

# 1. Programmatūra

### 1.1. Programmatūras veicamie uzdevumi

Rentģenu datortomogrāfijas uzņērumi ir vairāku pelēktoņu 2D attēlu kopums, kur katrs no attēliem reprezentē vienu izmeklējamā objekta slāni. Kaut arī tomogrāfijas uzņērumi satur 3D informāciju, tie tiešā veidā nav piemēroti 3D attēlošanai (3D ekrāni, 3D objektu 2D projekciju attēlošana 2D ekrānos). Šī iemesla dēļ ir izstrādāta programmatūra, kuras galvenais uzdevums ir: izmantojot datortomogrāfijas uzņēmumus esošo 3D informāciju, rekonstruēt plaušu asinsvadu tīkla 3D ģeometriju un saglabāt to standartizētā 3D ģeometrijas failu formātā. Šo uzdevumu var sadalīt sekojošos apakšuzdevumos:

- Datu ieguve no datortomogrāfijas uzņēmumu failiem (.mhd, .raw);
- Plaušas asinsvadu segmentēšana datortomogrāfijas uzņēmuma slāņos;
- Plaušas asinsvadu 3D ģeometrijas rekonstruēšana;
- Plaušas asinsvadu 3D ģeometrijas saglabāšana standart formāta failos (Wavefront .obj);
- Datortomogrāfijas slāņu 2D vizualizācija;
- Rekonstruēto asinsvadu 3D vizualizācija;
- Grafiskās lietotāja saskarnes izveide.

## 1.2. Programmatūras izstrādes valoda un esošu bibliotēku izmantošana

Par programmatūra izstrādes valodu tika izvēlēta JAVA programmēšanas valoda. Galvenie kritēriji kādēļ tika izvēlēta šī valoda bija: programmatūras spēja darboties uz dažādām platformām, esošu bibliotēku izmantošanas ērtība, bezmaksas izstrādes rīku pieejamība, kā arī projektā iesaistīto cilvēku pieredze ar šo valodu.

Lai atvieglotu programmatūras izstrādi tika meklētas esošas bibliotēkas, kas spētu veikt daļu no nepieciešamajiem uzdevumiem. Kā prasība šīm bibliotēkām tika izvirzīta tas, ka šīm bibliotēkām jābūt tīrai JAVA implementācijai, lai nebūtu problēmu uzstādīšanai uz dažādām platformām.

Datu ieguvei parasti tiek izmantoti DICOM standarta rīki. Ir vairākas bibliotēkas, kas piedāvā DICOM standarta rīkus, kuru salīdzinājums atrodams šeit [1]. Mums bija pieejami tomogrāfijas uzņēmumi no VESSEL12

sacensībām kuri bija pieejami MetaImage formātā, kas sastāv no diviem failiem: teksta faila, kas satur aprakstošo informāciju pat datortomogrāfijas uzņēmumu (paplašinājums .mhd) un bināru datu failu (paplašinājums .raw). Diemžēl nevienā no pieejamām bibliotēkām, kurai būtu pieejama laba dokumentācija, funkcionalitāte šo failu nolasīšanai netika atrasta, tādēļ šī daļa tika implementēta patstāvīgi.

Asinsvadu segmentēšanas daļai bija nepieciešama lineārās algebras bibliotēka. Galvenās prasības šai bibliotēkai bija ātrdarbība, matricu operāciju pieejamība, laba matricas datu struktūras implementācija ar funkcionalitāti ērtai un elastīgai matricu elementu datu uzstādīšanai un piekļūšanai, kā arī labas dokumentācijas esamība. Populārāko lineārās algebras bibliotēku ātrdarbības salīdzinājums ir salīdzināts šeit - [2]. Vadoties pēc iepriekš minētajiem kritērijiem tika izvēlēta EJML[3] bibliotēka.

Asinsvadu segmentēšanas algoritmam bija nepieciešamas arī vairākas datorredzes pielietojumiem specifiskas attēlu apstrādes operācijas: 2D konvolūcija, Gausa attēlu piramīdas aprēķināšana, attēlu izmēru maiņa ar bikubisko interpolāciju un integrālattēla iegūšana. Šīs operācijas parasti atrodamas standarta datorredzes nozares bibliotēkās. Kā populārāko var minēt OpenCV[4], kas ir atvērtā koda bibliotēka ar visai labu atbalstu, un pieejamu interfeisu vairākām programmēšanas valodām: C++, Java, Python, Matlab. Kaut arī bibliotēkai ir pieejams interfeiss Java programmēšanas valodai, bibliotēkas pirmkods ir izstrādāts C++ valodā un, lai lietotu šo bibliotēku jāveic platformas atkarīgs bibliotēkas būvēšanas process un tā kā sākotnēji izvirzījām uzstādījumu izmantot tīras Java bibliotēkas, tad izvēlējamies šo bibliotēku nelietot un šīs operācijas tika implementētas patstāvīgi.

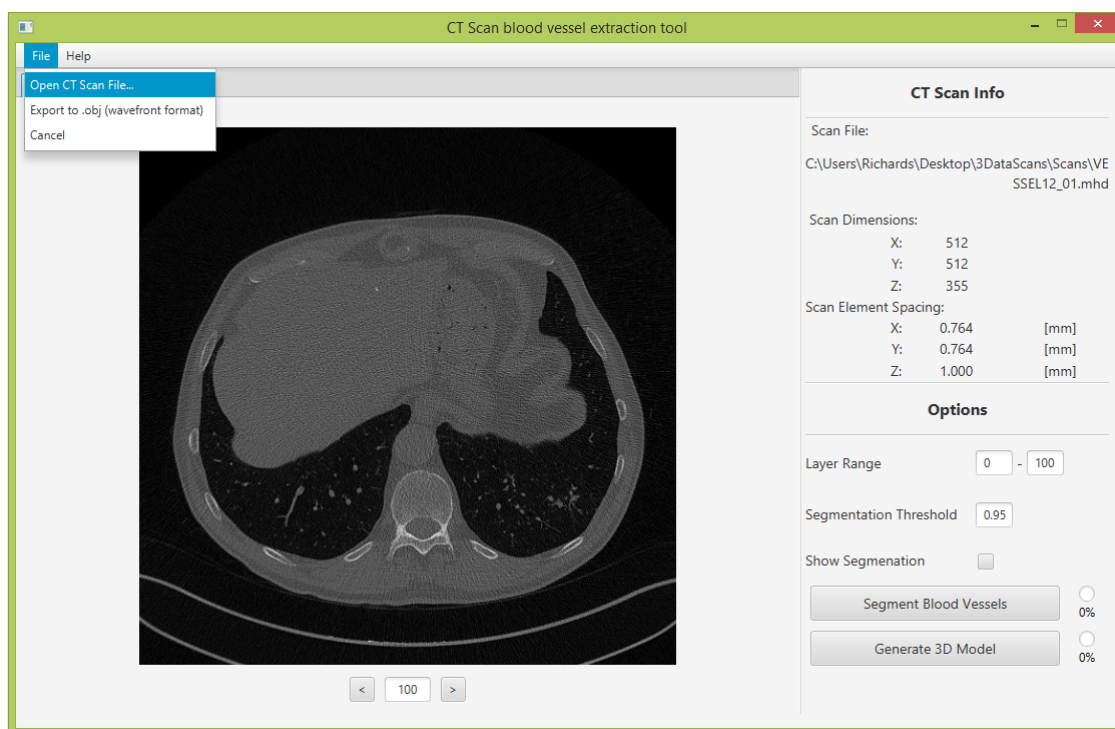
Tā kā implementētais asinsvadu segmentēšanas algoritms bija visai lēns (8 sekundēm vienam slāņa apstrādei), tika apskatīta iespēja izmantot OpenCV bibliotēku ar CUDA atbalstu, kas nodrošina vairāku attēlu apstrādes operāciju izpildīšanu uz grafiskās kartes, kas ļauj iegūt ātrāku operāciju izpildi (beigu implementācijā apstrādes laiks samazinājās līdz 2.3 sekundēm vienam slānim). Šim nolūkam asinsvadu segmentēšanas algoritms tika implementēts C++ valodā izmantojot OpenCV bibliotēkas funkcijas attēlu apstrādes operācijām, kuras tiek izpildītas uz grafiskās kartes. Lai tiktu izmantots iepriekš izstrādātās programmatūras daļas, tad izmantojot JNI (Java Native Interface), tika izveidots interfeiss, kas ļāva izsaukt no Java programmas izsaukt C++ valodā implementēto asinsvadu segmentēšanas algoritmu.

Asinsvadu tīklojuma 3D ģeometrijas konstruēšanai tika izmantots maršējošo kubu algoritms, kuram gatavas implementācijas Java valodā ar labu dokumentāciju atrasta netika, tādēļ šis algoritms tika pašu implementēts.

Grafiskās lietotāja saskarnes izveidei tika izmantota Java standarta bibliotēkas platforma JavaFX, kas nodrošina arī 3D vizualizāciju.

### 1.3. Lietotāja grafiskās saskarnes apraksts

Programmas logs ir apskatāms 1. attēlā. Programma sastāv no vairākiem laukiem. Datu vizualizācijas lauks ir paredzēts datortomogrāfijas slāņu vizualizācijai, kā arī asinsvadu tīklojuma 3D modeļa vizualizācijai. To, kas tiek attēlots var pārslēgt ar pogām: 2D View (Datortomogrāfija slāņu skats), 3D View (asinsvadu tīklojuma 3D modeļa skats). 3D modeļa skatā modelis ir aplūkojums tikai tad kad ir segmentēti asinsvadi, kā arī ģenerēts 3D modelis.



1. att. Asinsvadu segmentēšanas programmas logs

## Literatūras saraksts

- [1] DICOM rīku implementācija Java programmēšanas valodai. Interneta resurss - <http://www.schoech.de/diploma/toolkits.html#java>
- [2] JAVA lineārās algebras bibliotēku salīdzinājums. Interneta resurss - <https://code.google.com/p/java-matrix-benchmark/wiki/RuntimePerformanceBenchmark>

- [3] Lineārās algebras bibliotēkas - EJML mājaslapa: [http://ejml.org/wiki/index.php?title=Main\\_Page](http://ejml.org/wiki/index.php?title=Main_Page)
- [4] Datorredzes bibliotēkas OpenCV mājaslapa: <http://opencv.org/>