*. Поред стандардних библиотека *C++* језика, коришћена је *OpenCV* библиотека за обраду слика, као и *QT* платформа за развој корисничког интерфејса. Као такав, софтвер је портабилан, а тестиран је на *Ubuntu 14.04 LTS* и *Windows 7/8.1/10* оперативним системима.



Слика 4.1: Софтвер за детекцију трагова неутронске дозиметрије

## 4.1 Структура пројекта

Изворни код софтвера налази се на адреси <https://github.com/nik0la31/Application-of-pattern-recognition-algorithms-in-neutron-dosimetry>. Код је подељен у две целине, односно пројекта: `ndtr_engine` и `ndtr_app`, које су задужене за детекцију трагова неутронске дозиметрије обрадом слике и графички кориснички интерфејс, респективно.



Слика 4.2: Структура пројекта

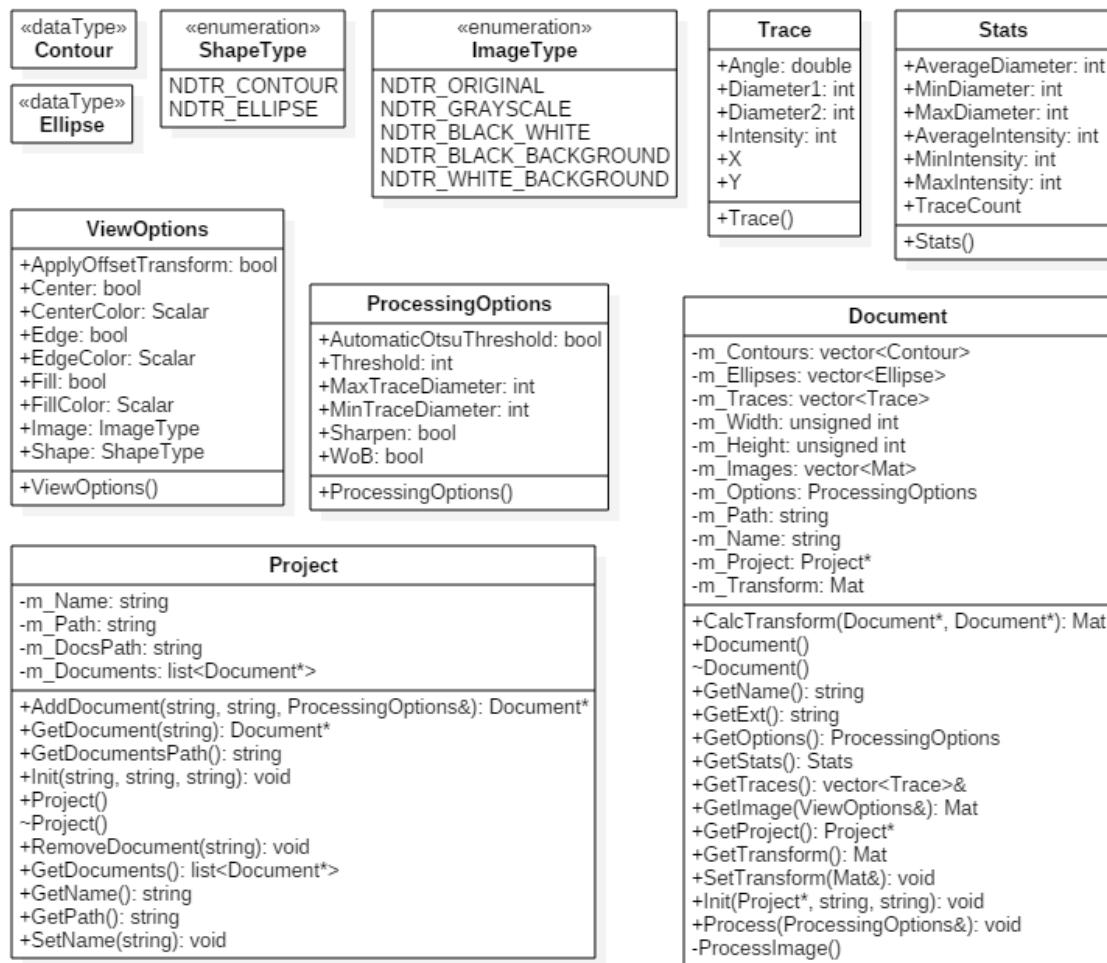
Пројекат `ndtr_app` садржи свега четири класе:

- `MainWindow` класа садржи дизајн корисничког интерфејса и управља интеракцијом са корисником;
- `Workspace` класа представља модел у *Model/View* архитектури. Чува све релевантне податке о учитаним пројектима и сликама, тренутном приказу, одабраним подешавањима и слично;
- `ProjectItem` је изведен из *QT* класе `QStandardItem` и дефинише специфично понашање чворова пројекта и докумената (слика) у хијерархијском приказу;
- `ProjectParser` је класа задужена за чување и учитавање пројекта.

Пројекат `ndtr_engine` представља срж рада и садржи следеће типове:

- `Contour` описује контуру трага, односно садржи вектор тачака који описује траг.
- `Ellipse` описује најмању елипсу која се може описати око детектованог трага. Дефинисана је центром, два дијаметра, и углом.
- `ShapeType` енумерација дефинише који се облици трага могу приказати (контура, елипса).
- `ImageType` енумерација дефинише који се типови слике могу приказати (изворна, сиво скалирана, црно бела, са црном позадином, са белом позадином).
- `ViewOptions` структура се користи да се задају опције приказа да би се генерисала одговарајућа слика.
- `ProcessingOptions` структура служи за задавање подешавања обраде слике. Могућа подешавања су: употреба аутоматског *Otsu* прага бинаризације, задавање конкретног прага бинаризације, изоштравање слике, дефинисање односа контраста трагова у односу на позадину, односно бели трагови на тамној позадини или супротно (*WoB - white on black*).
- `Trace` структура описује детектовани траг. Траг је дефинисан са два дијаметра, углом,  $x$  и  $y$  координатама центра трага, просечним интензитетом боје.
- `Stats` структура садржи статистичке податке који се рачунају за сваку слику: Најмањи, највећи и просечни дијаметар трага, најмањи, највећи и просечни интензитет боје трага.

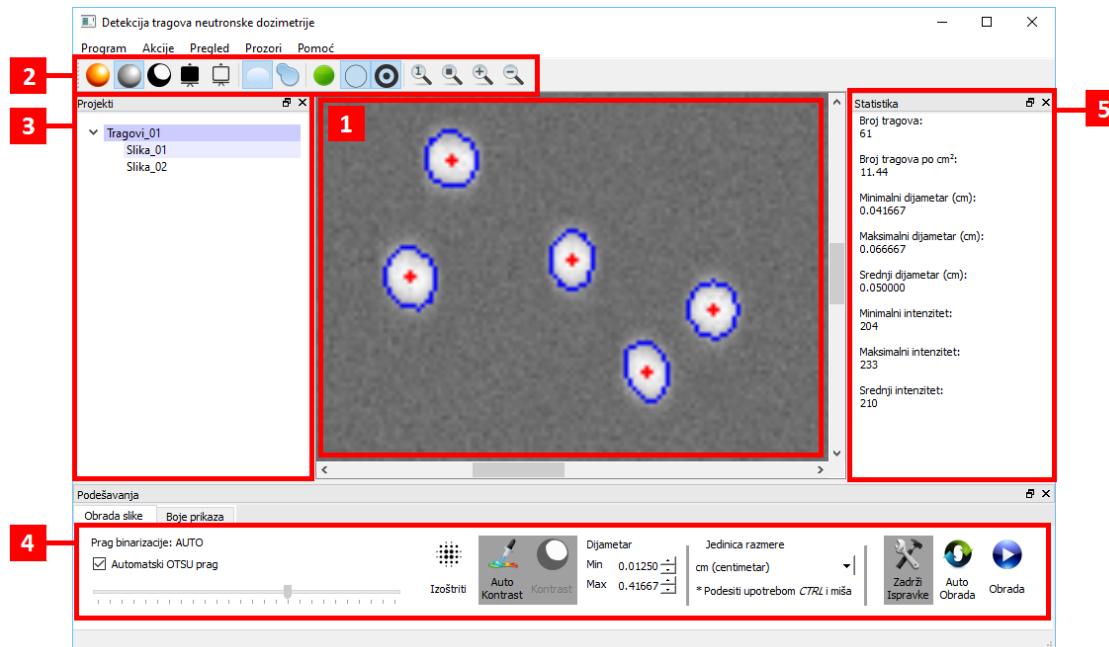
- **Document** класа садржи методе за детекцију трагова. Такође, садржи методу за детекцију трансформације, односно угла између две слике у секвенци. Ослања се на *OpenCV* библиотеку.
- **Project** класа садржи документе, основно слике са процесуираним подацима. Води рачуна о додавању, брисању и редоследу докумената у пројекту.

Слика 4.3: Дијаграм класа *ndtr\_engine* пројекта

## 4.2 Начин употребе

Кориснички интерфејс је једноставан за коришћење и прегледан, тако да је корисницима омогућено брзо прилагођавање на рад овим алатом. На слици 4.4 је приказан кориснички интерфејс програма. Подељен је пет главних целина:

1. Приказ и визуелизација детектованих трагова.
2. Опције приkaza детектованих трагова.
3. Управљање пројектима и сликама.
4. Опције обраде слике.
5. Приказ података добијених обрадом слике.



Слика 4.4: Основне компоненте софтвера за детекцију трагова неутронске дозиметрије

#### 4.2.1 Управљање пројектима

Контрола за управљање пројектима омогућава креирање, отварање, затварање и чување пројеката. Пројекат служи као контејнер који групише слике неутронске дозиметрије у јединствену целину. Такође, кроз ову контролу врши се и додавање и брисање слика неутронске дозиметрије из пројекта. Структура пројекта на диску је следећа:

```
NazivProjekta.ndtr
<NazivProjekta>
    NazivSlike.ext
    ...

```

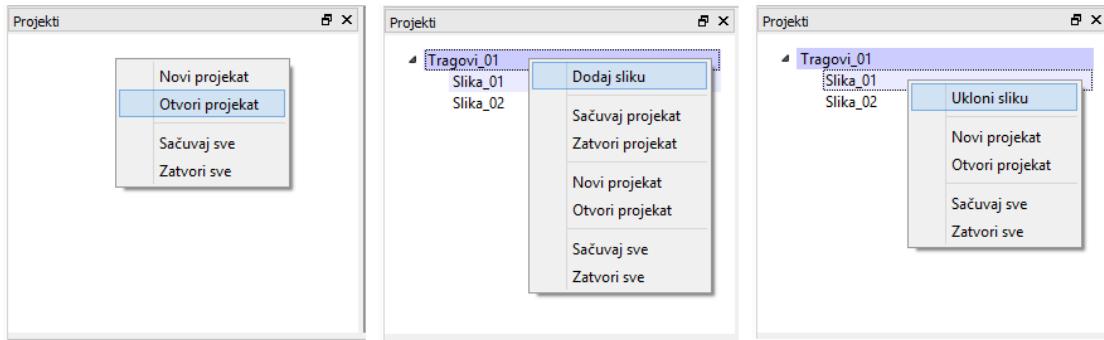
У директоријуму који носи назив пројекта налазе се све слике се које користе у пројекту, а истоимена датотека са екstenзијом *.ndtr* садржи и дефиницију пројекта.

Листинг 4.1: Пример дефиниције пројекта

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ndtr>
3     <documents>
4         <document name="Slika_01" ext=".png">
5             <options AutomaticOtsuThreshold="1" OtsuThreshold="160" Sharpen="0" WoB="1" MinTraceDiameter="5" MaxTraceDiameter="40"/>
6         </document>
7         <document name="Slika_02" ext=".png">
8             <options AutomaticOtsuThreshold="1" OtsuThreshold="161" Sharpen="0" WoB="1" MinTraceDiameter="5" MaxTraceDiameter="40"/>
9         </document>
10    </documents>
11 </ndtr>

```



Слика 4.5: Контрола за управљање пројектима и опције менија

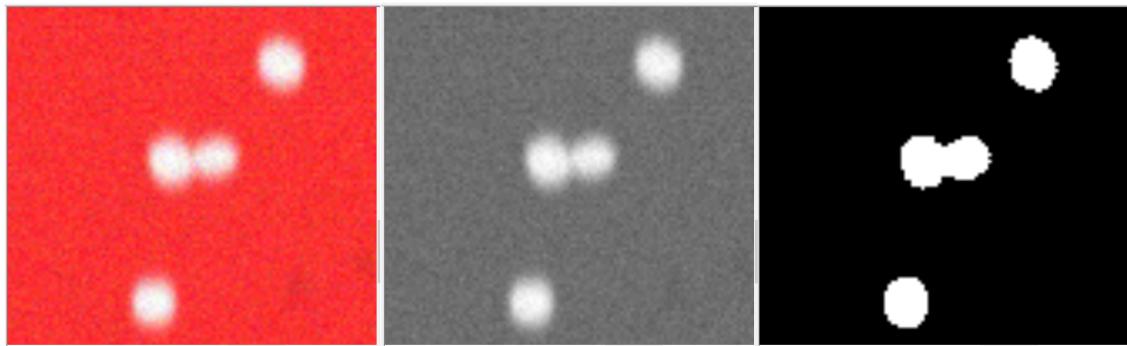
#### 4.2.2 Приказ и опције приказа детектованих трагова

У централном делу програма налази се приказ слике неутронске дозиметрије и детектованих трагова. Изнад се налазе опције приказа којима се управља шта се заправо приказује и шта је означено на приказу.



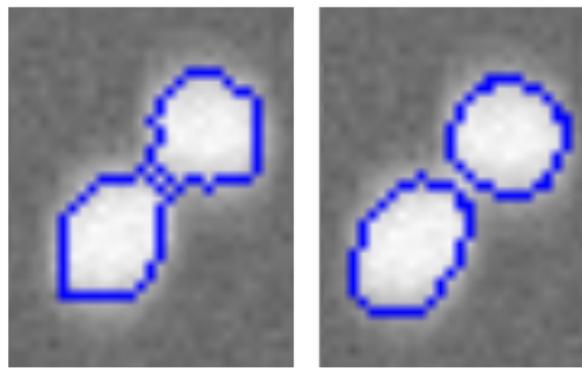
Слика 4.6: Опције приказа

Први сет опција дефинише тип слике која се приказује. Могуће је одабрати изворну слику, сиво скалирану слику, црно-белу слику, слику са црном позадином и слику са белом позадином. Приказ различитих типова слика може бити користан да упути на могућу корекцију параметара за обраду слика.



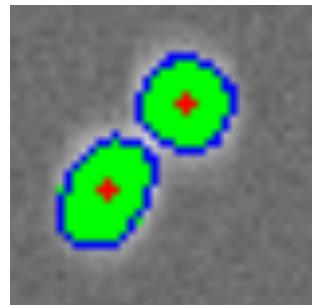
Слика 4.7: Опције приказа: изворна, сиво скалирана и црно-бела слика

Други сет опција дефинише у ком облику треба приказати детектовани траг. Могуће опције су контура или елипса.

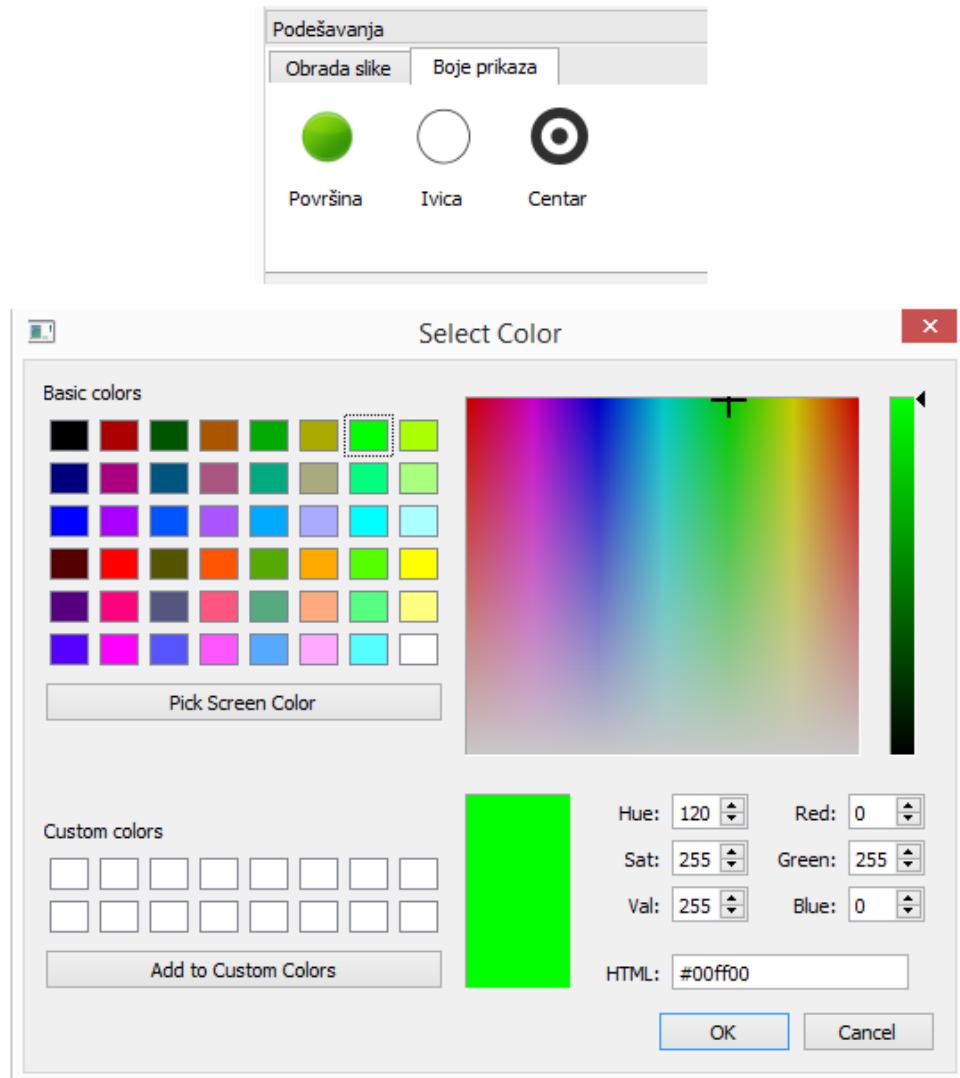


Слика 4.8: Опције приказа: контура трага или елипса која описује траг

Наредни сет опција дефинише како се детектовани траг означава на приказу. Могуће је означити ивицу, целу површину, као и центар трага. У контроли за подешавања, могуће је одредити којим се бојама приказују трагови на слици.



Слика 4.9: Опције приказа: ивица је приказана плавом, површина зеленом, а центар трага црвеном бојом



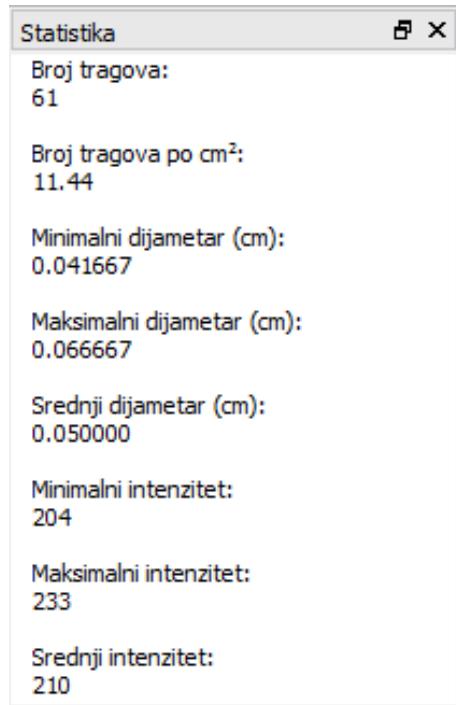
Слика 4.10: Подешавања: одабир боја приказа трагова

Последњи сет опција тиче се величине приказане слике. Слику је могуће увећати, умањити, приказати у извornoј величини или је приказати целу на доступној површини контроле.

Тренутни приказ се може копирати или сачувати одабиром одговарајућих наредби из менија Акције -> Копирај слику, односно Акције -> Сачувај слику.

#### 4.2.3 Приказ и извоз резултата обраде

Контрола за преглед резултата обраде слике пружа основне податке о детектованим траговима. Приказани су подаци попут укупног броја трагова, величине најмањег, највећег и средњег дијаметра трага као и најмањег, највећег и средњег интензитета боје трага.



Слика 4.11: Контрола за приказ резултата обраде слике

Кликом на одређени траг из приkaza, добијају се и основне информације датог трага попут позиције, величине и интензитета боје. Подаци везани за целу слику и за сваки детектовани траг појединачно могу се сачувати у .csv датотеку наредбом Акције -> Извоз из менија програма.

#### Листинг 4.2: Пример сачуваних резултата

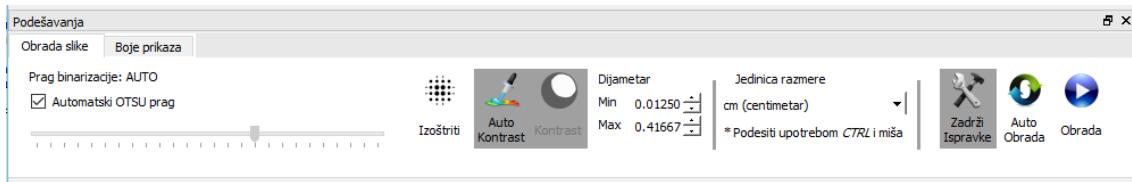
```

1 Broj tragova,
2 61,
3 Statistika ,
4 Dijametar ,
5 Min ,Max ,Avg
6 10 ,16 ,12 ,
7 Intenzitet ,
8 Min ,Max ,Avg
9 204 ,233 ,210 ,
10 Tragovi ,
11 x ,y ,ugao ,dijametar1 ,dijametar2 ,intenzitet ,
12 260 ,463 ,172.956 ,12 ,14 ,211 ,
13 199 ,454 ,128.87 ,12 ,14 ,209 ,
14 . . .

```

#### Подешавање параметара и обраде слике

На посебној контроли налазе се доступна подешавања обраде слике и контроле којима се обрада покреће. Доступна подешавања су: укључивање *Otsu* прага бинаризације, задавање прага бинаризације, изоштравање слике, као и одређивање да ли су трагови светлији или тамнији у односу на позадину (*Контраст* претпоставља детекцију светлих трагова на тамној позадини). Такође, детектоване трагове могуће је филтрирати према величини, односно могуће је задати најмањи и највећи дозвољени дијаметар трага.



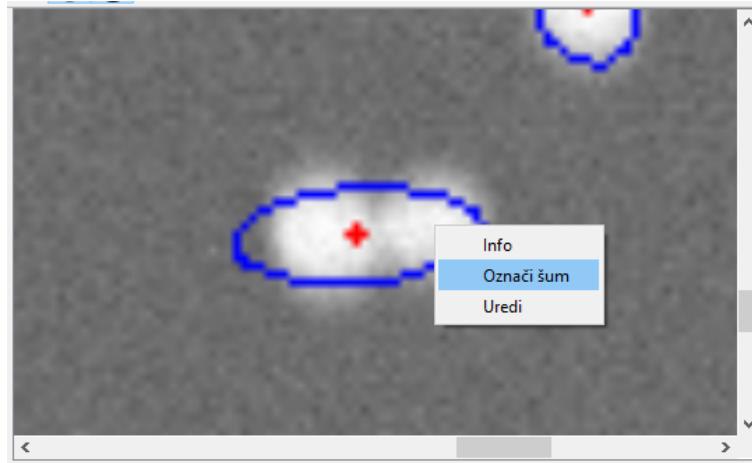
Слика 4.12: Контрола за подешавање и покретање обрада слике

Поред наведених подешавања, на контроли се налазе и команде за покретање обраде, као и опција за аутоматску обраду при промени било ког параметра.

### Корекција детектованих трагова

Како је могуће да програм погрешно класификује траг или шум, односно да не направи добру поделу између трагова, омогућена је ручна исправка.

Уколико је траг погрешно класификован, десним кликом на траг приказује се мени из којег се могу одабрати опције *Означи траг* или *Означи шум*.



Слика 4.13: Корекција класификације траг-шум

Уколико граница између трагова није тачна, опција из менија *Уреди* омогућава мануелно одреживање граница и класификовање трагова.



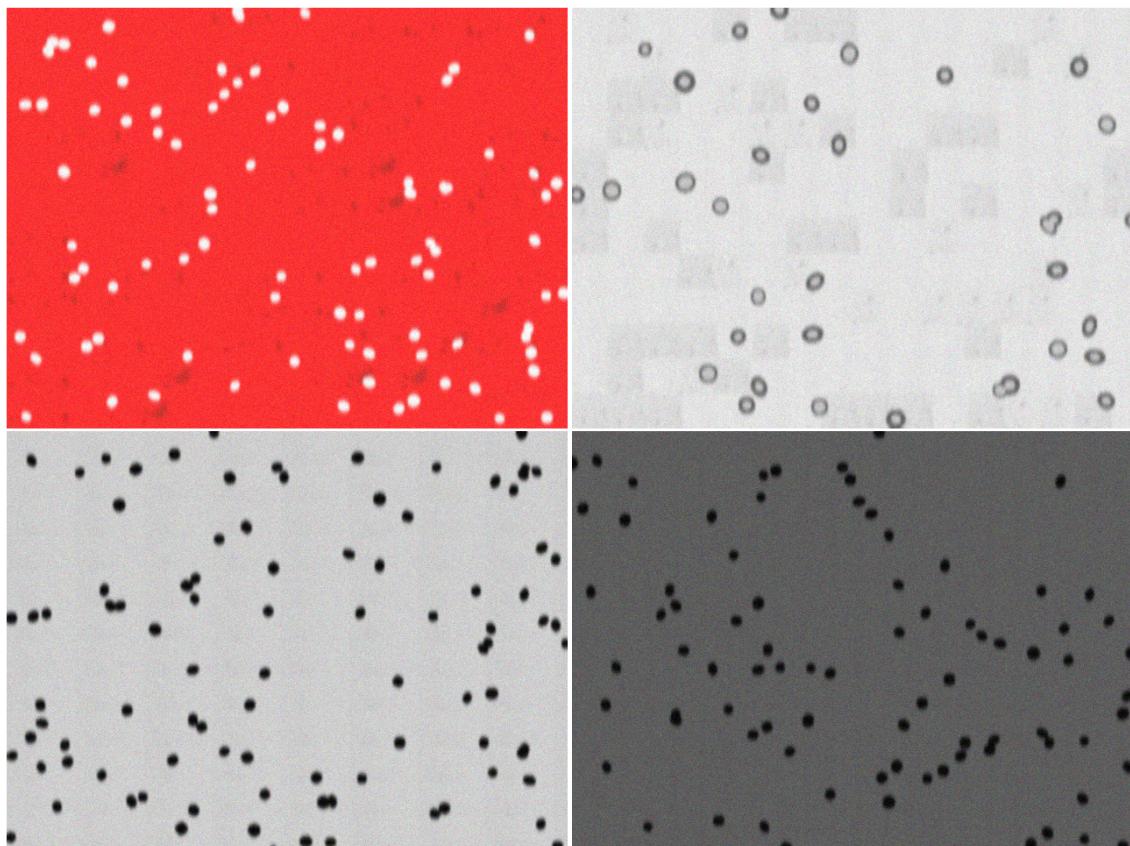
Слика 4.14: Корекција трагова

Десним кликом се обележавају трагови (плава боја). Левим кликом се обележава шум (црвена боја). Није потребно означити цео траг, већ само приближне границе. Означени делови се користе као полаз за претходно описаны *Watershed* алгоритам. У случају погрешне ознаке, означени траг се може и обрисати средњим кликом.

## Глава 5

### Анализа

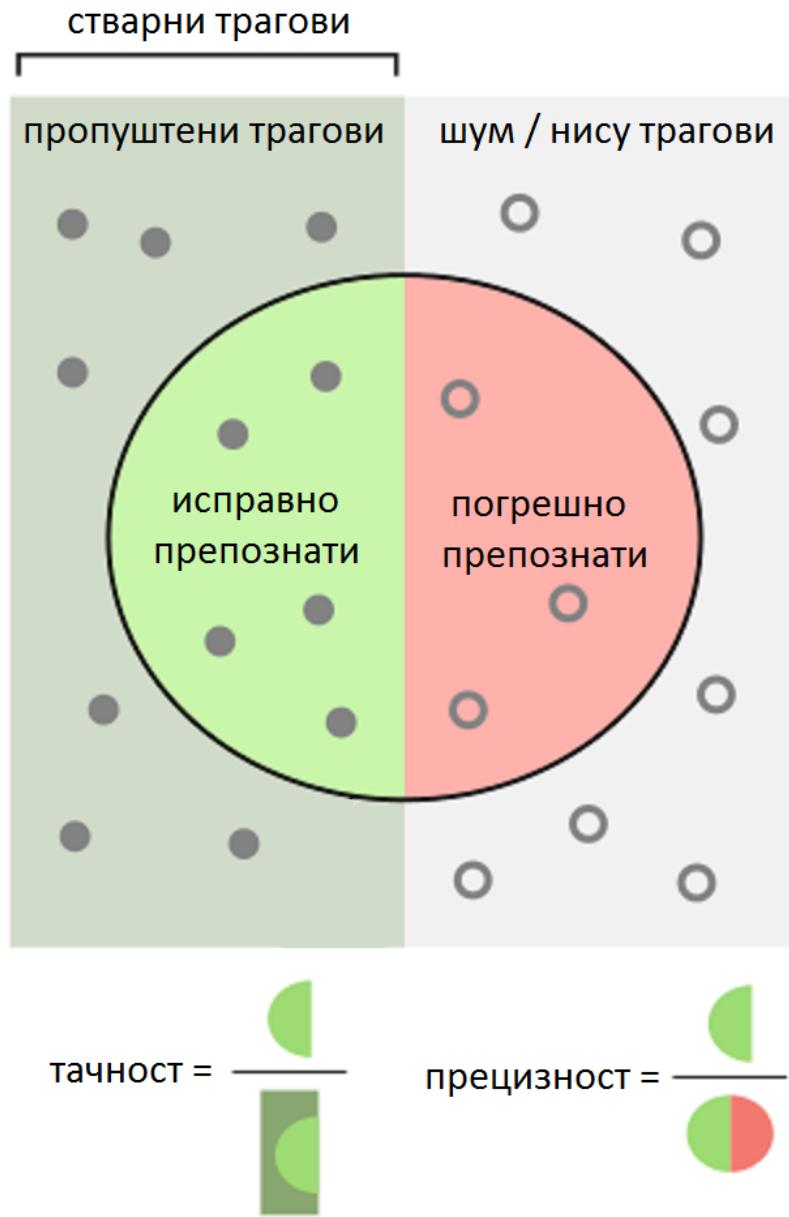
За мерење и вредновање детекције трагова неутронске дозиметрије генерисана су четири тест скупа по сто слика. Број трагова, њихова позиција и величина генерисани су насумице у задатом опсегу за сваку тест слику. Сваки тест скуп прављен је налик на репрезентативне примере пронађене на вебу, као и оне добијене од стране истраживача. Први тест скуп слика садржи светле трагове на тамној подлози. Други тест скуп садржи слике са тамним траговима који имају сенку у средњем делу. Трећи и четврти скуп садрже тамне трагове на светлијој позадини, с тим да је контраст интензитета боје између трагова и позадине већи у трећем скупу слика.



Слика 5.1: Примери слика из тест скупова: (горе-лево) први тест скуп, (горе-десно) други тест скуп, (доле-лево) трећи тест скуп и (доле-десно) четврти тест скуп.

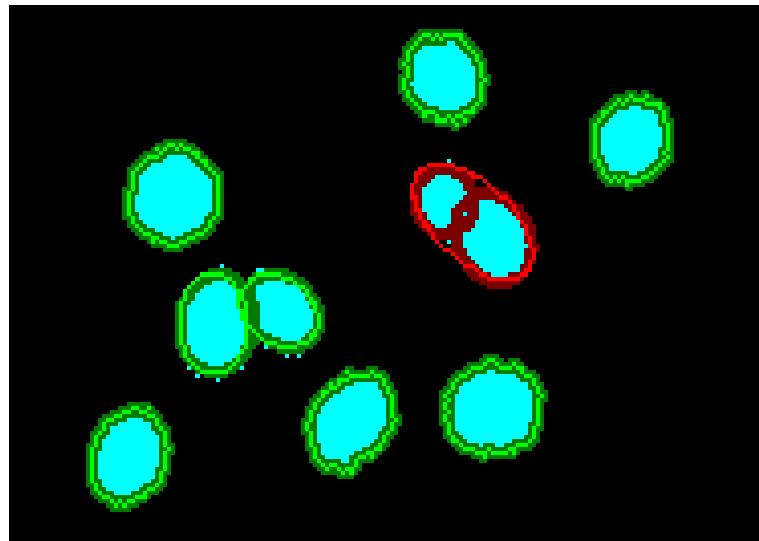
## 5.1 Резултати

Као метрике за вредновање користе се *тачност* и *прецизност* система. *Тачност* представља однос исправно детектованих трагова према стварном броју трагова, док *прецизност* представља однос исправно детектованих трагова према укупном броју детектованих трагова.



Слика 5.2: Тачност и прецизност

Написан је и програм који за сваку слику учитава податке о генерисаним траговима, извршава детекцију трагова и врши поређење. Резултат рада програма су *тачност* и *прецизност* на новоу целог тест скупа, а такође и на нивоу сваке појединачне тест слике, уз генерисану слику са назаченим грешкама ради лакше анализе.

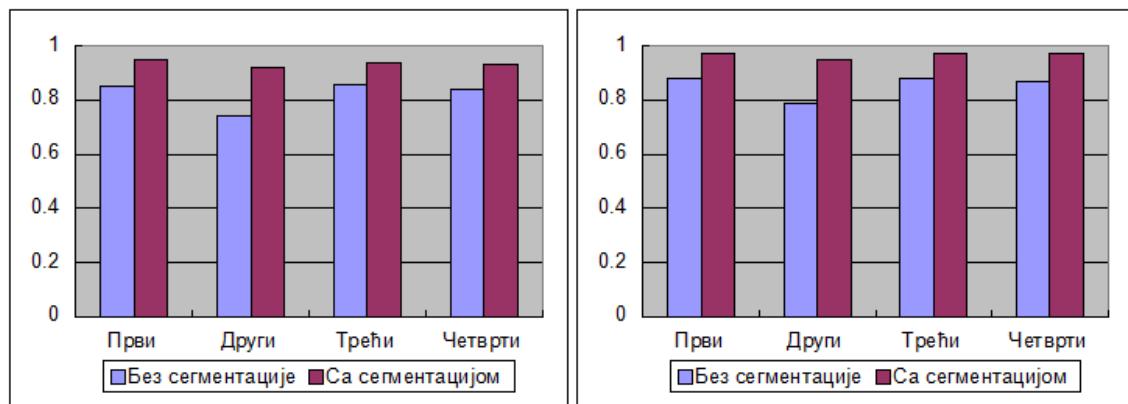


Слика 5.3: Анализа слике: зелено означени трагови су препознати исправно, црвено означени трагови су препознати погрешно (светло црвено означава шта је детектовано, а тамно црвено стварне трагове)

При мерењу тачности и прецизности одабран је *Otsu* аутоматски праг за бинаризацију, а остала подешавања попут изоштравања слике нису укључена.

Табела 5.1: Тачност и прецизност софтвера за детекцију неутронске дозиметрије

Тест скуп	Број слика	Број трагова	Без сегментације		Са сегментацијом	
			Тачност	Прецизност	Тачност	Прецизност
Први	100	6437	0.85	0.88	0.95	0.97
Други	100	4418	0.74	0.79	0.92	0.95
Трећи	100	6349	0.86	0.88	0.94	0.97
Четврти	100	6297	0.84	0.87	0.93	0.97

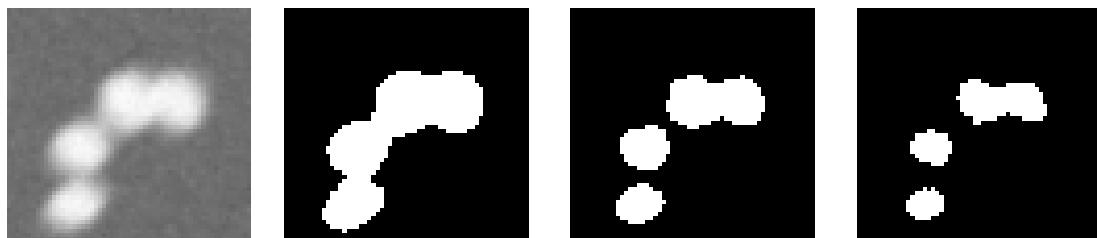


Слика 5.4: Утицај сегментације на тачност (лево) и прецизност (десно)

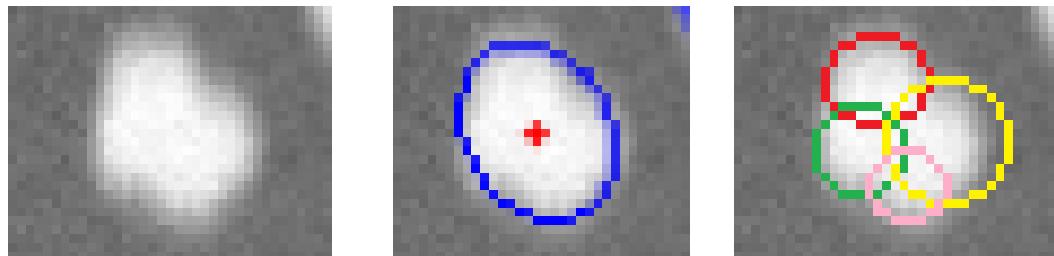
## 5.2 Најчешће грешке

Најчешће грешке у детекцији трагова неутронске дозиметрије приказане су на сликама 5.9, 5.5, 5.6, 5.7 и 5.8 . Грешке углавном настају када се два или више трагова превише преклапају. У једном делу таквих случајева промена прага бинаризације може помоћи. Међутим, код других трагова то може довести до тога да детектовани траг буде значајно мањи од реалног трага. Код дела преклопљених трагова на којима детектор греши, границе трагова нису видљиве ни за људско око. Код преклопљених трагова проблем је и процена величине трага, тј. део трага је сакривен иза једног или више других трагова и не изврши се задовољавајућа процена његове величине.

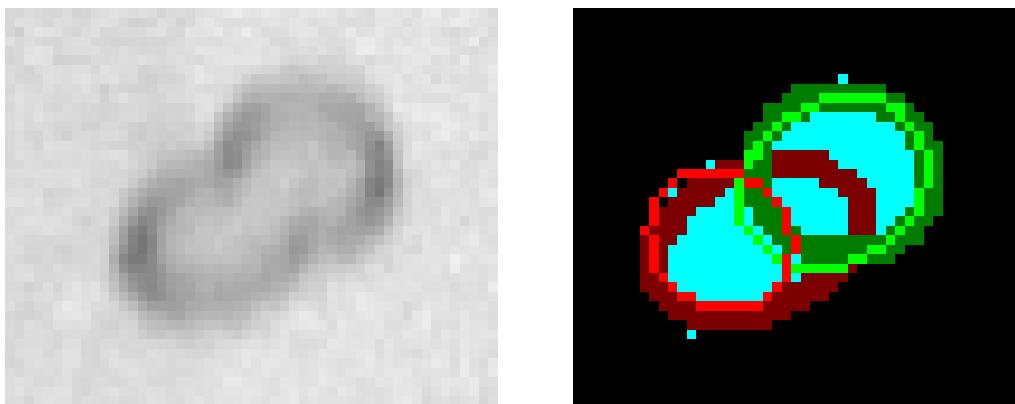
Додатно, код трагова са сенком у средишњем делу долази до тога да је траг, односно прстен трага сувише танак и да се сенка споји са позадином, чинећи да контура трага не буде затворена и за последицу препозната као више мањих трагова. Код преклапајућих трагова који имају сенку у средишњем делу није искоришћено само то својство да сенка може указати на центар трага.



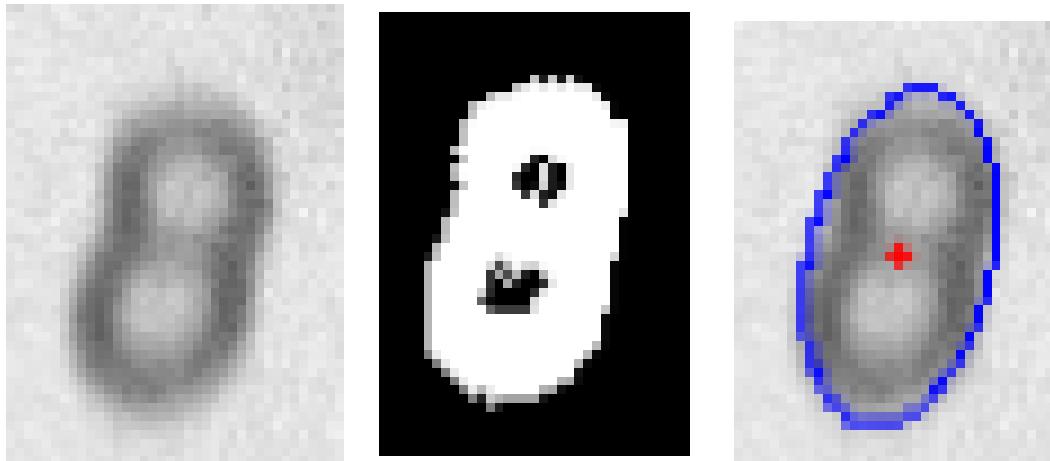
Слика 5.5: Код сегментације преклопљених трагова праг бинаризације битно утиче на исход. (i) Сиво скалирана слика. (ii) *Otsu* праг бинаризације (у датом случају 170) није увек одговарајући. (iii) У датом случају промена прага бинаризације на 190 даје боље резултате. (iv) Превише висок праг бинаризације (210) узрокује да детектовани трагови буду значајно мањи од реалних.



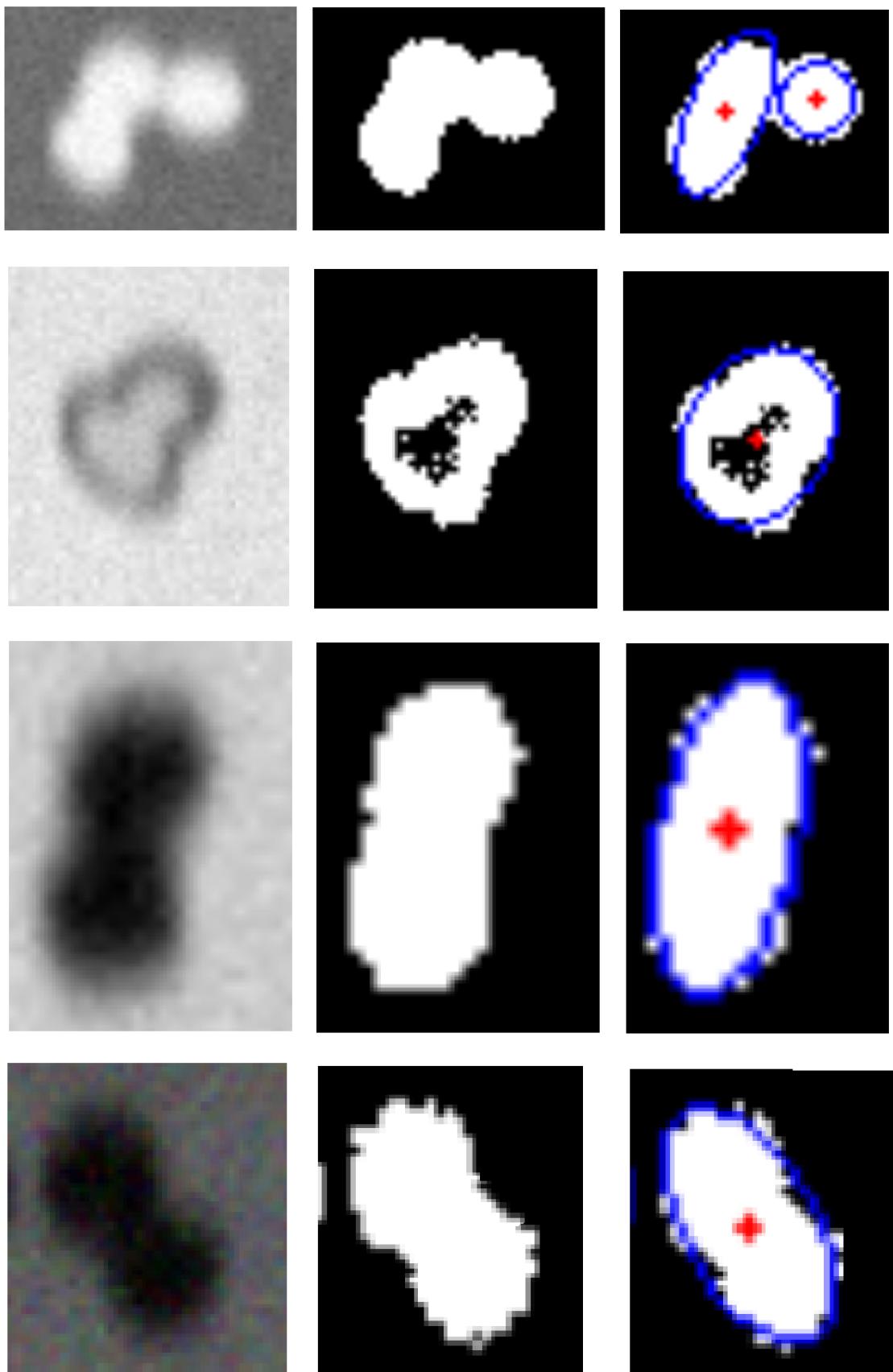
Слика 5.6: Грешка када се преклапа пуно трагова који формирају један велики траг. (i) Сиво скалирана слика. (ii) Детектовани траг. (iii) Генерисани трагови.



Слика 5.7: Код преклопљених трагова тешко се одређује права величина трага. (i) Сиво скалирана слика. (ii) Слика из анализе: Зелено означени траг је исправно препознат. Црвено означени траг се не рачуна као исправно детектован траг. Светло црвени је детектовани, а тамно црвени генерисани траг.



Слика 5.8: Грешка при сегментацији преклопљених трагова где би податак о сенци у средини трага могао да помогне при детекцији. (i) Сиво скалирана слика. (ii) Бинарна слика. (iii) Детектовани траг.



Слика 5.9: Честа грешка при детекцији преклопљених трагова: засебни трагови се могу разазнati људским оком, али се на бинарној слици губе пиксели који би на то указали

### 5.3 Дискусија

Сумирајући наведене резултате у табели 5.3 и посматрајући слике најчешћих грешака 5.5 - 5.9 можемо извести неколико закључака.

Детекција трагова неутронске дозиметрије има веома добру тачност ( $\sim 93.5\%$ ) и високу прецизност ( $\sim 97\%$ ). Успешност сегментације преклопљених трагова је добра (50-70%), поготово ако се узме у обзир да добар део ових проблема није решив ни за људско око.

Такође, важно је напоменути да није увек погодно користити подразумевана подешавања. На пример, *Otsu* праг бинаризације не даје увек најбоље решење, те је стога потребно прилагодити праг бинаризације. Проблем са бинаризацијом може се решити или ублажити увођењем адаптивне бинаризације која би тражила најбољи праг бинаризације на мањем прозору уместо на целој слици. У том случају, минимални и максимални дијаметар трага би морали строжије да се задају, односно могла би и да се уведе процена величине трагова пре саме детекције.

## Глава 6

### Закључак

Циљ овог рада је био да понуди решење које ће аутоматизовати и убрзати мерење густине трагова неутронске дозиметрије траг детектора *Cr-39*. Тренутно тај процес се врши визуелном проценом броја трагова помоћу микроскопа.

Као резултат рада направљен је софтверски пакет који обрађује слике неутронске дозиметрије и врши процену броја трагова, њихову позицију и величину. Такође, представљена је тачност и прецизност овог решења, као и најчешће ситуације у којима настају грешке. Програм омогућава промену подешавања обраде слика да би омогућио постизање бољих резултата.

Могу се навести могућности за даљи развој. Уколико би се рад наставио на сликама веће резолуције, систем би се могао унапредити да даје боље процене величине код преклопљених трагова. Може се додати детекција типа трага, односно детекција угла под којим је неутрон прошао кроз материјал и оставио траг. Затим, на основу угла, величине и интензитета трага може се извршити процена енергије неутрона који је оставио траг.

# Библиографија

- [1] James Turner, *Atoms, Radiation, and Radiation protection*. Wiley-VCH, New York, 2007.
- [2] Биљана Миленковић, *Примена детектора CR-39 у детекцији и дозиметрији неутрона*, Докторска дисертација, Природно-математички факултет, Универзитета Крагујевцу, Крагујевац, 2013.
- [3] <http://opencv.org>, OpenCV званична веб страна
- [4] Shapiro and Stockman, *Computer Vision*, Prentice-Hall, 2001
- [5] Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010
- [6] <http://qt-project.org/>, Qt званична веб страна