



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

PODPŮRNÉ ZPRACOVÁNÍ TOMOGRAFICKÝCH DAT PRO 3D SLICER

SUPPORTING PROCESSING OF TOMOGRAPHIC DATA FOR 3D SLICER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Dašek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vratislav Čmiel, Ph.D.

BRNO 2023

Diplomová práce

magisterský navazující studijní program **Bioinženýrství**

Ústav biomedicínského inženýrství

Student: Bc. Filip Dašek

ID: 195729

Ročník: 2

Akademický rok: 2022/23

NÁZEV TÉMATU:

Podpůrné zpracování tomografických dat pro 3D Slicer

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte rešerši v oblasti vizualizačních programů pro vizualizaci a segmentaci medicínských objemových dat. 2) Seznamte se s programovým vybavením 3D Slicer a jeho funkcemi. Seznamte se také s možnostmi využití podpůrného skriptování v jazyce Python ve 3D Sliceru za účelem zpracování tomografických dat. 3) Navrhněte vhodné postupy s využitím skriptování ve 3D Sliceru, které pomohou usnadnění předzpracování a segmentace tomografických dat. 4) Realizujte programový doplněk pro 3D Slicer pro předzpracování tomografických dat pro usnadnění segmentace. 5) Realizujte programový doplněk pro 3D Slicer pro podpůrnou automatizovanou segmentaci zvolených oblastí tomografických dat. 6) Ověřte řešení na tomografických datech srdce. 7) Zhodnoťte přínos vlastního řešení a zlepšení automatizace segmentace oproti běžnému řešení.

DOPORUČENÁ LITERATURA:

[1] 3D Slicer's Documentation. Dostupné z: <https://slicer.readthedocs.io/en/latest/>

[2] Slicer Introduction. Dostupné z:

https://chestimagingplatform.org/files/chestimagingplatform/files/slicerintroduction_tutorial.pdf

Termín zadání: 6.2.2023

Termín odevzdání: 22.5.2023

Vedoucí práce: Ing. Vratislav Čmiel, Ph.D.

doc. Ing. Radim Kolář, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práce shrnuje dostupné programy pro vizualizaci a segmentaci biomedicínských dat. Hlavní zaměření je na open source software 3D Slicer, u kterého jsou popsány všechny hlavní moduly a dostupné nástroje pro zpracování a segmentaci 3D obrazů. Čtenář je také seznámen s možností vytváření nových modulů a funkcí ve Sliceru za použití programování v pythnu. V praktické části je vytvořen modul pro rychlou a efektivní segmentaci srdce a jeho částí pomocí metody Watershed. Modul obsahuje nástroje pro předzpracování CT snímků a automatický post-processing vysegmentovaných částí srdce. Grafické rozhraní bylo vytvořené pomocí QT designeru. V testovací části jsou prezentovány výsledky segmentací. Při stejných vstupech dosahuje segmentace pomocí vytvořeného modulu lepších výsledků, než nabízená možnost ve 3D sliceru.

KLÍČOVÁ SLOVA

Segmentace, CT snímky, 3D Slicer, Open source Software, DICOM, biomedicínské obrazy, 3D modely, Watershed, Předzpracování obrazu

ABSTRACT

Překlad abstraktu Thesis summarize available softwares for visualization and segmentation of medical data. Main focus is on open source software 3D Slicer, where all main modules and functions for processing and segmentation of 3D images are described. The reader is also acquaint with possibility to create new modules and function in Python. In practical part of thesis new modul is created. Modul is focused on fast and effective segmentation of heart and their parts by using watershed method. Part of the module are tools for preprocessing of CT images and automatic post-processing. GUI was implemented and created with QT designer. Thesis contains testing and presenting segmented data. Segmentation by presented modul generally achieves better results then options offered by 3D Slicer. The last part of the thesis presents a proposal for utilizing supporting functions to achieve fast and effective segmentation of the heart and their parts.

KEYWORDS

Překlad klíčových slov Segmentation, CT image, 3D Slicer, Open source Software, DICOM, Medical imaging, 3D models, Watershed, Preprocessing

DAŠEK, Filip. *Podpůrné zpracování tomografických dat pro 3D Slicer*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství , 2023, 51 s. Diplomová práce. Vedoucí práce: Ing. Vratislav Čmiel, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Bc. Filip Dašek
VUT ID autora: 195729
Typ práce: Diplomová práce
Akademický rok: 2022/23
Téma závěrečné práce: Podpůrné zpracování tomografických dat pro 3D Slicer

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

* Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval za všechny rady a vedení svému vedoucímu diplomové práce panu Ing. Vratislavu Čmiel. Také bych rád poděkoval svému kamarádovi Jonáši Prokopovi, který si našel čas a pomohl s revizí diplomové práce.

Obsah

Úvod	1
1 Rešerše vizualizačních programů	2
1.1 Osiris MD	2
1.2 ImageJ	3
1.3 XNAT	3
1.4 Seg3D	5
1.5 ITK-SNAP	6
1.6 FreeSurfer	7
1.7 MedInria	8
1.8 MIPAV	9
1.9 Dragonfly	10
2 3DSlicer	11
2.1 Moduly	12
2.2 Extensions	15
2.2.1 Medical Reality Modeling Language	16
2.3 Uživatelské rozhraní	17
2.4 Segmentace	19
2.4.1 Segmentační nástroje Dostupné v 3D Sliceru	20
2.4.2 Metoda rozvodí	21
3 Vytvoření a implementace Modulu	
HeartSegmentation	24
3.1 Vytvoření Modulu	24
3.2 Vytvoření uživatelského rozhraní	24
3.2.1 Připojení a nastavení Widgetů	25
3.2.2 GUI	26
3.3 Předzpracování obrazu	29
3.3.1 Filtrace	29
3.3.2 Oříznutí a podvzorkování obrazu	29
3.3.3 Window/level	30
3.4 Segmentace	31
3.4.1 Gradient filtr	34
3.4.2 Watershed Segmentace	35

4 Testování modulu	38
4.1 Testování segmentace na datech	38
4.2 Diskuze	40
Závěr	42
Literatura	44
Seznam symbolů a zkratk	47
Seznam příloh	48
A Ukázky zdrojového kódu	49
B Elektronická příloha	51
B.1 HeartSegmentation.py	51
B.2 Heart-WatershedSegmentation	51
B.3 PřidáníModulu.docx	51
B.4 Segmentace	51
B.5 Watershed.png	51

Seznam obrázků

1.1	XNAT architektura	4
1.2	Příkad zobrazení kalosního tělesa(lat. Corpus Callosum) segmentovaný metodou aktivních kontur	6
1.3	Uživatelské rozhraní FreeSurfer - segmentace mozku	7
1.4	Vizuálizace nervových drah - traktografie	9
1.5	MIPAV - uživatelské rozhraní, 3D rendering	10
2.1	Diagram práce s DICOM daty ve Sliceru[15]	13
2.2	ukázka možné struktury MRLM souboru	16
2.3	Uživatelské rozhraní	17
2.4	a) binární reprezentace, b) uzavřený povrch,c)pravděpodobnostní mapa, d)rovinné kontury, e) Ribbon model [19]	19
2.5	Vlevo: vstup v transverzální rovině, Vpravo: výsledná segmentace vytvořená 3d slicerem metodou watershed	22
2.6	Vlevo: vstup ve 3d pro metodu Watershed, Vpravo: výsledná segmentace levé síně.	22
3.1	GUI při spuštění modulu	26
3.2	GUI: Preprocessing	27
3.3	GUI: Segmentation	28
3.4	GUI: Generating a Mask	29
3.5	Error Window: Chybí input pro Source Volume	32
3.6	BoundingBox kolem vstupů a kolem vysegmentovaného srdce	33
3.7	Ukázka nevhodně zvolených vstupů s vygenerovaným segmentem pozadí	33
3.8	GradientMagnitude filtr (zleva do prava): 3D zobrazení segmentace, zobrazení segmentace v transverzální rovině, parametrický obraz získaný filtrem	34
3.9	SobelEdge filtr(zleva do prava): 3D zobrazení segmentace, Zobrazení segmentace v transverzální rovině, parametrický obraz získaný filtrem	34
3.10	Parametrické obrazy získané filtrem GradientMagnitude: Sigma = 1.8(vlevo) ,Sigma = 2.4(střed), Sigma = 3.0 (vpravo)	35
4.1	CT snímek hrudníku	38
4.2	Segmentace srdce s nedokončenou segmentací levé komory	39
4.3	Rozdíl mezi zadanými vstupy pro jednotlivé části srdce: Modrá - pravá síň, Žlutá - pravá komora, Červená - Levá síň, Zelená - Levá komora	39
4.4	Porovnání výsledků se segment editorem: Levá komora (vlevo), Pravá komora (vpravo)	40
4.5	Porovnání výsledků se segment editorem: Pravá síň (vlevo), Levá síň (vpravo)	40

4.6	Porovnání výsledků se segment editorem: Levá síň (vlevo), Levá komora (vpravo),	41
-----	---	----

Seznam tabulek

2.1	Formáty obrazových dat	11
2.2	Formáty modelů	11

Úvod

Rychlá digitalizace probíhá ve všech medicínských odvětvích, a proto je vysoká poptávka po nových technologiích pro zpracování, vizualizaci a archivaci medicínských dat. Digitalizace je považována za jeden z hlavních komponentů pro zlepšení a zrychlení služeb v lékařství a výzkumu. Tato práce se zaměřuje na možnosti rozšíření open source medicínských aplikací pro vizualizaci a segmentaci obrazu. V práci jsou zdokumentovány a popsány základní nástroje, které medicínské aplikace používají pro vizualizaci a segmentaci. U vytváření segmentací anatomických struktur se potkáváme s mnoha výzvami. Různorodnost modalit a různá nastavení snímání komplikují automatizaci mnoha algoritmů. Důležitou součástí mnoha algoritmů je také předzpracování obrazu a editace získaných segmentů. Podrobnější popis nástrojů a struktury byl věnován aplikaci 3D Slicer, ve které byl následně navržen modul *Help-Segmentation*. Modul byl vytvořen pomocí základní šablony, kterou nabízí 3D slicer a jeho funkcionality je definována pomocí python skriptu. 3D Slicer je open source aplikace, a tudíž je přístup ke zdrojovým kódům všech funkcí volně dostupný. Jádro 3D sliceru je psáno v jazyce C++, ale díky MRML architektuře jsou i funkce modulů psaných v C++ dostupné a použitelné v pythnu. Rozšiřování 3D sliceru pomocí python skriptu má obrovský potenciál, ale pro správnou funkcionality je nutná znalost MRML architektury. Modul se zaměřuje na předzpracování obrazu, aplikování segmentace na části srdce a post-processing získané segmentace. Vytvořené uživatelské rozhraní bylo vytvořeno pomocí QT designeru. Pro získání segmentace je v práci využit watershed filtr s přednastavenými vstupy značící minima v obraze. Součástí modulu je konstruktor pro vytváření nových segmentací a segmentů a algoritmus pro automatické vytváření segmentu pozadí. Nastavení funkcí je optimalizováno na segmentaci srdce a jeho částí. V kapitole 4 jsou prezentovány výsledky segmentace a porovnání s watershed metodou přítomnou ve 3D sliceru. Modul má také prostředky pro vytváření dat určená k učení umělých inteligencí. Cílem práce je vytvořit podpůrné rozšíření do aplikace 3D Sliceru pro semi-automatickou segmentaci 3D snímků, která zrychlí pracovní postup uživatele.

1 Rešerše vizualizačních programů

V této části je popsáno několik softwarů používající se pro vizualizaci a zpracování biomedicínských obrazů. Mezi hlavní podmínky patří, aby tento software uměl pracovat s *Digital Imaging and Communications in Medicine* (DICOM) formátem. DICOM je standardem pro snímky pořízené pomocí CT, MRI a ultrazvukem, a proto je naprosto esenciální, aby aplikace uměli s tímto formátem pracovat. Všechny zmíněné softwary mají základní nástroje pro zobrazení a zpracování načtených dat (registrace obrazu, filtrace apod.). Nezabíráme se zde metodami, které tyto softwary využívají pro svoje nástroje, pouze hrubým výčtem funkcí a jejich použití, které softwary nabízejí. Vzhledem k tomu, že je velká většina softwarů open source, jsou zmíněny i možnosti rozšiřování nabízených nástrojů pomocí modulů a plug-inů.

1.1 Osiris MD

OsiriX je jeden z nejrozšířenějších programů pro prohlížení a zpracování medicínských dat. Používá se především pro DICOM data ale podporuje i mnoho dalších formátů jako LSM soubory, BioRadPIC, TIFF, ANALYZE, PNG, PDF, AVI, MPEG a mnoho dalších. Nabízí zobrazení a zpracování 2D, 3D a 4D dat, podporuje PET-CT a SPECT-CT a zcela integruje PACS (umožňuje zobrazení, správu a ukládání obrazových dokumentů). Jako programovací jazyk využívá nastavbu jazyka C Objectiv-C. OsiriX je podporován v operačním systému macOS / OS X, není kompatibilní s Windows a Linux, což může omezovat jeho využití. OsiriX začal jako open source (zdrojové kódy jsou volně k dispozici), ale nyní už tomu tak není, zdarma je pouze demo verze OsiriX Lite a pro plný přístup ke všem funkcím je zapotřebí zakoupit měsíční nebo roční předplatné. Součástí je i OsiriX Cloud, který umožňuje archivaci a sdílení dat s ostatními uživateli. Pro vytváření vlastních pluginů je zapotřebí aplikace Xcode. Výhodou může být i volitelnost používaného jazyka jako angličtina, němčina, portugalština, čínština, francouzština a další.

Zobrazování dat

Krom základních prvků pro zobrazování medicínských obrazů jako škálování, rotace, měření úhlu a plochy je schopen extrahovat a zobrazovat statistická data z 2D a 3D *region of interest* (ROI): průměr, min, max, s-dev a histogramy. OsiriX také umožňuje fúzi multimodálních obrazových dat: PET-CT, PET-MR, SPECT-CT a okamžitý výpočet SUV (koncentrace radiofarmaka ve tkáni). Součástí jsou také specifické pokročilé výpočty pro data pořízené různými metodami jako výpočet VTI (ang. Vertebral Tortuosity Index), gradient tlaku, rychlost (pro CT), výpočet T2 času, ejekční frakce a další.[1]

Zpracování dat

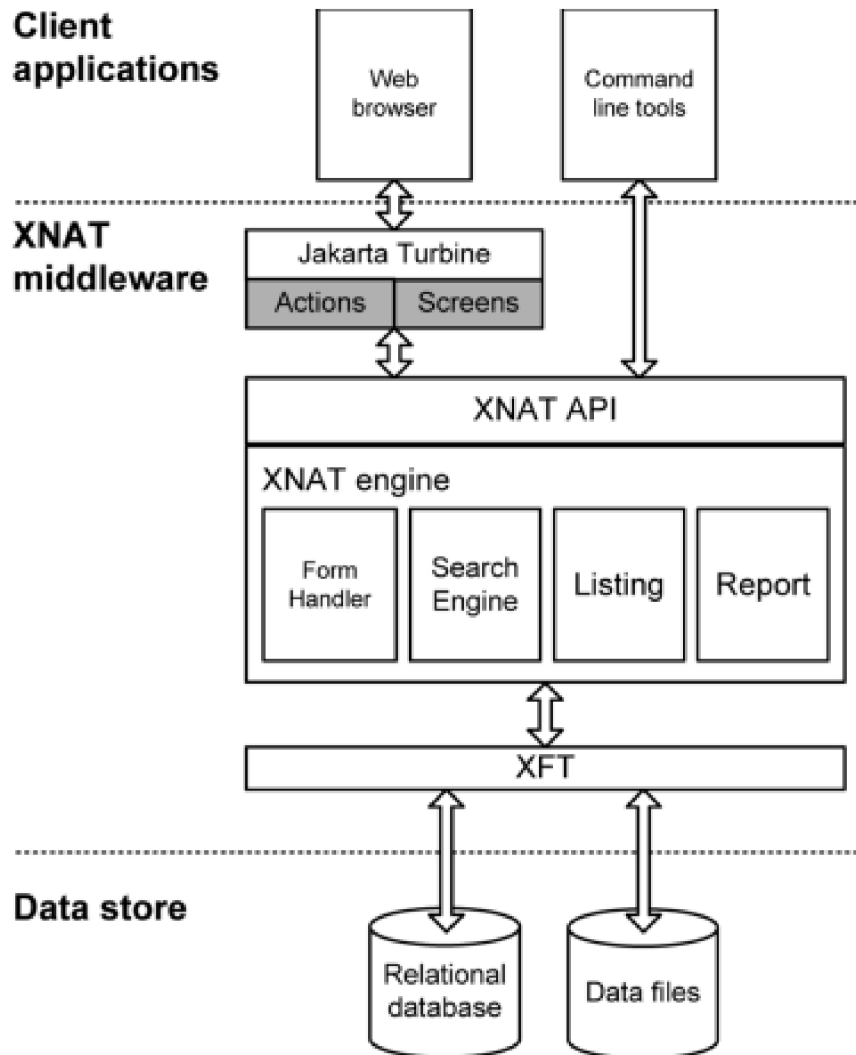
OsiriX má mnoho nástrojů pro vykreslení a zpracování 3D objektů: Multiplanární rekonstrukce, rekonstrukce křivek, vykreslení 3D objemu a plochy, 3D endoskopie, 3D rigidní registrace, vytvoření FlyThru filmu v 3D Volume Rendering nebo 3D Surface Rendering Součástí je i převedení do STL formátu pro 3D tiskárny[1]

1.2 ImageJ

ImageJ je open source software pro zpracování vědeckých obrazů. Zobrazování DICOM souborů a obecně multidimenzionálních souborů je limitováno, a proto je lepší pro medicínské obrazy používat rozšířené verze ImageJ2 nebo Fiji – v textu budeme na tyto rozšíření odkazovat společným názvem ImageJ. ImageJ je podporován pro Windows, MacOS a Linux. Je využíváno velké množství programovacích jazyků jako Java, JavaScript, Python, Matlab, Ruby, Groovy a další. Lze také využít jazyku ImageJ Macro language, který je přímou součástí ImageJ. Toto je velkou výhodou, protože různé komunity upřednostňují jiné programovací jazyky. Díky této rozmanitosti je podporováno více jak 130 formátů v mikroskopii, medicínských obrazů a příbuzných vědeckých disciplínách.[2] DICOM je v ImageJ přímo podporován a práce s tímto formátem lze dále rozšířit pomocí pluginů. Pluginy můžeme rozdělit do tří kategorií. Standartní PlugIn, PlugInFilter a PlugInTool. PlugInFilter přidává funkci, která zpracovává obrazy postupně po jednotlivých rovinách. PlugInTool přidává nové nástroje. Je možné přidávat i další typy pluginů a definovat nové typy. Dokonce i uživatelské rozhraní je vytvořeno pluginem zodpovědného za vytváření a ukazování oken. Protože se jedná o plugin může uživatel rozhraní upravovat či udělat zcela nové bez potřeby zasahovat do kódu systému. [3]

1.3 XNAT

Extensible Neuroimaging Archive Toolkit (XNAT) je open source webové rozhraní vytvořené ve Washington University oddělením neuroinformatiky. Konkrétně se jedná o spojení relační databáze, webové aplikace a Java technologií. Obsahuje online obrázkový prohlížeč, který podporuje běžné formáty využívané pro snímání v neurologii včetně formátů DICOM nebo Analyze. Jeho hlavní funkcí je nahrávání, archivace, zpracování a bezpečné sdílení medicínských obrazů a dat spojená s jejich výzkumem.[4] Součástí je webové rozhraní, které umožňuje procházení a získávání dat z velkých data setů. XNAT nabízí velké množství nástrojů pro zpracování obrazů. Krom manuálního vytváření masek a ROI nabízí poloautomatické a automatické nástroje pro rychlou segmentaci.



Obr. 1.1: XNAT architektura

Ve spolupráci s firmou NVIDIA došlo k rozšíření podpory machine learning a podpory anotací s pomocí AI (AIAA Tools), které značně zrychlují segmentaci 3D objektů. Pro segmentaci pomocí nástroje Deep Extreme cut je zapotřebí označení alespoň 6 bodů kolem orgánu. Pro automatickou segmentaci nástrojem deepgrow stačí označit pouze jeden bod, segmentace však probíhá pouze ve 2D. V případě že je součástí segmentace i pozadí lze přidat další bod označující pozadí pro zlepšení výsledku.[5] AIAA obsahuje i plně automatickou segmentaci, kdy není zapotřebí žádného označení, pouze výběru odpovídajícího modelu. XNAT je možné rozšiřovat pomocí pluginů, které jsou samostatné balíčky, oddělené od XNAT serveru. Pro vytváření pluginů je zapotřebí znalost programového jazyka Java.

1.4 Seg3D

Seg3D open source nástroj pro objemovou segmentaci a zpracování 3D obrazů vytvořená *National Institutes of Health* (NIH), Center for Integrative Biomedical Computing. Je dostupný pro operační systémy Windows, OS X a Linux. Krom formátu DICOM jsou podporovány všechny formáty podporované Insight ToolKit a také soubory matlab. Programovacími jazyky jsou Python, C++ a matlab. Pro vývojáře je k dispozici okno Python Console, které otevře prostředí pro psaní Python scriptů. Zde se vytvoří balíčky, který umožňují přístup k nástrojům, filtrům, zobrazení a datům. Uživatelské rozhraní se skládá z oken. Pouze tři okna se otevírají automaticky při startu.

- Project Window: Okno je rozděleno na 2 části. První část umožňuje přístup k nastavení právě zvoleného projektu, jako nastavení konkrétních barev pro masky v projektu. Nastavuje se zde také automatické ukládání projektu. Výchozí nastavení je nastaveno na ukládání projektu každých 10 minut. Druhá část obsahuje poznámky k projekt a základní informace o projektu. Umožňuje zobrazit několik uložených projektů, které vznikly během pracovní doby. V tomto okně se také vyskytují nástroje pro export zvoleného projektu nebo uložených sezení.
- Tools Window: Toto okno zobrazuje nástroje, které jsou momentálně k dispozici. Nachází se na stejném místě jako Project Window. Mezi okny se dá libovolně přepínat ve spodní části okna. Nástroje, které jsou používány jsou zvýrazněny oranžovou barvou a neaktivní nástroje jsou minimalizované a šedivé.
- Layer Manager Window: Obsahuje všechny masky a obrazové soubory. V případě že není zvolena složka bude okno prázdné. Po zvolení složky zobrazí všechny masky a obrazy které složka obsahuje. Také obsahuje nástroje pro práci s jednotlivými vrstvami a objemy jako průhlednost, skrytí/odkrytí, zobrazení izoplochy a další.

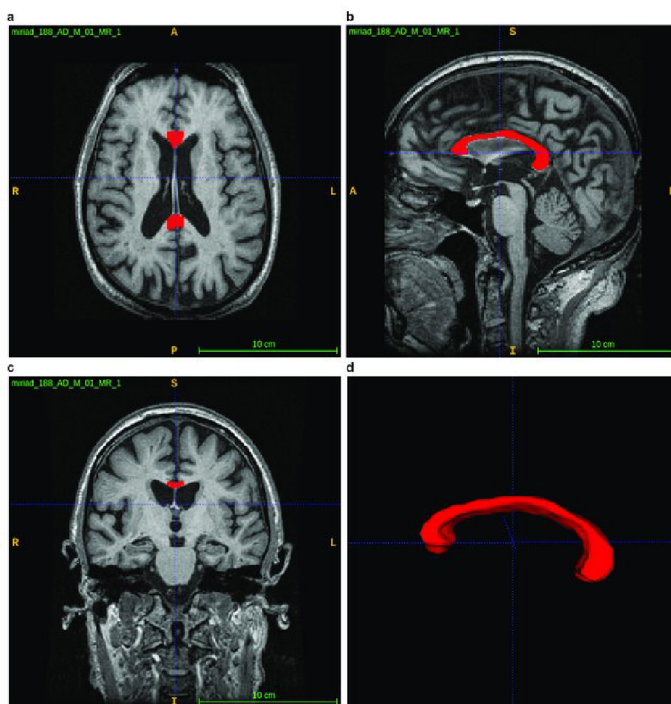
Mezi ostatní okna, které se musí manuálně otevřít, patří: Python Console, Volume View Window, Provenance Window, Controller Window a Message History Window.[6]

Seg3D nabízí poloautomatickou segmentaci a real-time výstup filtrovaní dat. Seg3D obsahuje mnoho různých nástrojů, sloužící k reformátování dat, vyhlazování, odstraňování šumu z obrazu, automatické a manuální segmentaci. Mezi nabízené segmentační nástroje patří PaintBrush, tresholding, Level set segmentace a další. Pro prohlížení malého množství 3D obrazů je Seg3D dostačující, ale pro větší data sety je lepší využívat další CIBC software ImageVis3D. Tato aplikace je schopna procházet terabyty dat a je dostupná i pro mobilní zařízení.

Pro rozšiřování a vytváření nových pluginů ve Windows je nutné mít Win64, Visual studio a Qt. [7]

1.5 ITK-SNAP

ITK-SNAP je další open source software primárně zaměřen na 3D segmentaci obrazů metodou aktivních kontur. Ostatní nástroje pro zpracování obrazu chybí, nebo nejsou příliš vyvinuty. Je dostupný pro uživatele Windows, Mac Os a Linux. Programovým jazykem je C++. Podporuje mnoho biomedicínských formátů včetně DICOM, NIFTI, and Mayo Analyze. Všechny tři roviny jsou po celou dobu ukazovány a jsou propojeny. Pohyb kurzoru v jedné rovině způsobí pohyb v ostatních rovinách. Mezi nabízené nástroje patří zobrazení a navigování v 3D obrazech, manuální značení ROI, kombinování několika segmentací a zpracování ve 2D a 3D.



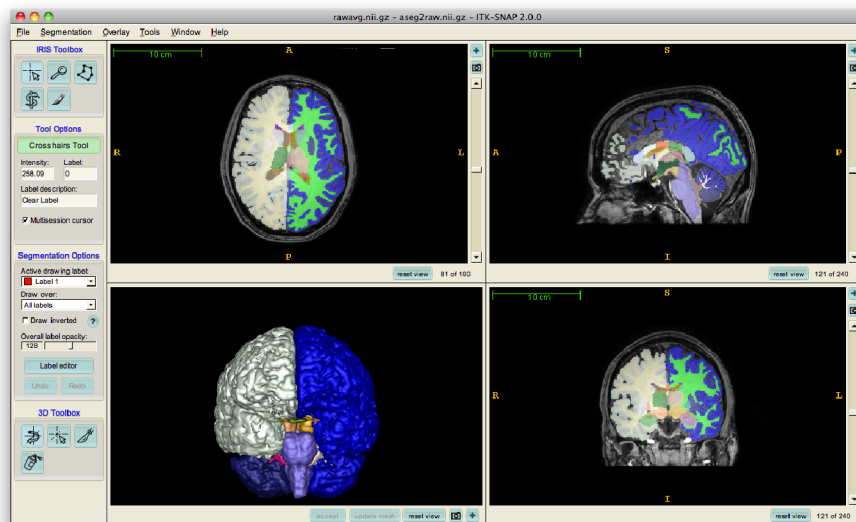
Obr. 1.2: Příklad zobrazení kalosního tělesa(lat. Corpus Callosum) segmentovaný metodou aktivních kontur

Segmentované regiony jsou reprezentovány jako celočíselné 3D masky mající stejnou dimenzi jako původní obraz. Každému voxelu ve vstupním obrazu může být přiřazena jedna nebo žádná značka. Nevýhodou tohoto značení je, že nemůžeme vytvořit částečně segmentovanou strukturu, ale umožňuje uživateli zobrazit a upravovat regiony souběžně v zobrazení ve třech rovinách a ve 3D zobrazení. K definování regionu je zapotřebí pomocí myši vytvořit uzavřenou konturu, jenž může být

upravována pomocí vrcholů této kontury. Pomocí manuálního vykreslení lze iniciovat automatickou segmentaci nebo upravovat výsledky segmentace.[8] Nástroj cut-plane umožňuje snadné rozdělení segmentace do regionů a odstranění přebytečných částí které nepatří do struktury zájmu. Manuální segmentaci můžeme provádět ve všech rovinách označováním hranic struktur a segmentovaná struktura je zobrazena jako 3D model.

1.6 FreeSurfer

FreeSurfer je open source softwarový balíček na zpracování a vizualizaci obrázků z neurologie. FreeSurfer byl ze začátku balíčkem, který se specificky zaměřoval na generování povrchové reprezentace mozkové kůry. Od té doby se však vyvinul v nástroj, který dokáže automaticky tvořit modely většiny makroskopických struktur nacházející se v mozku z T1 váženého obrazu. Je využíván jako strukturální MRI analyzační software pro Human Connectome Project, který se zabývá mapováním mozku s cílem spojit strukturu mozku s jeho funkcí a chováním. Využívá své vlastní formáty pro ukládání a manipulaci s objemovými daty. Mezi podporovanými externími formáty patří DICOM, Analaze a MINC. Je psán v programovacím jazyku C++ a je podporován v operačních systémech Linux a Mac Os.



Obr. 1.3: Uživatelské rozhraní FreeSurfer - segmentace mozku

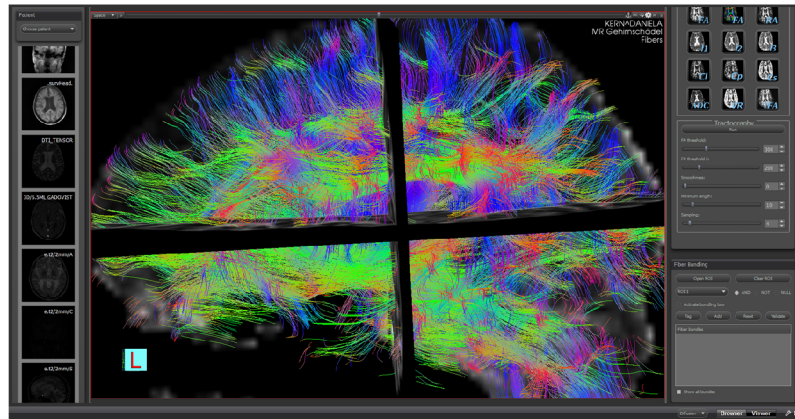
Obsahuje sadu nástrojů pro analýzu kortikálního povrchu z fMRI. V balíčku je také automatické značení 35 ne-kortikálních struktur. Pro difuzní MRI nabízí 3 hlavní nástroje.[9]

- Tracula – je nástroj pro automatickou pravděpodobnostní traktografii s využitím předem známých anatomických struktur. Rekonstruuje hlavní cesty bílé hmoty za použití již označených segmentovaných struktur, kterými jednotlivé cesty prochází.
- AnatomuCuts – je nástroj pro shlukovou analýzu založenou na anatomické podobnosti.
- DiffusionTool je pomocný nástroj generující nastavení pro TRACULA složky a nastavování parametrů pro AnatomCuts. FreeSurfer také nabízí nástroj pro segmentování struktur, povrchu mozku, včetně analýzy celého mozku pořízené metodou PET.

1.7 MedInria

MedInria je open source platforma pro vizualizaci a zpracování medicínských obrazových dat. Dokáže zpracovávat multi-dimenzionální snímky, včetně 4D. Podporuje formáty DICOM a integruje PACS, které umožňují zobrazení, správu a ukládání obrazových dokumentů. Umožňuje zobrazení od tenzorů difúze - DTI (ang. diffusion tensor imaging) až po fraktografii, včetně zobrazení tenzorových polí. Pracuje s formáty DICOM a podporu pro připojení k Shanoir serveru (webová platforma pro archivaci medicínských obrazových dat) a balíček pro nahrávání DTI dat. MedInria nabízí těchto 5 základních funkcí.[10]

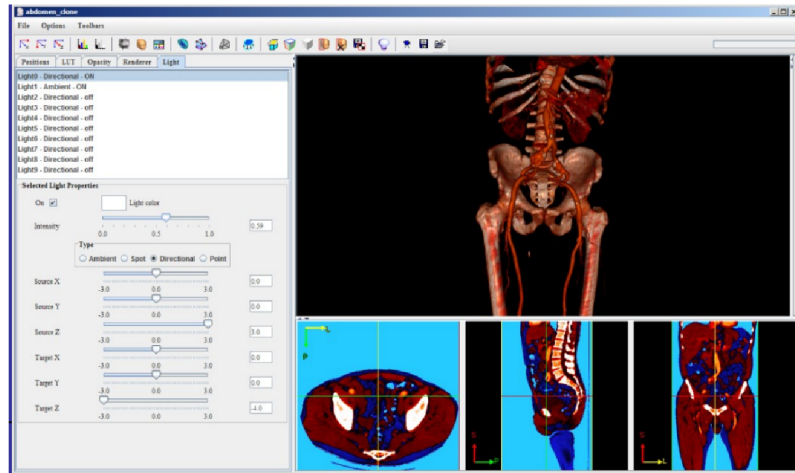
- Vizualizace: dle nastavení zobrazuje ovládací prvky pro 2D, 3D nebo 4D data. Zobrazení je možné pro jeden pohled. Další data je možné přidávat, ale budou se na sebe vrstvit a v daném okamžiku je možné vidět pouze jednu sadu dat. Nastavit se dá i Multi mód kdy data budou rovnoměrně rozprostřena a zobrazena vedle sebe.
- Difúze: Tato část je nakonfigurována pro zpracování a vizualizaci difuzní MRI. Součástí jsou nástroje pro úpravu tenzorů a zobrazení nervových drah. V této části jsou také algoritmy pro MR traktografii jako například algoritmus pro automatickou extrakci mozku, kdy je ze snímků odstraněna lebka.
- Registrace obrazů: obsahuje algoritmy pro registraci obrazu. Algoritmy transformují pohybující se obraz, aby co nejlépe seděl na referenční nebo fixní obrazu.
- Segmentace: MedInria nabízí v základu pouze Painting Toolbox, který obsahuje štětec pro manuální segmentaci. automatická, ani semi-automatická metoda prozatím není dostupná.
- Filtrace: pro filtraci lze používat základní filtry obsažené v ITK, ale i pokročilejší algoritmy pro potlačení šumu.



Obr. 1.4: Vizualizace nervových drah - traktografie

1.8 MIPAV

MIPAV je open source aplikace pro zpracování, analýzu a vizualizaci medicínských obrazů. Jedná se o multimodální nástroj pro zobrazování obrazů z PET, MRI, CT a mikroskopie. Vzhledem k rozsáhlému využití napříč odvětvími je MIPAV podporován v operačních systémech Windows, Linux a Mac Os. Standartní uživatelské rozhraní nabízí analytické nástroje a sdílení studií a projektů na webové stránky. Součástí je rozhraní pro vývojáře, které může být použito pro rozšíření dalších analyzačních, registračních a vizuálních pomůcek pro zobrazování medicínských dat. Programovacím jazykem je Java. Mezi podporované formáty patří DICOM, RAW, Analyze, QuickTime, Tiff, MINC a další. Dokáže pracovat s několika dimenzionálními daty včetně 4D dat. Pro prohlížení obrazů využívá grafické rozhraní, které kromě zobrazování dovoluje manipulaci s obrazem a přidaných struktur. MIPAV je vybaven více než 20 nástroji pro zpracování a analýzu obrazu. Pro segmentaci obrazu je v nabídce manuální, semi-automatická i automatická segmentace. Automatická analýza však často potřebuje manuální opravu hranic. Umožňuje také vytvořit a používat vlastní segmentační algoritmy. Součástí je výpočet základních statistik segmentu jako je objem/plocha, počet voxelů/pixelů, orientace, hustota nebo hmotný střed.[11] MIPAV je například využíván pro segmentaci a vizualizaci strukturální a funkční MRI obrázků mozku. Zde se využívá optimalizovaná segmentace metodou watershed. Dalším využitím je ve zpracování obrazů z oční angiografie, kde je nutné lokalizace cév a dále superponovat obrazy pořízené různými metodami.



Obr. 1.5: MIPAV - uživatelské rozhraní, 3D rendering

1.9 Dragonfly

Dragonfly je platforma pro zobrazování a vizualizaci multimodálních obrazových dat. Nejedná se o open source systém. Je zdarma pouze pro výzkumné účely, pro komerční použití je nutné zaplatit předplatné. Dragonfly pro operační systémy Windows, Linux, MacOS, Chrome OS. Dokáže pracovat s 2D, 3D i 4D daty. Je podporováno mnoho formátů, včetně DICOM, Analyze, RAW, Tiff a mnoho dalších. Lze využít velkou řadu nástrojů pro zpracování obrazů jako rekonstrukce obrazu (např. vějířové CT), filtrace nebo segmentace. Nechybí synchronizované 2D a 3D zobrazení pro okamžité porovnání různých datasetů. Dragonfly umožňuje přístup k rozhraní pro psaní makra v Pythnu, která dokážou automatizovat rutinní kroky.[12] Součástí jsou i běžné nástroje pro filtrace jako vyhlazování, ostření, extrakce znaků nebo velké množství tresholding metod. Filtry se mohou ručně nastavit na různé velikosti 2D a 3D kernel filtrů. Velkou výhodou může být široké využití umělé inteligence využívající hlubokého učení. V nabídce neurálních sítí je například UNet, DenseNet nebo FusionNet. Využit lze již existující přednastavení pro učení nebo vytvořit síť podle vlastních parametrů pro správnou segmentaci. Pro segmentaci můžeme využít nástrojů z metod aktivních kontur, interaktivního tresholdingu nebo automatické objektové segmentace. Paint tools obsahují 2D/3D Brush, Point-and-Click, Grow, Smart Grid, and Snap. Mezi automatické segmentační metody nabízí například metodu watershed. Automatické segmentace lze vylepšit editací pomocí nástroje štětce (brush tools).

2 3DSlicer

V této kapitole se seznámíme s 3Dsliceru, jeho rozhraním a programovým vybavením. Slicer je open source software pro vizualizaci, zpracování, segmentaci, registraci a analýzu medicínských a jiných 3D obrazů. Programovými jazyky jsou C++ a Python a využívá platformu QT. Podporované operační systémy:

- Windows: Win 10 nebo Win 11, verze Win 7 a Win 8 již nejsou testovány, ale mohou stále fungovat.
- MacOS: MacOS High Sierra (10.13) nebo starší verze
- Linux: Ubuntu verze 18.04 nebo starší

Všechny podporované formáty pro obrazy a modely nalezneme v *Tab. 2.1* a *Tab. 2.2*.

Tab. 2.1: Formáty obrazových dat

Formát	koncovka souboru	Formát	koncovka souboru
DICOM	.dcm	Brains2	.mask
NRRD	.nrrd, .nhdr	GIPL	.gipl .gipl.gz
MetaImage	.mhd, .mha	JPEG	.jpg, .jpeg
VTK	.vtk	LSM	.lsm
Analyze	.hdr, .img, .img.gz	PNG	.png
NifTI	.nia, .nii, .nii.gz	Stimulate	.spr
BMP	.bmp	TIFF	.tif, .tiff
BioRad	.pic	MGH-NMR	.mgz

Tab. 2.2: Formáty modelů

Formát	koncovka souboru
VTK Polygonal Data	.vtk
VTK XML Polygonal Data	.vtp
STL	.stl
OBJ	.obj
Ostatní (právě testované)	.g, .byu, .orig, .inflated, .sphere, .white, .smoothwm, .pia

2.1 Moduly

Slicere má v základní nabídce více jak 100 modulů. Každý modul obsahuje specifickou sadu funkcí, které vytvářejí nebo manipulují s daty ve scéně. V této části je stručně popsáno základních 11 modelů, které jsou důležité pro práci s medicínskými obrazy.[14]

Data

Data modul ukazuje všechny data sety nahrané ve scéně a umožňuje měnit základní vlastnosti a provádět základní operace na všech datech bez nutnosti přepínat k jiným modulům. Jeho hlavními funkcemi jsou:

- vyhledávání, přejmenování, odstranění, přesunutí objektů v MRML stromě.
- Organizace dat ve složkách
- export DICOM dat
- řídit transformační řetězce
- schovávání a ukazování větví zobrazitelných dat
- zobrazení všech uzlů (pro vývojáře)

DICOM

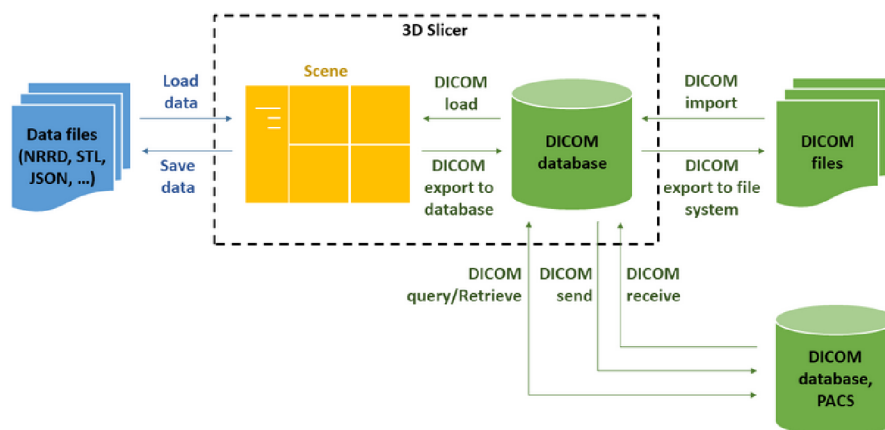
Tento modul umožňuje importovat a exportovat DICOM data a také převádět DICOM data přes síť. Slicer podporuje většinu běžných používaných funkcí, jako je čtení/psaní data setů z disku a do disku v DICOM formátu nebo přenášení data setu přes síť za použití DIMSE a DICOMweb protokolů. Standardní DICOM rozšíření jsou dostupná pro segmentované objekty, RT dat a PET-CT dat. Hlavní funkce DICOM modulu jsou:

- vytváření DICOM databází
- čtení DICOM souborů – probíhá ve 2 krocích, nejdříve importuje data a poté je načte a zobrazí do scény
- DICOMweb networking – posílání DICOM položek na vzdálený server
- Export dat ze scény do DICOM databáze a DICOM souborů
- mazání dat z databáze
- zobrazení DICOM metadat

Markups

Tento modul vytváří a edituje značení jako body, linie, úhly, křivky, ROI a další. Používané funkce můžeme rozdělit na několik částí:

- Display – obsahuje nástroje pro zobrazení značení. Zviditelnění nebo skrytí značení, průhlednost, velikost hraničních bodů, velikost písma. Je zde přístup i k pokročilejším nastavení jako typ a barva hraničních bodů a linií, tloušťka linií nebo font písma a mnoho dalších nastavení.
- Control points



Obr. 2.1: Diagram práce s DICOM daty ve Sliceru[15]

- Measurements – 3D slicer nabízí různé nástroje pro měření značených částí a objektů: délka křivky nebo linie, měření úhlů, průměr zakřivení a maximum zakřivení, plocha označených rovin a obsah pro ROI.
- Export/Import Table
- Curve setting – obsahuje funkce pro nastavení a úpravu křivek. Typy křivek: lineární (kontrolní body jsou spojeny rovnou linií), spline (vyhlazená interpolovaná křivka), Polynomiální křivky, nejkratší vzdálenost na povrchu (tato křivka je vytvořena na bodech povrchového modelu, vytváří křivku přes model a spojuje 2 body nejkratší možnou cestou)
- Resample – obsahuje nástroje pro záměnu kontrolních bodů za jiné kontrolní body. Output node nahradí kontrolní body za body křivky s kontrolními body, které jsou od sebe rovnoměrně vzdálené. Vytvoření nových bodů za použití modelu (kontrolní body budou promítnuty na povrch zvoleného modelu)
- Plane settings – nastavuje se zde metoda vytváření roviny pomocí kontrolních bodů a také jejich velikost a hranice. Nastavení pomocí 3 bodů, kdy první bodem je počátek, druhý bod značí X osu a normálový vektor je produktem vektorů z 2 a 3 bodu do počátku. Nastavení pomocí jednoho bodu, kdy normálový vektor je paralelní ke směru aktuálního pohledu na scéně. Poslední možností je pomocí fitování, kdy je rovina definována alespoň třemi body.
- ROI settings – Oblast zájmu může být definována jedním kontrolním bodem nebo pomocí mnoha kontrolních bodů vytvářející hranice oblasti. V případě definování oblasti pomocí jednoho bodu je hranice definována osovými směry a vzdálenostmi od kontrolního bodu, jenž tvoří počátek. ROI lze editovat pomocí interaktivního módu, kdy můžeme měnit orientaci a pozici oblasti přímo ve scéně.

Model

Tento modul je používán pro načítání a seřizování zobrazených parametrů načtených nebo vygenerovaných 3D povrchových modelů. Můžeme měnit barvu a průhlednost modelu a zobrazovat informace o daném modelu jako jeho plocha, obsah, počet bodů nebo počet buněk na povrchu modelu.

Scene Views

Obsahuje nástroje pro organizaci několika „živých“ zobrazení dat na scéně. Lze vytvořit libovolné množství zobrazení a upravovat jejich parametry jako průhlednost dat, 3D view, rozložení a další. V nabídce je i nastavení již předefinovaných startovacích bodů pro bližší práci s částmi dat. Mezi hlavní funkce patří vytvoření, editace a obnovení scény.

Segmentations

Tento modul řídí segmentace. Každá segmentace obsahuje několik segmentů, které odpovídají jedné struktuře nebo oblasti zájmu. Každý segment může být tvořen několika různými daty reprezentující stejnou strukturu. Modul podporuje automatickou konverzi mezi těmito daty, pokročilé nastavení zobrazení a import/export příznaků. Mezi základní reprezentace patří planární kontura, binární příznaková mapa, uzavřený model povrchu.

Tento modul umožňuje vizualizaci struktur ve 2D a 3D, Definuje region jako vstup pro analýzy, vytváří povrchovou síť z obrazu pro 3D tisk a editaci uzavřeného 3D povrchu.

Segment Editor

Segment editor specifikuje oblasti zájmu ve 2D,3D a 4D obrazech. Editor funguje jako photoshop pro 3D struktury. Pomocí editoru můžeme snadno upravovat překrývající se segmenty, zobrazovat jak ve 2D tak i ve 3D pohledu, segmentovat pomocí interpolace a extrapolace řezů nebo editovat segmentaci na řezech v jakékoli orientaci.

Transforms

Slouží k vytváření, vizualizaci a editaci prostorových transformací. Jednotlivé transformace jsou uloženy v transformačním uzlu, kde jsou popsány jejich pozice, orientace a deformace ve světovém souřadnicovém systému. Mezi podporované transformace patří lineární, b-spline transformace, Thin-plate spline interpolace, grid transformace

View Controllers

Tento modul dává přístup k nástrojům ovládající několik zobrazení v jednom panelu. Slice view a 3D view jsou 2 typy zobrazení, ve kterých dostaneme ovládání v tomto modulu. Každý typ má svůj vlastní panel s ovládacími prvky. Při práci s 3D obrazy je možné vyvolat panel navíc pro ovládací prvky určené pro jednotlivé řezy.

Volume rendering

Vizualizační technika pro vytváření 3D objektů z objemových dat bez potřeby použití segmentace. Toho je docíleno přiřazováním barvy a průhlednosti jednotlivým voxelům, na základě jejich jasové intenzity. V nabídce je několik přednastavení pro vyobrazení kostí, měkkých tkání, tuku a dalších struktur na CT a MRI snímcích. V případě potřeby můžeme manuálně měnit prahování pro jednotlivá přednastavení.

Volume

Poskytuje informace o Volume node. Dovoluje upravovat nastavení vyobrazení dat a měnit jednotlivé objemové typy. Objemový uzel uchovává voxely v přímočaré síti. Osy sítě jsou na sebe kolmé a mohou být libovolně umístěny v prostoru. Velikost Voxelů se může v osách lišit.

2.2 Extensions

Nové prvky mohou být přidány do 3D Sliceru instalací rozšíření v Extension Manageru. Rozšíření je balíček, který obsahuje jeden nebo více Slicer modulů. Po instalaci rozšíření jsou dané balíčky k dispozici stejně jako přítomné základní moduly. Pro rychlé a snadné vyhledávání a stahování rozšíření jsou v provozu webové stránky nazývající se Slicer Extensions Catalog. Tyto stránky jsou přímo dostupné z aplikace a dovolují rozšíření snadno instalovat nebo odinstalovat. Přidání rozšíření bez přístupu k internetu je možné udělat manuálně v případě kdy již máme rozšíření stažené z Webových stránek.[15]

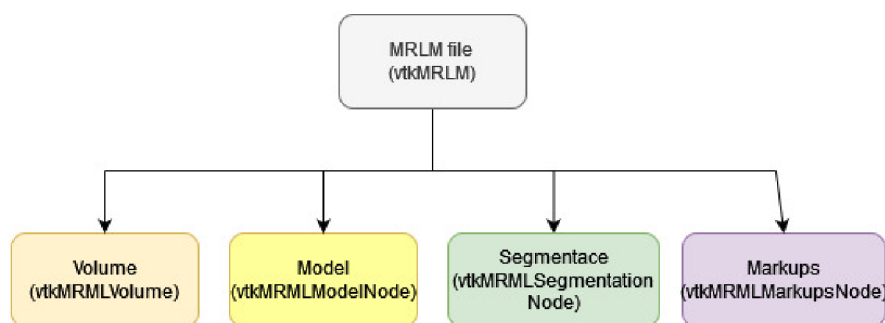
Slicer je vybudován především na C++ a je zpřístupněn v Pythnu pomocí balíčků. Konkrétně se jedná o balíčky mrml, slicer, vtkTeem, vtkAddon a vtkITK. C++ třídy jsou v pythnu zpřístupněny pomocí PythonQt a VTKPython wrapperů. Jména Qt tříd začínají písmeny q nebo Q, tyto třídy jsou odvozeny z Q objektů a řídí se Qt konvencemi. Jména VTK tříd začínají písmeny vtk, jsou odvozeny z VTKObjektů a řídí se jejími konvencemi.

Při vytvoření nové rozšíření je potřeba udělat několik kroků pro jeho zveřejnění. Pro testování rozšíření ve Sliceru je nutné přidat cestu ke všem složkám obsahující kód vytvořeného modulu v .py formátu do Sliceru. V případě, kdy je rozšíření zveřejněno pomocí Extension wizard je zdrojový kód rozšíření pravidelně kontrolován a automaticky se vytvářejí balíčky pro Linux, MacOS a Windows. Status rozšíření je zveřejněn na webových stránkách, kde lze sledovat, jestli je rozšíření kompatibilní. V případě, kdy rozšíření selže v některých testech je možné pomocí odkazu si prohlédnout seznam chyb. Při zveřejnění je nutné napsat alespoň readme textový soubor, který popisuje základní funkce a informace jak rozšíření používat. Před předložením

rozšíření do Slicer Extensions Index je nutné splnit všechny podmínky, které by mělo rozšíření obsahovat.

2.2.1 Medical Reality Modeling Language

3D slicer využívá *Medical Reality Modeling Language* (MRML), jedná se o datový model vytvořený pro reprezentaci data setů, které jsou přítomny v aplikacích pracujících s medicínskými daty. MRML software knihovna je založena na *The Visualization Toolkit* (VTK) a používá *Insight Toolkit* (ITK) pro čtení a psaní některých formátů. Knihovna je zcela nezávislá a může být použita i v jiných medicínských aplikacích. Když je MLMR soubor uložen vytvoří se textový dokument, který obsahuje indexy všech data setů. Každý MLMR soubor obsahuje odkazy na 3D dataset a MRLM nodes (vlastnosti scény), které jsou použity pro vytvoření 3D zobrazení. Typickou strukturu MLMR souboru můžeme vidět na *obrázku 2.2*. Každý uzel má vlastní jméno, příznaky a Id ve scéně. Nodes (uzly) na sebe mohou i odkazovat. [16]



Obr. 2.2: ukázka možné struktury MRLM souboru

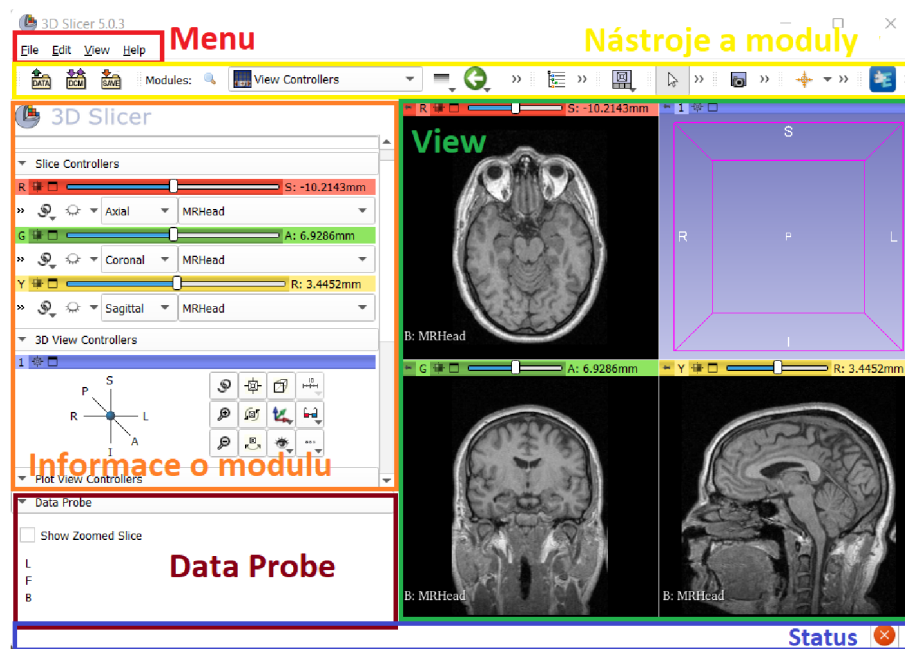
Základními Uzly jsou Volume, Model, Segmentation, Markups a Transformation. Uzly mohou vyvolat akci při změně obsahu nebo když se změní jejich vnitřní stav. Nejčastěji je vyvolána akce “Modified”, která je přivolána pokaždé když dojde ke změně obsahu.[15]

- Volume - vtkMRMLVolume a jeho podtřídy: ukládá 3D obrázky. Může se jednat o CT, MRI nebo jiné medicínské scény. Každý voxel může být skalární pro ukládání obrázků se spojitými hodnotami, značením - labels, vektory nebo tenzory v případě difuzní MRI. 2D obrázky jsou reprezentovány jako jeden řez 3D scény. 4D hodnoty jsou uloženy v sekvenčním uzlu(vtkMRMLSequenceNode).
- Model (vtkMRMLModelNode): ukládá povrchovou síť (tvořenou polygony, body, křivkami a pod.) a objemové síť (tvořenou nestrukturovanou sítí, wedge elementy a pod.)

- Segmentation (vtkMRMLSegmentationNode): zde se ukládá komplexní data vytvořená segmentací. Může vnitřně ukládat několik reprezentací segmentace, například jako labelmap obrázek a uzavřenou povrchovou síť.
- Markups (vtkMRMLMarkupsNode a jeho podtřídy): Ukládá jednoduché geometrické objekty jako linie, úhly, křivky nebo roviny pro měření a anotace.
- Transformation (vtkMRMLTransformNode): ukládá geometrické transformace. Může obsahovat jakékoli množství lineárních i nelineárních transformací.

2.3 Uživatelské rozhraní

Grafické uživatelské rozhraní se skládá z 6 částí: Menu, Panel pro modely, Nástroje, Status, Zobrazení, datová sonda.[15]



Obr. 2.3: Uživatelské rozhraní

Menu Menu je umístěno v levém horním rohu aplikace a nabízí několik základních funkcí pro práci s daty.

- File: Slouží k nahrávání dat nebo datasetů, také umožňuje stahování datasetů z internetu. Najdeme zde nástroje pro ukládání scény a datasetů. Modul DICOM je zde pro správné získávání dat z formátu DICOM a nahrávání získaných DICOM dat.
- Edit: Obsahuje možnosti, které umožňují uživateli nastavovat a customizovat vzhled a chování Sliceru, jako je zobrazení a schovávání modulů v nástrojích, velikost písma v aplikaci nebo nastavení lokace adresáře.

- **View:** Zde se nachází funkce pro zviditelnění nebo schování přídatných oken např. konzoli pro Python nebo Extension Manager. Můžeme zde také editovat rozložení zobrazování dat.

Vizualizace dat - View V nabídce je několik možností pro zobrazení dat. Lze nastavovat počet přednastavených oken, které mohou zobrazovat řezy, 3D objekty, tabulky nebo grafy. Při normálním ukončení aplikace dojde k uložení nastaveného rozložení oken.

Nástroje Umožňuje rychlý přístup k často používaným funkcím. Tento panel si může uživatel optimalizovat dle potřeb.

Pro získání přístupu k nástrojům je zapotřebí výběr modulu. Součástí je vyhledávání modulů a jeho nástrojů a také seznam historie použitých modulů, kdy je možné rychlého přepínání modulů obsažených v tomto seznamu.

Status Tento panel zobrazuje aktuální status aplikace jako například stav probíhající operace nebo chybové hlášky. Nachází se zde chybový protokol pro zobrazení seznamu vyskytnutých chyb.

informace o Modulu Tento panel se v továrním nastavení nachází v levé části aplikace hlavního okna. Zobrazuje všechny možnosti a prvky, jenž aktivní modul nabízí uživateli.

Data Probe

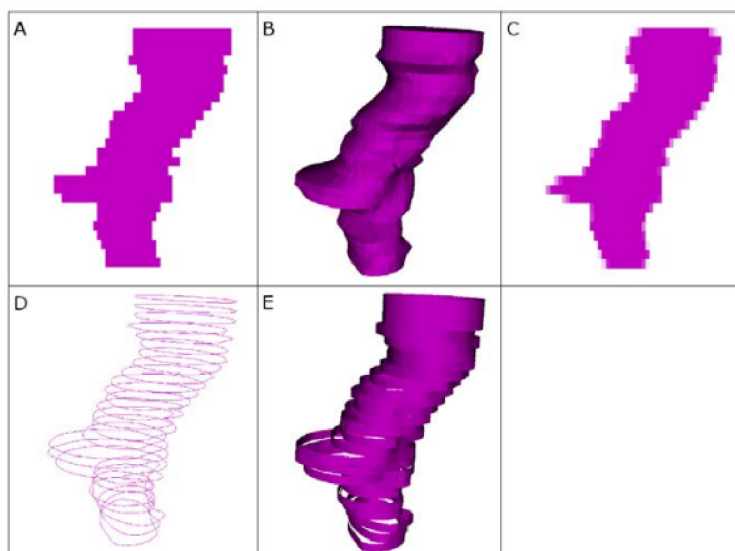
Data Probe zobrazuje aktuální informace ze zobrazených dat v místě ukazatele myši.

- **Informace o vrstvě:** Zobrazuje název, souřadnice, orientaci vrstvy a vzdálenost jednotlivých řezů od sebe. Souřadnice se skládají ze 3 hodnot, které mají zafixované značení: R/L (right/left), A/P (anterior/posterior), S/I (superior/inferior). Ukazují aktuální hodnoty od bodu počátku (0,0,0). Orientace je axiální, sagitální, koronální pro standardní anatomické orientace.
- **Informace o objemu:** Zobrazuje typ vrstvy – L (label), F (foreground), B (background). Název, nebo None, když není zvolen žádný objem. Souřadnice a hodnoty voxelu
- **Informace o segmentaci-** pro každou segmentovanou část viditelné na pozici ukazuje typ segmentace, jméno segmentovaných částí (vytvoří list v případě, když se segmentované části překrývají)

2.4 Segmentace

Slicer jako mnoho jiných softwarů pro zpracování a zobrazení medicínských obrazů využívá Insight Toolkit a The Visualization Toolkit. ITK byl vytvořen pro analýzu dat z Visible Human Project v roce 1999. Jedná se o open source knihovnu, která slouží jako platforma pro implementaci a rozvoj analyzačních a jiných algoritmů. Velká část ITK se zaměřuje na 3D segmentaci, registrační algoritmy a zpracování multimodálních dat.[17] VTK je široce využívaný open source software pro zobrazení a zpracování 3D obrazů. VTK obsahuje velké množství algoritmů využívající skalární, vektorové, texturní a objemové metody. Nabízí řadu modelovacích technik jako redukcí polygonů, vyhlazování, vystřihování nebo vytváření kontur.[18]

Segmentovaný obraz může být reprezentován různými daty. Nejčastěji je segmentace uložena jako 3D binární labelmapa, kterou využívají manuální metody pro segmentaci a jedná se o preferovaný vstup pro algoritmy zpracovávající obrazová data. Pro 3D vizualizaci struktury však tato reprezentace není příliš vhodná, a proto se volí například model povrchu struktury. Model povrchu je tvořen velkým počtem na sebe navázaných trojúhelníků. Některé segmentační algoritmy využívají pravděpodobnostní mapu, kdy je každý voxel reprezentován pravděpodobností, že patří do dané struktury. V radiologii dle DICOM standardů je zapotřebí ukládat struktury jako sady vyjádřené jako série kontur. Tyto sady je také možné ukládat jako ribbons reprezentaci, která lze snadno vypočítat z kontur. Vyjadřuje tloušťku řezu a vytváří přehlednější 3D tvary.[19]



Obr. 2.4: a) binární reprezentace, b) uzavřený povrch, c) pravděpodobnostní mapa, d) rovinné kontury, e) Ribbon model [19]

Při používání segmentace anatomických struktur a následné ukládání a zpracování segmentů musí být dodržována řada pravidel.[19]

- Správná volba konverze: uživatel by měl být vědom, že při používání některých metod je potřeba změnit reprezentaci některých dat. Při konverzi může docházet ke ztrátám nebo změnám informací, o kterých by měl být uživatel informován.
- konzistence dat: Při konverzi může být výsledná reprezentace jiná než původní. Toho může být docíleno například manuální editací reprezentace. V tomto případě se všechny ostatní reprezentace stejného segmentu stávají neplatné a scéna postrádá konzistenci. Neplatná data nesmí být přístupná.
- Koherence: sady struktur obvykle odpovídají jedné entitě (pacientovi), takže by data měla být možná ukládat, zobrazovat či zpracovávat jako jeden jednotný celek.

2.4.1 Segmentační nástroje Dostupné v 3D Sliceru

Slicer obsahuje několik nástrojů pro segmentaci obrazu. V této části jsou stručně popsány jednotlivé nástroje.

GrowCut

Jednou z využívajících metod pro segmentaci ve Sliceru je Fast GrowCut algoritmus. Fast GrowCut snadno zpracovává multidimenzionální obrazy, podporuje segmentaci různých reprezentací a okamžitě se přizpůsobuje změnám vstupních parametrů. Nejdříve je nutné manuálně zadat vstupní hodnoty. Prvním vstupem je semínko uvnitř struktury a druhým vstupem označíme pozadí obrazu. V každé iteraci jsou vyhodnoceny všechny pixely, a proto výpočetní náročnost roste s velikostí obrazu. GrowCut vyhledává nejkratší cestu za použití adaptivního Dijkstra algoritmu.

Paint

Pomocí tohoto nástroje můžeme manuálně označit segment. Průměr plochy štětce se dá libovolně nastavovat. Vytváří jednoduchou binární reprezentaci.

Erase

Pomocí toho nástroje můžeme manuálně vyřadit pixely ze segmentu a vrátit je do pozadí.

Fill between slices

Vytvoří kompletní segmentaci podle zvolených řezů. Segment je vytvořen pomocí interpolace mezi jednotlivými řezy, které mohou být od sebe libovolně vzdáleny. Tento nástroj je užitečný v případě kdy dojde k nedostatečné úvodní segmentace.

Hollow

Vstupním parametrem je segment a požadovaná tloušťka stěny. Vytvoří dutý model s uniformní tloušťkou stěny.

Smoothing

Tento nástroj vyplňuje díry ve struktuře a maže nechtěné výklenky. Mezi nabízené metody patří:

- Median: Odstraní všechny malé výklenky a zaplní malé díry. Tuto metodu lze použít pouze na zvolené segmenty (nejde aplikovat globálně na všechny segmenty).
- opening: Vymaže výklenky, které jsou větší než nastavená velikost kernel filtru. Do segmentu není nic přidáno.
- closing: vyplní ostré hrany a díry menší než nastavená velikost kernel filtru. Tento nástroj ze segmentu nic neodstraňuje.
- Gaussian: Vyhladí veškeré detaily. Segment se z tohoto důvodu často zmenší.
- Joint smoothing: vyhladí několik segmentů najednou. V případě kdy segmenty přesahují přes sebe má přednost segment, který je výše v tabulce.

Islands

Zpracovává regiony, které jsou obklopeny prázdnými voxely. Tímto například můžeme odstranit malé ostrovy kolem hlavního segmentu.

Scissors

Nástroj slouží k vystříhnutí specifického regionu. Také dokáže vyplnit a vytvořit nový segment, který je manuálně nakreslen do scény. Držení levého tlačítka myši způsobí kreslení a uvolněním myši je nástroj aplikován.

2.4.2 Metoda rozvodí

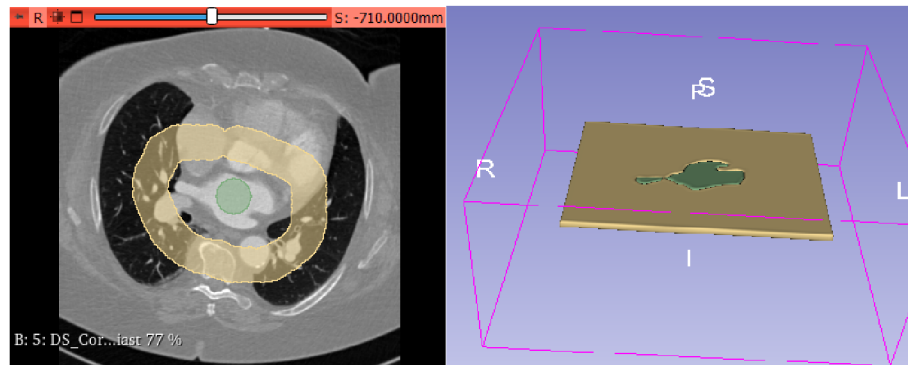
Segmentace pomocí metody rozvodí (angl. Watershed transform) je převzata z topografie. Obraz je v tomto případě vnímán jako reliéf. Princip je založen na postupném zaplavování, kdy se hladina zvedá z lokálních minim. V momentě, kdy by mělo dojít ke spojení dvou zaplavených oblastí je mezi oblasti vložena hráz. Algoritmus je zastaven, když dosáhne maximální úrovně v obraze. Vztyčené hráze reprezentují hranice mezi jednotlivými oblastmi, čímž dojde k segmentaci obrazu. Metoda je zpravidla aplikována na gradientní obraz. Při aplikování metody však často dochází k vytvoření příliš velkého množství segmentů.

Využití semínek

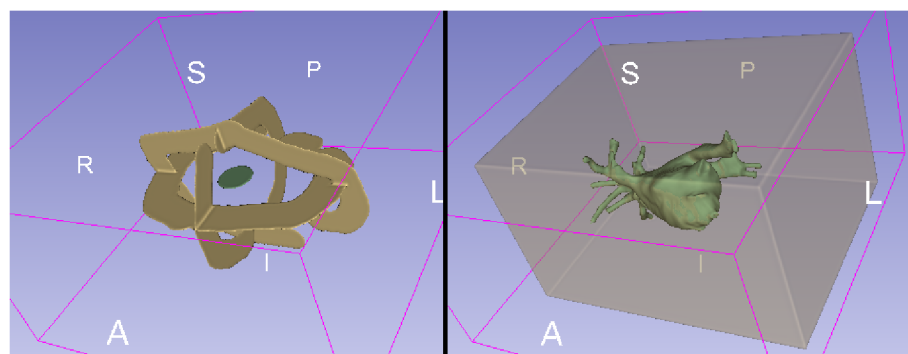
Způsob, jak se vyhnout problému přesegmentování obrazu je implementací semínek. Vstupem filtru je obraz, ve kterém jsou uživatelem vyznačeny minima (angl. Markerbased watershed transform). Zadaná minima jsou promítnuta do vstupního obrazu a každému minimu je přiřazeno jméno. Obraz je zaplavován pouze ze zadaných minim. Jména výsledných segmentů jsou odvozena z jmen označených minim.[22]

Segmentace Grow from seed a Watershed ve 3D Slicer

Slicer nabízí pro segmentaci hned několik metod jako flood filling, local thresholding, fast marching, watershed a Grow from seeds. V této části se zaměříme na poslední dvě zmíněné metody, jež jsou prakticky využitelné pro segmentaci srdce. Segmentace je vytvořená ze segmentů, které zadá uživatel. Nevýhodou těchto metod je, že berou pozice zadaných vstupů a poté je vytvořena segmentace pouze v blízkém okolí těchto segmentů. Myšlenkou tohoto způsobu je omezení výpočetní náročnosti, počet voxelů které jsou zpracovány algoritmem jsou udrženy na minimu. Názorná ukázka segmentace metodou watershed je znázorněna na obrázku *obr. 2.5*, kdy vstupem bylo označení segmentu levé síně a jeho okolí v transverzální rovině. Na obrázku *obr. 2.6* je znázorněno, že pro kompletní segmentaci je od uživatele vyžadován vstup, který bude ohraničovat segment v každé rovině.



Obr. 2.5: Vlevo: vstup v transverzální rovině, Vpravo: výsledná segmentace vytvořená 3d slicerem metodou watershed



Obr. 2.6: Vlevo: vstup ve 3d pro metodu Watershed, Vpravo: výsledná segmentace levé síně.

Segmentování různých částí často není ukončena aplikováním algoritmu Watershed nebo Grow from Seeds. Následné úprava získaných segmentů je často nutná

pro získání kvalitní segmentace. 3D slicer obsahuje funkce jako smoothing, Margin, Hollow, Islands nebo Logical operators. Všechny tyto funkce jsou dostupné v Segment editoru.

3 Vytvoření a implementace Modulu HeartSegmentation

Motivací pro vytvoření nového modulu do 3D sliceru je zautomatizování a zjednodušení segmentace srdce. K docílení automatizace byly využity nástroje dostupné ve 3D sliceru a v pythnu byl vytvořen algoritmus pro vytváření nových segmentů, a to včetně segmentů pozadí. Pro získání segmentace je využita metoda watershed. Součástí segmentace je také automatický post-processing, který aplikuje funkce pro vyhlazení segmentu, odstranění vzniklých malých segmentů a vyplnění děr v segmentu. Vstup od uživatele je udržován na minimu, kdy se po uživateli chce pouze vstup v jedné rovině. Segmenty pozadí se generují automaticky kolem vytvořených segmentů.

3.1 Vytvoření Modulu

Pro vytvoření nového skriptovaného modulu byla využita šablona, která je dostupná ve 3D Sliceru. Využitím této šablony bylo vytvořeno rozšíření (angl. Extension) pojmenované *Heart_WatershedSegmentation*. Součástí této složky je CMakeLists, png obrázek (icon) a jednotlivé Moduly. Součástí rozšíření *Heart_WatershedSegmentation* je pouze modul *HeartSegmentation*. Složka Modulu obsahuje složky `__pycache__`, Resources a Testing. Také se zde nachází CMakeLists a soubor s Python skriptem. Python soubor obsahuje již určitá přednastavení a jeho součástí je vytvořená třída *ScriptedLoadableModuleWidget*. Součástí jsou funkce, které definují chování modulu při startu modulu, jeho ukončení, při uzavření 3D Sliceru a pod. Třída *HeartSegmentationWidget* vytváří grafické rozhraní pro modul. Slouží především k přístupu ke scéně, ukládání parametrů a k ovládání a nastavení grafického rozhraní. Tato třída komunikuje s třídou *HeartSegmentationLogic*.

HeartSegmentationLogic je třída, která zadává funkcionalitu modulu. Obsahuje všechny funkce, které provádí veškeré zpracování obrazu, segmentace nebo jiné úkony pro fungování modulu. Logická třída komunikuje zpět s widget třídou, která podle operací aktualizuje zobrazení v grafickém rozhraní.

3.2 Vytvoření uživatelského rozhraní

Grafické rozhraní pro vytvořený modul bylo vytvořeno pomocí QT designer[20], jenž je přístupný ze 3D Sliceru. Při spuštění modulu skript načte grafické rozhraní ze souboru „UI/HeartSegmentation.ui“. V tomto souboru jsou uloženy všechny použité

widgets, jejich nastavení a rozložení, která byla vytvořena ve QT designer. Funkcionalita a vizualizace widgetů je dále upravována pomocí kódu. Krom obvyčejných widgetů dostupných z knihovny *Qt widget gallery*, jsou použity také přizpůsobené widgety pro Medical Reality Markup Language (MRML) architekturu jako widget `qMRMLNodeComboBox`. Použity byly také widgety dostupné z Visualization Toolkit (VTK) a the Common Toolkit (CTK).

3.2.1 Připojení a nastavení Widgetů

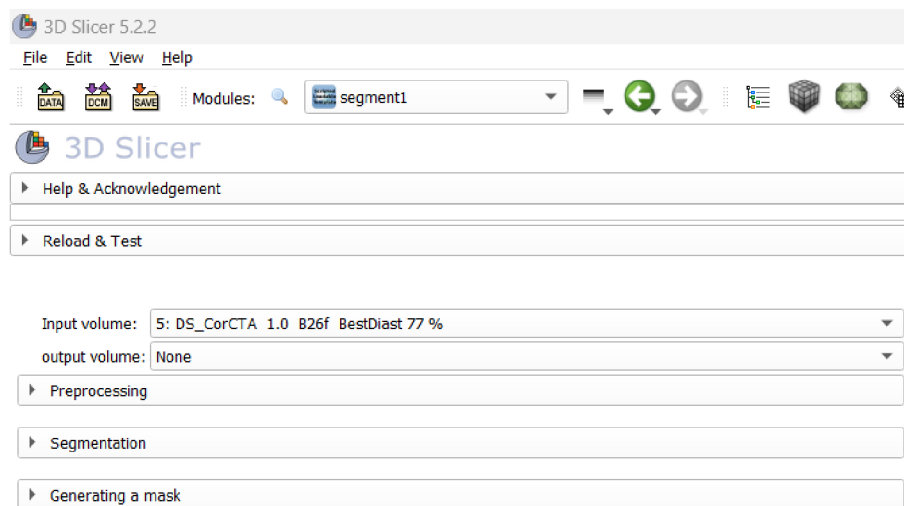
Pro widgety je nastaveno spojení. Spojení zaručuje, že kdykoli uživatel změní nastavení v grafickém rozhraní tak je změna uložena do MRML scény (uložení do příslušného parameter node). V této části jsou také nastavena spojení pro tlačítka, která mění parametry přes definované funkce. Nastavení widgetů je primárně nastaveno v QT designeru a následně načteno ze souboru `'UI/HeartSegmentation.ui'`. Widgety typu `qMRML` musí být navíc připojeny k aktuální MRML scéně. V případě pro widgety `NodeComboBox` je tak učiněno v QT designeru. Widget `SegmentEditor` je připojen ke scéně pomocí kódu. Widget `SegmentEditor` funguje jako celek a není potřeba připojovat jeho součásti (součástí je například `NodeCombobox` widget nebo tlačítko pro 3D zobrazení). Grafické rozhraní je z největší části aktualizováno funkcí `updateGUIFromParameterNode`. Tato funkce je volána pokaždé, kdy dojde ke změně v `ParameterNode`. `ParameterNode` uchovává data pro náš modul. Například parametr `"InputVolume"` uchovává ID aktuálně zvoleného objemu (3D obrazu) . Pokud je zavolána funkce, která mění tento parametr, vyvolá to i změnu v zobrazení v grafickém rozhraní. Nachází se zde také nastavení stavu některých widgetů. Například stav tlačítka `CropVolumeButton` je ovlivněno parametry: „ROI“, „InputVolume“ a „OutputVolume“. Je-li jeden z těchto parametrů `None` nebude toto tlačítko pro uživatele přístupné (tyto parametry jsou nutné pro jeho funkcionalitu).

Aktualizace parametrů z grafického rozhraní je z velké části řízeno funkcí `updateParameterNodeFromGUI`. Tato funkce je zavolána v případě, kdy uživatel změní nastavení v grafickém rozhraní. Parametry jsou v této funkci aktualizovány v jednom sledu (in batch). Některé parametry nejsou přímo ovlivnitelné grafickým rozhraním. Například parametr `activeSegments` je definován podle zakliknutí segmentů, které chce uživatel segmentovat, ale k získání parametru dochází až ve funkci `StartSegmentation`. Z tohoto důvodu změna stavu v GUI (výběr segmentů k segmentaci) nevyvolá okamžitou změnu ve scéně.

3.2.2 GUI

Vytvořené grafické rozhraní je rozděleno do 3 hlavních částí: Preprocessing, Segmentation, and Generating a Mask. Před těmito částmi se nachází widgety pro nastavení vstupního a výstupního objemu.

Pro nastavení vstupu a výstupu (a také ROI v Preprocessing) je použit widget `nodeCollapsibleButton`. Tento widget umožňuje zobrazovat možnosti načtených dat podle typu. Widget zobrazující vstup (angl. input) a výstup (angl. output) jsou typu „`vtkMRMLScalarVolumeNode`“ a tudíž zobrazují pouze názvy načtených objemových dat tohoto typu. Output `collapsibleButton` je přístupný pouze v případě, kdy je již vybrán vstupní objem (angl. Input volume). `CollapsibleButton` pro výstupní objem má možnosti editace a vytváření nových objemových dat typu „`vtkMRMLScalarVolumeNode`“. `NodeCollapsibleButton` je propojen s MRML scénou. Propojení zaručuje zobrazení dat nastaveného typu a aktualizaci při vytvoření nových dat.

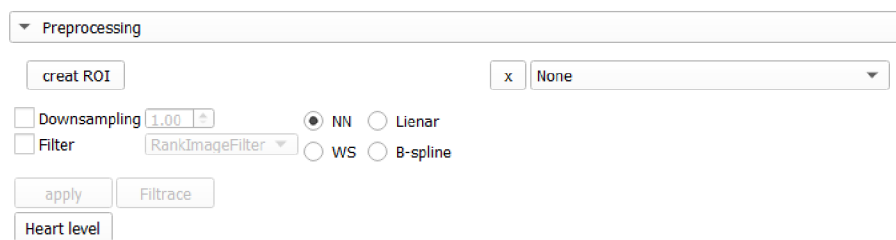


Obr. 3.1: GUI při spuštění modulu

Preprocessing

V sekci Preprocessing uživatel může vykonat řadu operací pro přípravu obrazu k segmentaci. Mezi dostupné funkce patří vytváření oblasti zájmu (ROI) ve formátu „`vtkMRMLMarkupsROINode`“. Všechna dostupná ROI tohoto typu lze vybírat ze seznamu a podle potřeby je pomocí tlačítka ve scéně zobrazovat či schovávat. ROI mohou být také reprezentovány pomocí Annotations, tyto ROI jsou uloženy v odlišném MRML uzlu a nebudou zobrazeny. Modul není nastaven pro práci s ROI vytvořených pomocí Annotation (3D Slicer se do budoucna snaží převést všechnu funkcionalitu

z AnnotationNode do MarkupsNode). Dále má uživatel možnost použít funkci pro podvzorkování vybraného objemu (angl. downsampling). Uživatel musí tuto funkci povolit pomocí checkboxu a následně nastavit míru podvzorkování. Podvzorkování je spuštěno pomocí tlačítka *Apply*. Toto tlačítko je zpřístupněno pouze pokud je také zvolená konkrétní ROI. Při podvzorkování je uživateli nabídnuta možnost zvolit typ interpolace. Mezi dostupné interpolace je Nearest neighbor (NN), Linear, Windowed Sinc(WS) a B-spline. Další funkcí v této části je filtrace obrazu. Pro zpřístupnění této operace musí uživatel povolit funkci pomocí checkboxu. Z listu je možné vybrat druh filtru, který bude použit pro filtraci objemu. Pro aplikaci filtru na objem slouží tlačítko *Filter Image*. Poslední nabízenou funkcí je nastavení Window/Level. To je umožněno pomocí tlačítka *Heart level*.

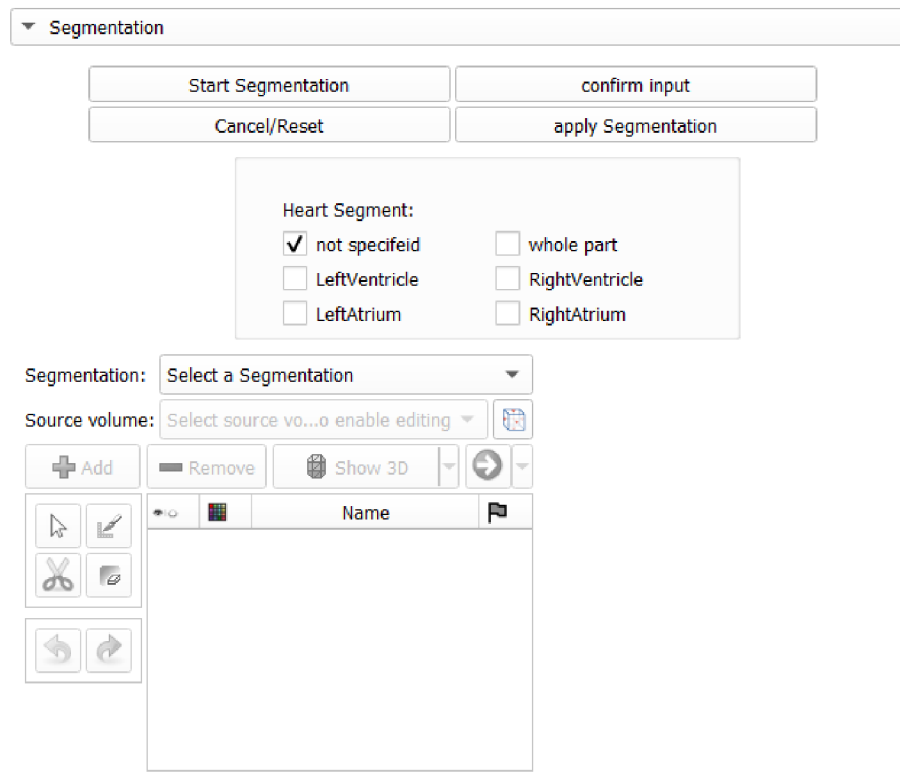


Obr. 3.2: GUI: Preprocessing

Segmentation

V sekci Segmentation je přístup k editoru segmentů a jsou zde tlačítka pro zahájení segmentace. Tlačítko *Start Segmentation* zahájí segmentaci a navede uživatele k vložení potřebných inputů. Uživatel má možnost zvolit jakou část srdce chce segmentovat. Defaultní nastavení je pro segment „Unspecified“. Mezi nabízené možnosti jsou dále „Left Ventricle“, „Right Ventricle“, „Left Atrium“ a „Right Atrium“. Lze nastavit libovolné kombinace těchto čtyř částí. Další možností je „Whole Heart“, jež zvolí segmenty všech čtyř hlavních částí srdce. V případě této volby je znemožněna další editace a pro změnu výběru segmentů musí být možnost Whole Heart odškrtnuta. Podobný případ nastane při výběru segmentu „Unspecified“, kdy při zaškrtnutí této možnosti není umožněno zvolit specifický segment. Při zahájení segmentace tlačítkem *Start Segmentation* je widget obsahující výběr segmentů schován a nelze tím pádem měnit zvolené Segmenty při vkládání vstupu. Segment editor je widget třídy „qMRMLSegmentEditorWidget“. Funkcionalitou je identický s editorem v modulu Segment Editor. Pro náš modul je přizpůsobený a vybírá užší výběr nabízených nástrojů. Ostatní nástroje jsou stále součástí editoru, jenom nejsou viditelná. Mezi viditelné nástroje patří Paint, Scissors a Erase, Margin, Islands a Smoothing. V případě, kdy by uživatel chtěl mít přístup k dalším nástrojům je nutné

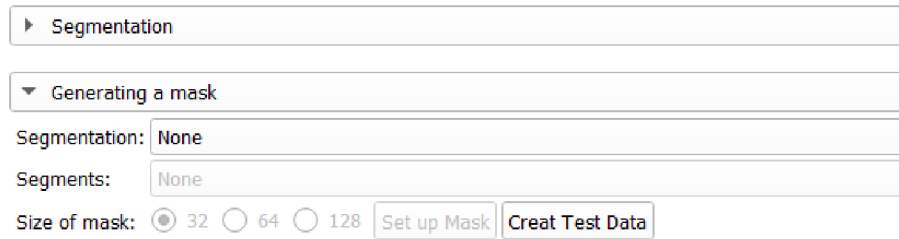
změnit list aktivních nástrojů v python skriptu (proměnná: `self.activeEffets`). Tlačítkem *apply Segmentation* se spustí funkce s watershed filtrací využívající vstupy ze segment editoru. Nastavení *Input volume* je v tomto případě ignorováno. Při vytváření segmentace a segmentů pomocí *Start Segmentation* bude však jako vstupní objem nastaven podle *Input volume* nikoli podle *Source volume* v Segment editoru.



Obr. 3.3: GUI: Segmentation

Generating Mask

V sekci Generating Mask má uživatel možnost vybrat segmentaci, která mu zpřístupní list segmentů uvnitř této segmentace. Uživatel může dále zvolit velikost masky. Mezi nabízenými možnostmi je vytvoření „`vtkMRMLMarkupsROINode`“ o velikost 32, 64 a 128. Při generaci se vytvoří ROI zvolené velikosti kolem zvoleného segmentu. ROI lze editovat ale nedoporučuje se měnit velikost ROI. Tato část je v modulu k vytváření dat pro učení neuronových sítí například typu U-Net. Pro učení je nutné mít připravená data objemů a jejich masek stejné velikosti. Data jsou uloženy do zvoleného dočasného adresáře.



Obr. 3.4: GUI: Generating a Mask

3.3 Předzpracování obrazu

Předzpracování obrazu je důležitou součástí segmentace obrazu. Modul nabízí uživateli vytváření nových oblastí zájmu, zobrazení ROI, filtraci obrazu a podvzorkování obrazu. Grafické rozhraní této části je popsáno v sekci GUI. Na obrázku *Obr. 3.2* je možné vidět rozložení jednotlivých prvků, které budou v této části popisovány.

3.3.1 Filtrace

Šum negativně ovlivňuje výsledky segmentace, proto je vhodné provést vhodnou filtraci před zahájením segmentace. Nabízenými filtry jsou *RankImageFilter*, *MedianImageFilter*. Oba filtry jsou součástí Simple ITK filtrů. Implementace filtrování je provedena využitím logické funkce z modulu *Simple filters*

RankImageFilter je nelineární filtr, kde každý výstupní pixel je definován hodnotou vstupních pixelů v uživatelem zadaném okolí. Filtry jenž používají váhování okolních pixelů dosahují lepších výsledků a lépe zachovávají hraniční pixely. Radius filtru je nastaven na hodnotu (1,1,1) a hodnota filtru je nastavena na 0.5 (median).[21]

MedianImageFilter je nelineární filtr. Aplikuje mediánový filtr na obraz. Každý pixel získává hodnotu podle medianu hodnot v jeho okolí. Díky využití medianu není výsledný obraz ovlivněn odlehlými hodnotami. Radius filtru je nastaven na hodnotu (1,1,1).

3.3.2 Oříznutí a podvzorkování obrazu

Vytvoření ROI K oříznutí nebo podvzorkování obrazu je nutné mít vytvořenou oblast zájmu (dále jako ROI - angl. region of interest). Tlačítkem *creat ROI* se zavolá funkce *CreatROI*. Funkcí je vytvořen nový objekt typu "vtkMRMLMarkupsROI-Node". Dále je vytvořen objekt typu "vtkMRMLCropVolumeParametersNode". Do parametersNode vložíme informace ohledně ID námi vytvořené ROI a ID zvoleného obrazu. Tyto informace jsou nutné, jelikož jsou využívány logické funkce modulu

Crop Volume, které vyžadují jako vstup tento konkrétní *ParametersNode*. K získání ROI je využito dvou funkcí a to *SnapROIToVoxelGrid*, který zrotuje ROI podle směru os našeho obrazu. Druhou funkcí je *FitROIToInputVolume*, která upraví velikost ROI podle velikosti vstupního obrazu. ROI je možné editovat v jednotlivých řezech a měnit její polohu či velikost. Vytvořená ROI je dostupná z widgetu *NodeComboBox* na pravé straně GUI. Nově vytvořená ROI se automaticky nezvolí a musí být manuálně zvolena. *NodeComboBox* má dostupnou možnost odstranění zvolených ROI. Widget zobrazuje pouze ROI typu "vtkMRMLMarkupsROINode". Nachází se zde také tlačítko s ikonkou oka, které nastavuje viditelnost zvolené ROI.

Podvzorkování obrazu.

Podvzorkování obrazu se odemkne pokud uživatel zaklikne *checkBox Downsampling*. Pomocí widgetu typu *spinBox* může uživatel nastavit míru podvzorkování. Hodnota 2 zmenší velikost obrazu na polovinu. Volba typu interpolace je možná pouze v případě, kdy je tlačítko *Downsapling* zaškrtnuto. Oříznutí obrazu a podvzorkování je aplikováno stejným tlačítkem. Je to dáno tím, že obě tyto operace využívají stejnou funkci *CropInterpolated* z modulu *Crop volume*. Tyto operace můžou být provedeny zvlášť nebo dohromady. Tato funkce má dohromady 7 vstupů. Vstup „fillValue“ je v obou případech nastaven na hodnotu 0 a vstup „isotropicResampling“ je v obou případech nastaven na hodnotu "true". Vstup který se mění při podvzorkování je pouze „spacingscale“, který určuje míru podvzorkování . Při použití pouze oříznutí obrazu je vstup „spacingscale“ nastaven na hodnotu 1. Vstup „interpolationMode“ je defaultně nastaven na metodu Nearest neighbor. Zbylé vstupy jsou vstupní a výstupní obraz a ROI.

3.3.3 Window/level

CT snímky běžně obsahují tisíce odstínů šedi. Vzhledem k rozlišovacím schopnostem oka a limitům zobrazení monitorů, které většinou zobrazují do 256 odstínů, jsou tyto hodnoty příliš vysoké a redundantní pro zobrazení 3D obrazu. Pomocí funkce *window/level* můžeme změnit kontrast a světlost obrazu. *Window/level* lze kdykoli manuálně upravit, ale *Slicer* již nyní nabízí řadu přednastavení (například pro zobrazení kostí nebo mozkové tkáně). Správné zobrazení srdce a jeho částí je důležité pro přehlednější zadávání inputu jednotlivých segmentů. Pro lepší zobrazení struktury srdce se v GUI nachází tlačítko, které pracuje v režimu *toggle*. Struktura srdce nabývá hodnot kolem 350 HU, ale tyto hodnoty můžou být ovlivněny kontrastní látkou. V případě zmáčknutí tlačítka se nastaví hodnoty *window/level* na 900/350. Opětvným zmáčknutím tlačítka se *window/level* nastaví do režimu *auto*. V tomto případě je *window/level* vypočítán a nastaven *Slicerem*.

3.4 Segmentace

Start Segmentation

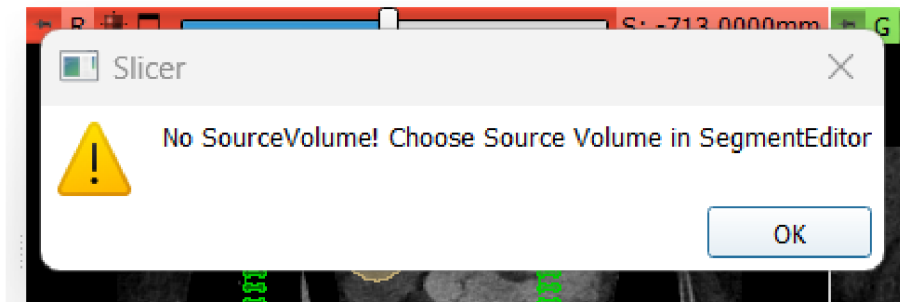
Tlačítkem *Start segmentation* je volána funkce ve widget třídě jménem *startSegmentation*. Při zahájení funkce dojde ke změně rozložení Grafického rozhraní pro přehlednější práci při vkládání vstupu. Pomocí funkce *activeSegments* je zjištěn počet a druh segmentů, které byly vybrány pro segmentaci. Dále je volána funkce z logické třídy jménem *initiateSegmentEditor*. Tato funkce zajistí správné nastavení segment editoru pro vkládání nových segmentů. Generuje se zde nový „vtkMRML-SegmentationNode“ pojmenovaný *preHeartSegmentation* a také obsahuje konstruktor pro vytvoření segment editoru v logické části v případě, kdy ještě nebyl vytvořen. Funkce *activeSegmentChangeStatus* detekuje segment, který se generuje a vyřadí ho z aktivních. Tím je zajišťováno, že bude segment jednoho typu vygenerován pouze jednou. Aktivním segmentem je myšlen segment, který uživatel vybral pro segmentaci. Další logickou funkcí v této části je *InputForSegments*, která vygeneruje první segment a dále se čeká na vložení vstupu od uživatele. Pro uživatele je automaticky aktivován nástroj Paint. Vstup stačí zadat pouze v jednom řezu, kdy nezáleží jaký řez si uživatel vybere. Segmenty jsou generovány v tomto pořadí: „Unspecified“, „LeftVentricle“, „RightVentricle“, „LeftAtrium“, „RightAtrium“. Po vložení vstupu pomocí nástroj Paint uživatel potvrdí vstup tlačítkem *Confirm input*.

Confirm input

Tlačítkem *Confirm input* je volána funkce *setBackground* a vykonává 2 hlavní operace. Generování nových segmentů a generování segmentu pozadí. Toto tlačítko přímo neukládá vstup od uživatele, jak by mohl název napovídat. V okamžiku, kdy uživatel použije nástroj Paint je tato změna uložena do segmentu. Zmáčknutím tlačítka se pouze vygeneruje nový segment v případě, kdy ještě nebyly vytvořeny všechny segmenty určené uživatelem a změní aktivní segment na nově vygenerovaný, čímž ukončí editaci předešlého segmentu. Nový segment se vygeneruje opět funkcí *InputForSegments*. V případě, kdy byly vytvořeny všechny segmenty a vloženy všechny vstupy pomocí Paint nástroje zmáčknutím tlačítka se automaticky vygeneruje segment pozadí.

Oproti funkci *startSegmentation*, kde uživatel nemůže zasahovat do vstupů volaných funkcí je funkce *setBackground* vybaveno mnoha výjimkami a hlídáním vstupu. V případě, kdy uživatel nemá vybráno *Segmentation* nebo *Source volume* v segment editoru bude uživatel vyzván, aby vložil tyto vstupy. Toto tlačítko je dostupné i bez použití tlačítka *Start Segmentation* a uživatel tedy může generovat segment pozadí na vlastní segmentaci, které vytvořil nebo získal jakýmkoli způsobem. Segment „Background“ není vytvořen, pokud je jakýkoli segment v segmentaci prázdný. Na

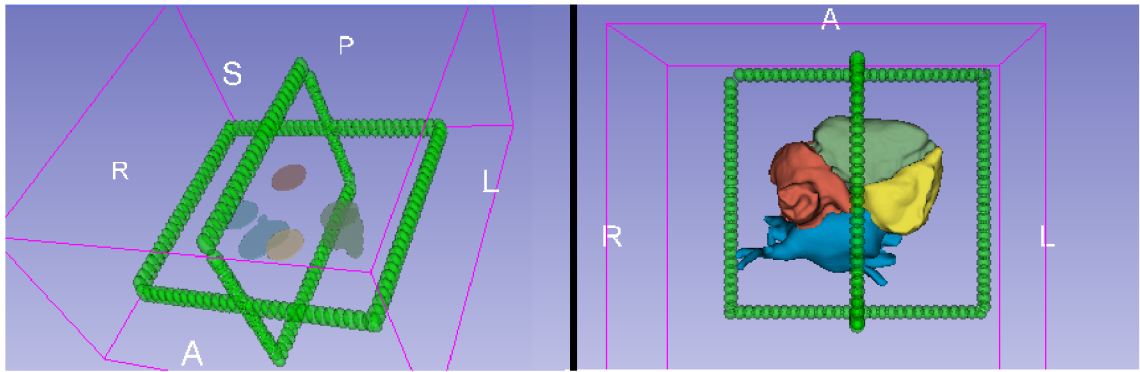
chybějící vstupy je uživatel upozorněn vyskakovacím oknem. Ukázka tohoto okna je znázorněna na obrázku *Obr. 3.5*.



Obr. 3.5: Error Window: Chybí input pro Source Volume

Segment pozadí

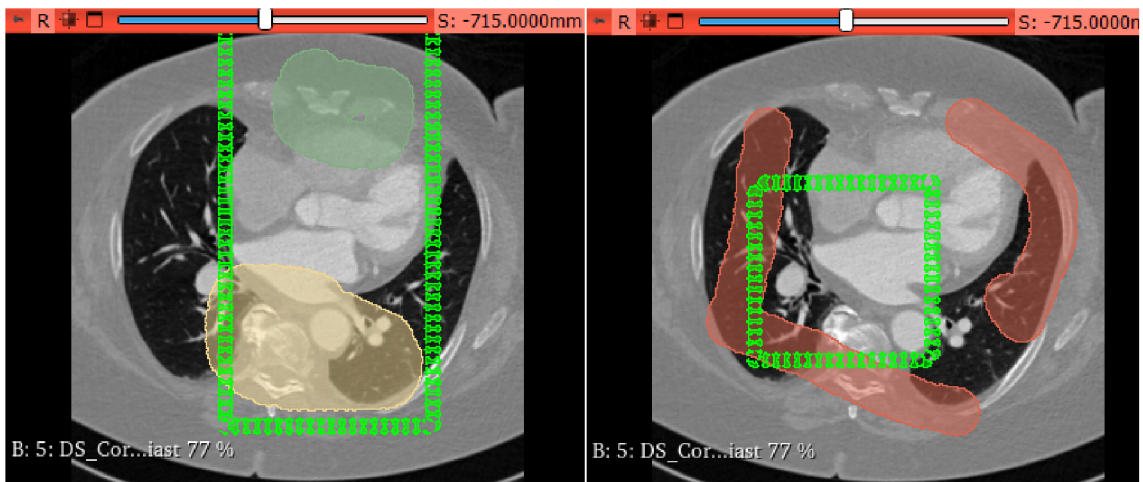
Při vygenerování posledního segmentu a potvrzení jeho vstupu aktivuje tlačítko *Confirm input* funkce pro generování segmentu pozadí. Pro vytvoření segmentu pozadí je potřeba znát středy segmentů(vstupů), které uživatel zadal. Pro získání středů je použita funkce *GetCenter*. Tato funkce nepřevádí pouze středy segmentů, ale také transformuje souřadnice. *SegmentationNode* má funkci *GetSegmentCenterRAS* pomocí které získáváme středy jednotlivých segmentů. V našem provedení však využíváme souřadnice typu IJK. RAS souřadnice jsou ovlivněny nastavením při snímání obrazu. Vytvořené algoritmy využívají souřadnice typu IJK, které udávají souřadnice jednotlivých voxelů v obraze. Další funkcí je *getMinMaxCenter*. Funkce vykoná operace pro zjištění maximální a minimální hodnoty všech středů v každé ose. Tato funkce není potřeba pokud je vstup pouze jeden segment. Segment pozadí se tvoří ve funkci *GetBackgroundBox*. Funkce využívá maximálních hodnot pro určení hranic výsledného segmentu, který bude vytvořen ze 2 čtyřúhelníků. Jeden čtyřúhelník je vytvořen v transverzální rovině a druhý v rovině sagitální. Tyto čtyřúhelníky nejsou identické. Je to dáno především kompozicí obrazu. CT snímky obsahují zpravidla méně řezů v Z rovině (transverzální). Výsledek se však může přibližovat tvaru krychle. Vytváření pozadí je optimalizováno na velikost srdce. Je předpokládáno, že jedna část srdce nebude větší než 10 centimetrů. Algoritmus využívá informace o rozestupech jednotlivých řezů (angl. spacing). Po získání středu segmentu přidá přibližně 6 cm do každého směru(+ a - od středu v Transverzální a v Sagitální rovině). Segment je poté vytvořen pomocí *vtkSphereSource*, jenž vytvoří polyData reprezentující segment pozadí. Pozice Segmentu je poté převedena zpět do RAS souřadnic a segment je přidán do Segmentace.



Obr. 3.6: BackgroundBox kolem vstupů a kolem vysegmentovaného srdce

Nevhodné zadávání vstupů

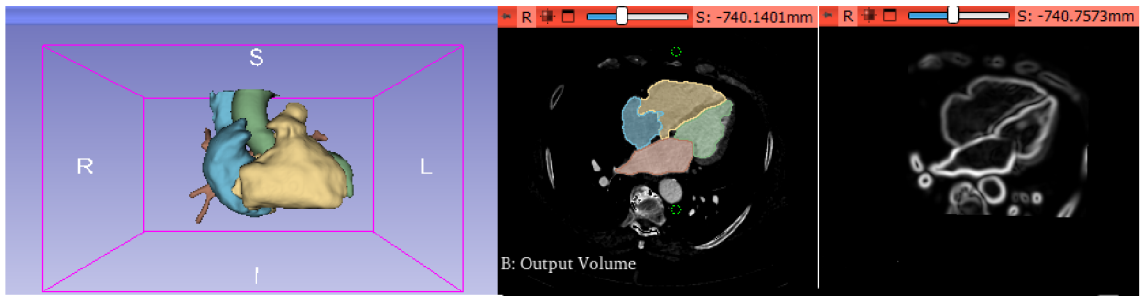
Algoritmus vytvářející segment pozadí spoléhá na skutečnost, že segmenty budou blízko sebe a jejich střed zhruba odpovídá středu srdce. Uživatel může tuto funkci využít i pro získání segmentu pozadí i jiných anatomických struktur, ale pro správnou generaci pozadí by tyto struktury neměly být silně asymetrické. Také je nutno počítat, že je algoritmus nastaven na velikost srdce a vytváří hranice zhruba 6 centimetrů od maxim a minim center segmentů v každé ose. Pro větší struktury by bylo nutné tyto parametry změnit v python souboru. Ukázka nevhodně zadaných vstupů je znázorněna na obrázku *Obr 3.7*. Vstupy na obrázku nemají reprezentovat určitou anatomickou strukturu, ale pouze demonstrovat špatně zadaný vstup.



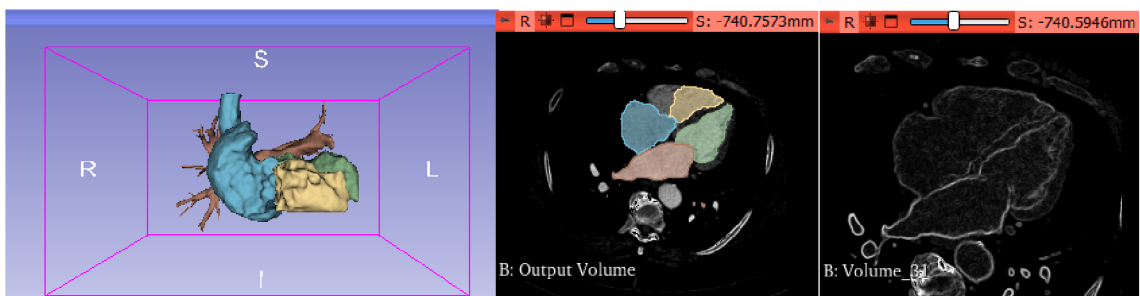
Obr. 3.7: Ukázka nevhodně zvolených vstupů s vygenerovaným segmentem pozadí

3.4.1 Gradient filtr

Pro aplikaci watershed filtru je zapotřebí upravit vstupní obraz. Vstupem by měl být vhodný parametrický obraz, který je vnímán jako reliéf. Pro získání tohoto parametrického obrazu byl využit *GradientMagnitudeRecursiveGaussian* z knihovny SimpleITK (dále jako GradientMagnitude filtr). K získání hranové reprezentace obrazu byl také vyzkoušen filtr *SobelEdgeDetectionImageFilter*, jenž je také součástí SimpleITK knihovny (dále jako SobelEdge filtr). Porovnání výsledného parametrického obrazu obou filtrů lze vidět na obrázku *Obr. 3.8* a *Obr. 3.9*. GradientMagnitude filtr dosahuje přesnějších výsledků segmentace což je i viditelné na přiložených obrázcích. Největší rozdíly v segmentaci jsou patrné v segmentaci pravé síně. Filtry byly testovány na identických vstupech. Výhodou GradientMagnitude filtru je také možnost měnit parametr Sigma, který může výrazně ovlivnit výslednou segmentaci. Vyšší hodnota tohoto parametru zabraňuje „protékání“.



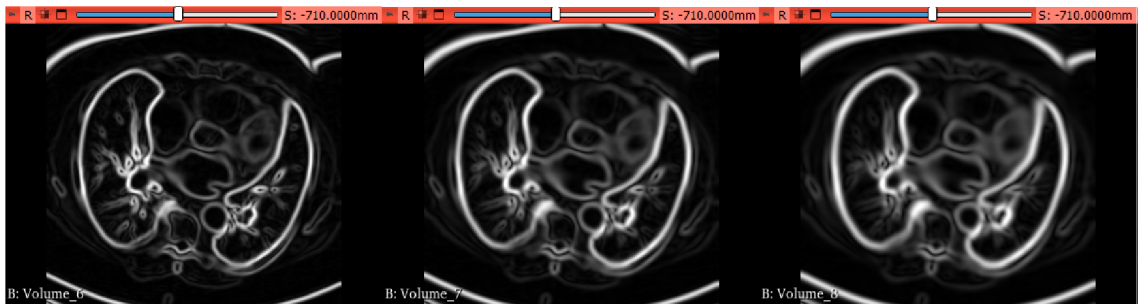
Obr. 3.8: GradientMagnitude filtr (zleva do prava): 3D zobrazení segmentace, zobrazení segmentace v transverzální rovině, parametrický obraz získaný filtrem



Obr. 3.9: SobelEdge filtr(zleva do prava): 3D zobrazení segmentace, Zobrazení segmentace v transverzální rovině, parametrický obraz získaný filtrem

Vstupem pro GradientMagnitude filtr je *itkImage*. Pro získání tohoto vstupu je nutné převést objem („*vtkMRMLScalarVolumeNode*“) ze Sliceru. *ItkImage* byl získán pomocí funkce *sitUtils.PullVolumeFromSlicer*. Druhým vstupem je již zmíněná

Sigma (typ double), která je defaultně nastavená na 2.3 (2.0 v nativním modulu Segment Editor). Rozdílné výsledky parametrického obrazu získané odlišným parametrem sigma jsou znázorněny na obrázku *Obr. 3.10*.



Obr. 3.10: Parametrické obrazy získané filtrem GradientMagnitude: Sigma = 1.8(vlevo) ,Sigma = 2.4(střed), Sigma = 3.0 (vpravo)

3.4.2 Watershed Segmentace

K segmentaci využíváme filtr *MorphologicalWatershedFromMarkersImageFilter*[22], dále jako Watershed filtr. Vstupem filtru je obraz získaný GradientMagnitude filtrem a obraz obsahující informace o segmentech. V obou případech se musí jednat o itkImage. Další 2 vstupy ovlivňují nastavení filtru. Jedná se o parametry FullyConnected a MarkWatershedLine. Tyto parametry mohou nabývat hodnot *True* nebo *False*.

Předzpracování vstupních obrazů

Pro urychlení výpočtů by segmentace neměla být aplikována na celý objem, a proto je nutné nejdříve upravit vstupní obrazy. Při zahájení segmentace je nejprve zavolána funkce *BoundingBoxAndCropping*. Tato funkce využívá modulu *VolumeStatistic*. Funkce získá informace o segmentu pozadí. Konkrétně jeho počátek, průměr a polohu. Podle získaného průměru se vytvoří nová ROI dané velikosti a nafituje se na segment pozadí. Nasledně je vytvořen nový prázdný objem typu „vtkMRMLScalarVolumeNode“. Tento objem je dále oříznut pomocí vytvořené ROI využitím logické funkce *CropInterpolated* z modulu *CropVolume*. Interpolační metoda je zvolena na NN (Nearest neighbor) a rozložení obrazu je zachováno. Pro správné fungování tohoto procesu je nutné mít segment reprezentující pozadí pojmenovaný jako „Background“. V opačném případě bude segmentace aplikována na celý objem, což zvyšuje výpočetní náročnost. Na oříznutý objem je poté aplikován thresholding s horní hranicí 950 a s dolní hranicí -50. Původní hodnoty si zachovají pouze voxely, které se nachází v tomto rozmezí. Obraz je poté převeden do formátu itkImage a je na něj aplikován GradientMagnitude filtr (viz. 3.4.1 Gradient filtr). Pro vytvoření obrazu

obsahující reprezentace segmentů v objemu je nutné vytvořit nový prázdný objem typu „vtkMRMLLabelMapVolumeNode“. Do tohoto objemu jsou exportovány segmenty zadané uživatelem. Jako referenční obraz je zádán již oříznutý první vstupní obraz. To zajistí stejnou velikost vstupních obrazů. Poslední krokem je převedení obrazu obsahující informace o segmentech do formátu vtkImage.

Aplikování Watershed filtru

Výstupem watershed filtru je itkImage obsahující reprezentaci segmentů. Výstupní itkImage je převeden zpět do „vtkMRMLLabelMapVolumeNode“, který byl vytvořen pro vstupní objem obsahující segmenty. Z tohoto uzlu importujeme segmenty do nově vytvořeného uzlu „vtkMRMLSegmentationNode“. Tato segmentace je nově vytvořená a neobsahuje displayNode, který uchovává informace jak se mají segmenty zobrazovat. Proto je pro Segmentaci vytvořen nový displayNode. Viditelnost všech segmentů je nastavena, a to jak v normálním zobrazení v řezech, tak i ve 3D. Následně je viditelnost segmentu „Background“ vypnuta. To zajistí zviditelnění všech segmentů krom segmentu pozadí. Toto nastavení nezajistí zobrazení segmentů ve view. K zobrazení segmentů ve view je pro displayNode nastaven konkrétní ViewNode. Je předpokládáno, že uživatel má defaultní nastavení pro View. To je jedno zobrazení pro každý řez a jedno 3D zobrazení. Zobrazení je funkční i pro jiné rozložení, než je Conventional rozložení. Je však nutné, aby jednotlivé ViewNodes měli původní ID. Například pokud uživatel využívá dvoje zobrazení Red view (transverzální zobrazení), bude segmentace vyobrazena pouze v jednom (v uzlu, který je v hierarchii nejvýše). Zobrazení lze manuálně měnit v segment editoru nebo v Data modulu.

Postprocessing

Slicer nabízí funkce, které pomáhají dokončit výslednou segmentaci. Na segmentaci jsou automaticky aplikovány funkce Margin, Smoothing a Islands. Funkce jsou součástí segment editoru. Operace s těmito funkcemi jsou obsaženy ve funkci *DilatationErosion*. Jak už název napovídá funkce Margin je zde použita pro dilataci a následnou erozi segmentů. Funkce Margin má parametr „MarginSizeMm“, který ovlivňuje o kolik se bude segment zvětšovat a zmenšovat. Pro dilataci byl tento parametr nastaven na hodnotu 4.0. Pro erozi je tento parametr nastaven na hodnotu -4.0. Tyto operace zajišťují vyplnění případných děr, které se nachází v segmentu, také má malý vyhlazující efekt. Poměr parametrů pro dilataci a erozi je 1:1, což zajišťuje přibližně stejnou velikost segmentů jako před použitím funkce. Konečné vyhlazení segmentu je provedeno pomocí funkce Smoothing. Pro aplikaci funkce Smoothing je nutné zadat metodu vyhlazování a velikost kernel filtru. Nastavená metoda je typu Median, která odstraní malé výběžky a vyplní malé díry na povrchu segmentu. Velikost kernel filtru byla nastavena na hodnotu 3 milimetry. Poslední

využitou funkcí je *Islands*. Druh operace je nastaven na *Keep largest island*. Po aplikování jsou vymazány všechny kousky segmentu, které se nedotýkají hlavního největšího segmentu.

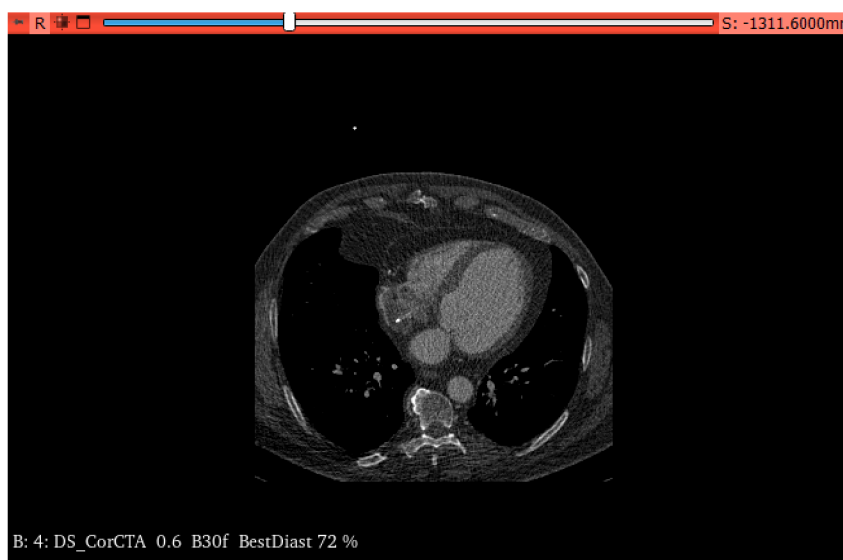
Ukončení segmentace

Segmentace musí být ukončena tlačítkem *Finish Segmentation*. Do té doby je možné aktualizovat Segmentaci. Tlačítko *apply segmentation* je nahrazeno tlačítkem *Update*. Uživatel vidí výslednou segmentaci v jednotlivých řezech, avšak v segment editoru je stále nastavená segmentace původních vstupů (preview). Pomocí paint může uživatel upravit vstup pro jednotlivé segmenty. Tlačítko *Update* poté opět spustí watershed segmentaci na nové vstupy. Po ukončení segmentace jsou vymazány všechny objekty, které byly vytvořeny a využity pro získání segmentace. Jedná se například o oříznutý obraz. Uživatel by měl vždy dokončit segmentaci, než začne se získanou segmentací pracovat. Odstranění segmentace před jejím dokončením způsobí znehodnocení parametrů a modul musí být poté restartován.

4 Testování modulu

Data

Použité snímky byly pořízeny pomocí Výpočetní tomografie. Jedná se o snímky hrudníku, které obsahují anatomickou strukturu srdce a jeho okolí. K testování byly využity 2 datasety. Dataset AGEKANONIX je pořízen pomocí CT s 16 řady detektorů, obsahuje snímky srdce s mírnou kalcifikací koronárních tepen. Součástí je snímání srdce v 10 různých fázích. Snímky jsou velikosti 512 x 512 x 51. Dataset ARTIFIX obsahuje snímky srdce s rozšířenou aortou. Obsahuje tři normální snímky a jedno snímání v multiphase, které je jiného typu a při testování nebylo využito. Snímky jsou velikosti 512 x 512 x 347. Oba datasety jsou dostupné na z webových stránek Osirix viewer[23]

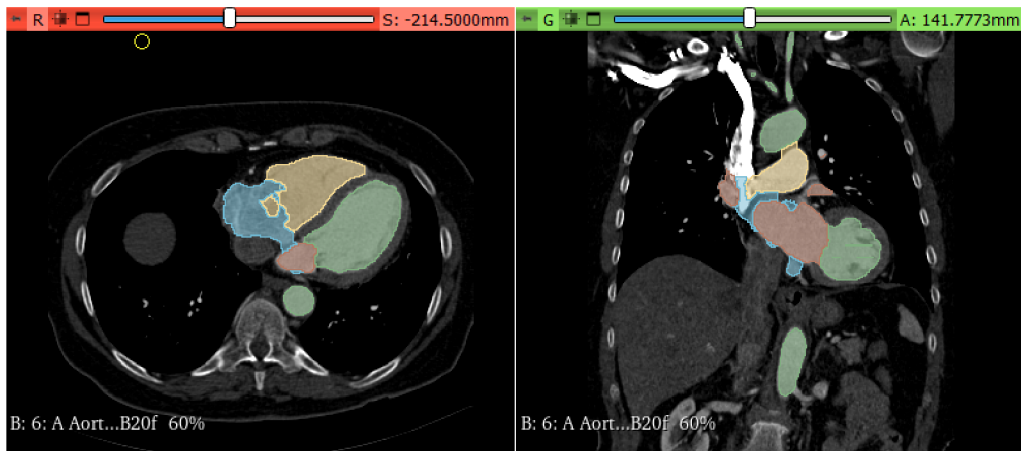


Obr. 4.1: CT snímek hrudníku

4.1 Testování segmentace na datech

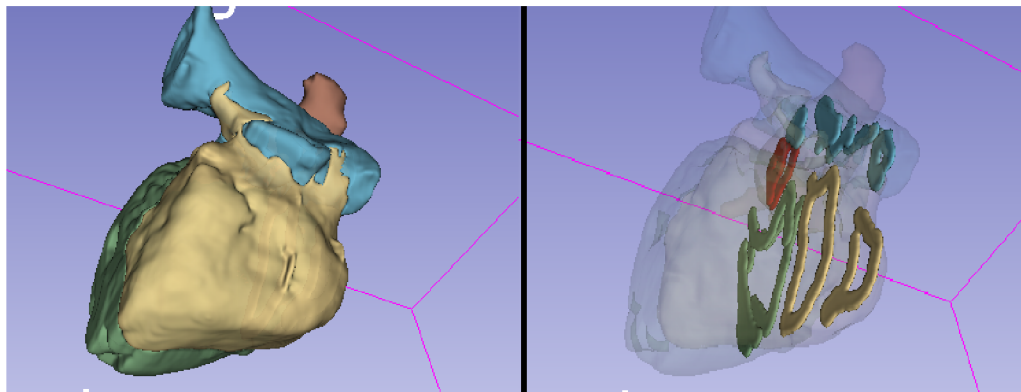
Při testování modulu byla otestována funkcionality na jednotlivé segmenty i na srdce jako celek. Ke správné segmentaci na první pokus je zapotřebí kvalitní CT snímek s dobře rozlišitelnými částmi srdce. Nejlepší výsledků, bylo dosaženo na snímkách srdce v diastole. V případě, kdy je srdce v jiné fázi je zapotřebí mnohem více editace a vstup musí být vícekrát aktualizován. Segmentace levé komory je silně ovlivněna na aktuální fázi srdce. Svalovina nabývá vyšších hodnot a vytváří výrazný rozdíl mezi výdutí komory a jejími okraji. Tento problém je však možné korigovat vhodným

zadáním vstupu. Na obrátku *Obr. 4.2* je ukázka vysegmentování levé komory bez její svaloviny.



Obr. 4.2: Segmentace srdce s nedokončenou segmentací levé komory

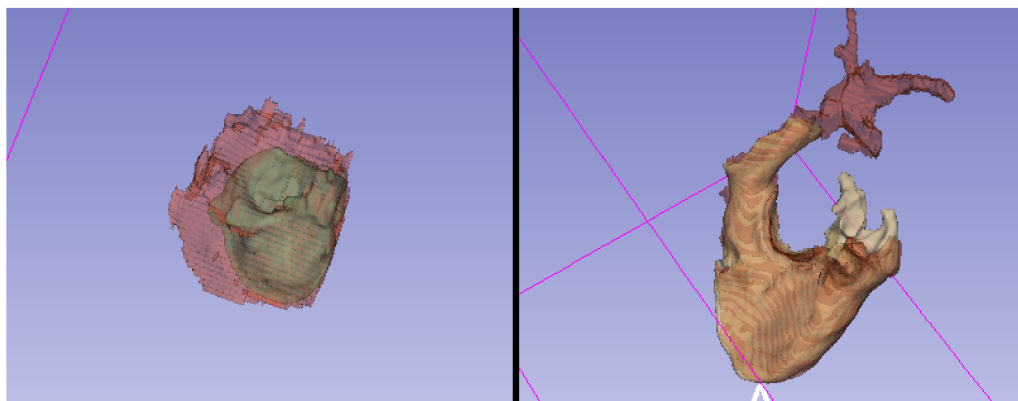
Segmentace pravé komory a pravé síně zpravidla vyžadovala více editace, kdy jeden ze segmentů často zasahuje do oblasti segmentu druhého. Především segmentace pravé síně je náročnější, protože se segment často omotává kolem aorty. Tento problém lze nejjednodušeji vyřešit pomocí nástroje Scissors. Názorná ukázka rozdílů mezi vstupy pro jednotlivé části srdce je vyobrazena na obrázku *Obr. 4.3*.



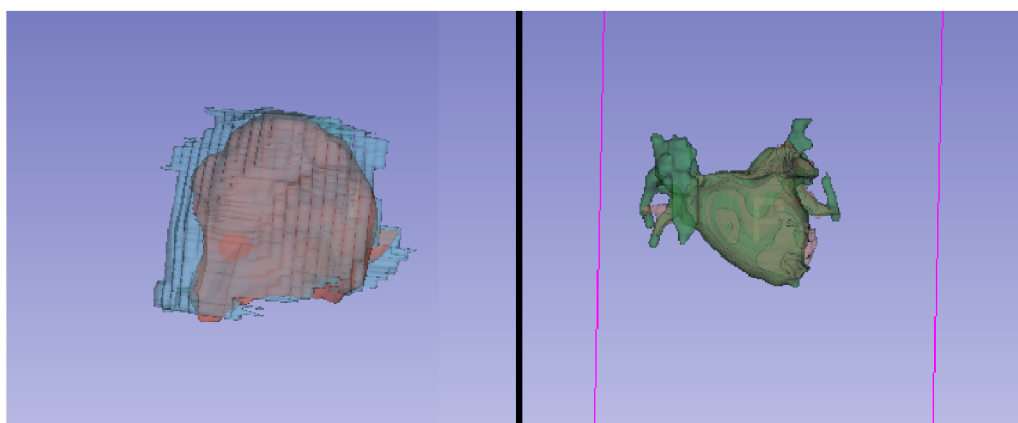
Obr. 4.3: Rozdíl mezi zadanými vstupy pro jednotlivé části srdce: Modrá - pravá síň, Žlutá - pravá komora, Červená - Levá síň, Zelená - Levá komora

Kvalita segmentace je často ovlivněna podobou vstupů. Pro objektivní porovnání se segmentací, kterou nabízí segment editor byla nejdříve vytvořena segmentace v modulu představeném v této práci a poté byla vytvořena segmentace pomocí segment editoru se stejnými vstupy. Segment editor využívá pro segmentaci watershed stejný typ filtru a bez jakéhokoli předzpracování, nastavení a post-processingu by byly vytvořené segmentace stejné. V příložených obrázcích byl segment vytvořený

v modulu HeartSegmentation nastaven jako neprůhledný a na něj byl nasazen průhledný segment vytvořený segment editorem.



Obr. 4.4: Porovnání výsledků se segment editorem: Levá komora (vlevo), Pravá komora (vpravo)



Obr. 4.5: Porovnání výsledků se segment editorem: Pravá síň (vlevo), Levá síň (vpravo)

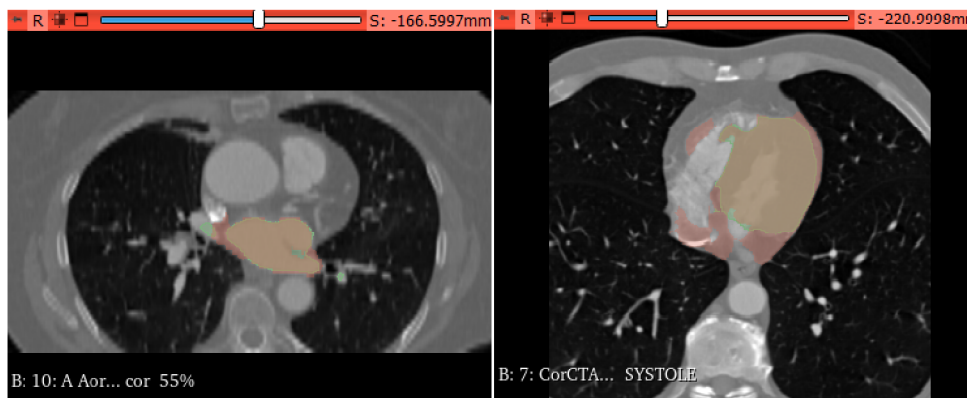
4.2 Diskuze

Funkcí specifickou pro tento modul je vytváření segmentu pozadí, který je vždy pojmenován „Background“. Generování tohoto segmentu je založeno na pozici ostatních segmentů, které zadal uživatel. Velikost segmentu pozadí je limitována pouze pozicí ostatních segmentů. 3D Sliceru tuto funkci nenabízí a uživatel je nucen vytvářet segment pozadí a to kolem celého segmentu minimálně ve 2 osách. Další výhodou modulu HeartSegmentation, avšak ne příliš významnou, je automatická aktivace efektů segment editoru. Například při automatickém vytváření segmentů

je pro uživatele aktivován nástroj Paint. Kvalitní segmentování pomocí watershed, kterou nabízí 3D Slicer je možná, ale vyžaduje větší množství editace a manuální předzpracování obrazu. Součástí segmentace v modulu HeartSegmentation, je také automatický post-processing. Využívané funkce jsou běžnou součástí segment editoru a výsledné segmentaci lze dosáhnout i bez použití modulu. Tento proces je pro uživatele však velice zdoluhavý. Pro post-processing byla vytvořena funkce využívající tři nástroje. Jedná se o Margin, Islands a smoothing. Nastavení jednotlivých operací jsou optimalizována pro segmentaci srdce.

3D Slicer je open source aplikace a to umožňuje přístup ke kódu všech modulů. Díky tomuto faktu je možné nahlédnout do fungování jednotlivých funkcí a upravovat je pro konkrétní využití. Hlavní část 3D Sliceru je napsána v C++, díky MRML architektuře jsou však i tyto funkce dostupné a je možné je volat pomocí kódu v pythnu. Měnit funkcionalitu těchto funkcí ale není možné za použití python skriptování. Dle výsledků z watershed 3D Slicerem je viditelné, že také používá nějaký druh ostříhnutí vstupního obrazu. Tento proces však není součástí zdrojového python scriptu watershed filtrace a jedná se tedy o vnitřní funkci, kterou modul využívá.

K nejlepším výsledkům dosahovala segmentace Levé síně. Lepší výsledky jsou nejpravděpodobněji způsobeny malou tloušťkou srdeční stěny a dostatečnou odlišností od hodnot pozadí. Naopak nejhorších výsledků dosahovala segmentace pravé síně. Problematika je již v samotném zadávání minim, kdy hodnoty pravé komory mohou splývat s hodnotami pozadí a hodnotami Pravé komory. Pro oddělení struktur pravé komory a pravé síně je doporučeno uživateli použít více specifické vstupy, které pomohou algoritmu struktury úspěšně oddělit. Při zadávání vstupů je doporučeno využít přednastavený window/level, jenž zjednodušuje zadávání jednotlivých vstupů.



Obr. 4.6: Porovnání výsledků se segment editorem: Levá síň (vlevo), Levá komora (vpravo),

Závěr

V diplomové práci byla provedena rešerše známých softwarů pro práci s medicínskými 3D obrazy. V teoretické části je také podrobněji popsána struktura a zabudované funkce 3D sliceru. Hlavním zaměřením bylo na vysvětlení MRML struktury a popis základních nabízených modulů a možnosti jejich rozšíření pomocí skriptů v pythnu. V praktické části bylo vytvořeno rozšíření s modulem *HeartSegmentation* a jeho funkcionality byla definována pomocí skriptování v pythnu. Grafické rozhraní pro modul bylo vytvořeno v QT designeru. Modul je rozdělen do tří částí pojmenovaných Preprocessing, Segmentation a Create Mask. V části preprocessing modul nenabízí nic co by ve 3D sliceru nebylo již přítomno. Parametry pro uživatele jsou již předem nastavené. Při využívání modulu se nepočítá, že by uživatel prováděl velké operace pro předzpracování obrazu. K tomu má ve 3D sliceru mnoho jiných modulů, ve kterých lze nastavovat specifické parametry. Funkce v této části slouží pouze jako rychlá možnost bez potřeby využívat jiný modul. Filtry pro eliminaci šumu jsou nastaveny pouze na mírné zašumnění obrazu. Je předpokládáno že pro filtraci velkého šumu a různých artefaktů uživatel využije jiné moduly, které se zaměřují na tuto problematiku. Vzhledem k tomu, že modul využívá pro podvzorkování a ostříhnutí obrazu logickou funkci z modulu *Crop volume* je funkcionality s tímto modulem téměř totožná. Modul *Crop volume* samozřejmě nabízí i jiné funkce a větší míru specifičnosti pro zadávání jednotlivých parametrů. V části segmentace bylo vytvořeno hned několik funkcí, které by uživateli měli zjednodušit segmentování srdce. Součástí je konstruktor, jenž vytváří novou segmentaci se specifickými segmenty, které uživatel vybral pro segmentování. Vytvořené segmenty jsou patřičně pojmenovány dle výběru uživatele. Výběr možností pro automatickou generaci segmentů je však limitován na pět. Z toho maximálně čtyři mohou být vykonstruovány najednou. Vkládání dalších segmentů, například segment pro aortu je možný, ale musí být vložen manuálně pomocí segment editoru. Modul obsahuje algoritmus pro automatické generování pozadí segmentu. Tato funkce je dostupná jak pro data vytvořená uživatelem v modulu, tak i pro již existující segmentace, které uživatel získal nebo vytvořil jiným způsobem. Při testování tato funkce dosahovala vysoké přesnosti správného vytvoření segmentu pozadí.

Cílem modulu je usnadnit uživateli práci při segmentaci srdce a co nejvíce automatizovat segmentování. Modul oproti 3D sliceru vyžaduje od uživatele mnohem méně vstupů. Vytvořené funkce pro aplikování segmentace využívají pre-processing a post-processing pro získání kvalitní segmentace. Přestože je vstup od uživatele udržen na minimu, metodu není možné plně automatizovat. Toto by bylo možné pouze implementací umělé inteligence, která by byla naučená na konkrétní modely. Nevýhodou v modulu může být příliš velká specifičnost nastavených parametrů.

Algoritmy jsou optimalizovány na strukturu a velikost srdce. V případě větších anatomických anomálií jako extrémně velké srdce nebo kalcifikace myokardu může toto vést k méně kvalitním segmentacím. Při testování byla segmentace aplikována na data a výstup byl porovnán se segmentací, kterou nabízí 3D slicer. Modul HeartSegmentation dosahoval ve velké míře lepších výsledků na zadaný vstup, než metoda nabízená v Segment editoru. Využití automatického generování pozadí je rychlé a snadné, ale uživatel by na tuto funkci neměl spolehat a v případě horší kvality snímků upravit segment pozadí pro získání lepší segmentace.

Modul se snaží hlídat všechny vstupy, které jsou zadány a korigovat uživatele v případě, kdy vstupy nejsou kompletní. Uživatel může provést operace, která modul neočekává, což může způsobovat nepředvídatelné chování modulu. K omezení výskytu těchto událostí jsou proto některé prvky v GUI při různých operacích schovány, nebo zneprístupněny. Zobrazování dat jako reprezentace segmentací ve 3D sliceru je poměrně komplikovaná a řídí se podle hierarchie. Z tohoto důvodu není v modulu automaticky generována 3D reprezentace pro vytvořenou segmentaci srdce. Místo toho má uživatel přístup k zobrazování jednotlivých segmentů pomocí segment editoru. Rozšiřování 3D sliceru pomocí skriptování v pythnu má téměř nevyčerpatelný potenciál. Vytvořený modul v této práci lze dále rozšiřovat a upravovat jeho vzhled a funkcionalitu. Funkcionalita modulu by například mohla být rozšířena přidáním další části, která by obsahovala možnosti upravovat parametry segmentace. K velkému množství nastavení nemá uživatel přístup a jsou tvrdě nastavená v kódu jednotlivých funkcí. Mezi dostupná nastavení by například mohlo patřit nastavení velikosti segmentu pozadí, míra protahování obrazu nebo nastavení jestli má algoritmus dělat post-processing automaticky.

Literatura

- [1] Rosset, A., Spadola, L. and Ratib, O. *OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images*. Medical Image Analysis, Volume 33, 2004, Pages 205-216, <<https://doi.org/10.1007/s10278-004-1014-6>>
- [2] Rueden, C.T., Schindelin, J., Hiner, M.C. et al. *ImageJ2: ImageJ for the next generation of scientific image data*. BMC Bioinformatics, Volume 18, 2017, article number 529, <<https://doi.org/10.1186/s12859-017-1934-z>>
- [3] Schindelin, Johannes and Rueden, Curtis T. and Hiner, Mark C. and Eliceiri, Kevin W. *The ImageJ ecosystem: An open platform for biomedical image analysis*. Molecular Reproduction and Development, Volume 82, 2015, Pages 518-529, <<https://doi.org/10.1002/mrd.22489>>
- [4] Marcus, D.S., Olsen, T.R., Ramaratnam, M. et al. *The extensible neuroimaging archive toolkit*. Neuroinform, Volume 5, 2007, Pages 11-33, <<https://doi.org/10.1385/NI:5:1:11>>
- [5] NRG Lab - the Washington University School of Medicine *XNAT - Home [online]*. [cit. 22. 12. 2022], Dostupné na <<https://www.xnat.org/>>
- [6] Jess Tate, Brett Burton, Ayla Khan *Seg3D Basic Functionality*. [cit. 22. 12. 2022], Dostupné na <<https://www.sci.utah.edu/cibc>>
- [7] CIBC - Volumetric Image Segmentation and Visualization. Scientific Computing and Imaging Institute (SCI) *OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images* [online]. [cit. 19. 12. 2022], Dostupné na <<http://www.seg3d.org>>
- [8] Paul A. Yushkevich and Joseph Piven and Cody Hazlett, Heather and Gimpel Smith, Rachel and Sean Ho and James C. Gee and Guido Gerig. *User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability*. Neuroimage, Volume 31, Issue 3, 2006, Pages 1116-1128, ISSN 1053-8119, <<https://doi.org/10.1016/j.neuroimage.2006.01.015>>.
- [9] Bruce Fischl, *FreeSurfer*, NeuroImage, Volume 62, 2012, Pages 774-781, Dostupné na <<https://doi.org/10.1016/j.neuroimage.2012.01.021>>
- [10] Nicolas Toussaint, Jean-Christophe Souplet, Pierre Fillard. *MedINRIA: Medical Image Navigation and Research Tool by INRIA*. Proc. of MICCAI'07 Workshop on Interaction in medical image analysis and visualization, 2007, Brisbane, Australia, Australia. ffinria-00616047

- [11] MIPAV *Medical Image Processing, Analysis and Visualization [online]*. [cit. 9. 12. 2022], Dostupné na <https://mipav.cit.nih.gov/pubwiki/index.php/Main_Page>
- [12] Object Research Systems (ORS) Inc *3D Visualization and Analysis Solutions for Scientific and Industrial Data | ORS* [online], [cit. 7. 12. 2022], dostupné na <<https://www.theobjects.com/dragonfly/index.htmlz>>
- [13] Tina Kapur and Steve Pieper and Andriy Fedorov and J-C Fillion-Robin and Michael Halle and Lauren O'Donnell and Andras Lasso and Tamas Ungi and Csaba Pinter and Julien Finet and Sonia Pujol and Jayender Jagadeesan and Junichi Tokuda and Isaiah Norton and Raul San Jose Estepar and David Gering and Hugo J.W.L. Aerts and Marianna Jakab and Nobuhiko Hata and Luiz Ibanez and Daniel Blezek and Jim Miller and Stephen Aylward and W. Eric L. Grimson and Gabor Fichtinger and William M Wells and William E. Lorensen and Will Schroeder and Ron Kikinis, *Increasing the impact of medical image computing using community-based open-access hackathons: The NA-MIC and 3D Slicer experience*. Digit Imaging, Volume 17, 2016, Pages 176-180, ISSN 1361-8415, dostupné na <<https://doi.org/10.1016/j.media.2016.06.035>>
- [14] Andriy Fedorov and Reinhard Beichel and Jayashree Kalpathy-Cramer and Julien Finet and Jean-Christophe Fillion-Robin and Sonia Pujol and Christian Bauer and Dominique Jennings and Fiona Fennessy and Milan Sonka and John Buatti and Stephen Aylward and James V. Miller and Steve Pieper and Ron Kikinis, *3D Slicer as an image computing platform for the Quantitative Imaging Network*. Magnetic Resonance Imaging, Volume 30, 2012, Pages 1323-1341, ISSN 0730-725X, dostupné na <<https://doi.org/10.1016/j.mri.2012.05.001>>
- [15] 3D Slicer image computing platform *3D Slicer image computing platform | 3D Slicer [online]*, [cit. 7. 12. 2022], dostupné na <<https://www.slicer.org>>
- [16] 3D Slicer image computing platform - Slicer Wiki *Adding MRML [online]*. [cit. 8. 12. 2022], dostupné na <https://www.slicer.org/wiki/Adding_MRML>
- [17] Yoo, Terry and Ackerman, Michael and Lorensen, William and Schroeder, William and Chalana, Vikram and Aylward, Stephen and Metaxas, Dimitris and Whitaker, Ross *Engineering and Algorithm Design for an Image Processing API: A Technical Report on ITK - the Insight Toolkit*, Studies in health technology and informatics, volume 85, 2002, Page 586-592, <<https://doi.org/10.3233/978-1-60750-929-5-586>>

- [18] C. Bane Sullivan and Alexander A. Kaszynski *PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)*, Journal of Open Source Software, volume 4, 2019, Page 1450, dostupné na <<https://doi.org/10.21105/joss.01450>>
- [19] Csaba Pinter and Andras Lasso and Gabor Fichtinger *Polymorph segmentation representation for medical image computing*. Computer Methods and Programs in Biomedicine, Volume 171, 2019, artical number 529, dostupné na <<https://doi.org/10.1016/j.cmpb.2019.02.011>>
- [20] 2023 The QT Company *Qt Designer Manual/ QT Designer [online]*, [cit. 10. 05. 2023], dostupné na <<https://doc.qt.io/qt-6/qtdesigner-manual.html>>
- [21] Efficient implementation of kernel filtering The Insight Journal, 2007 MICCAI Open Science Workshop. dostupné na <<https://doi.org/10.54294/igq8fn>>
- [22] The watershed transform in ITK - discussion and new developments The Insight Journal, 2006, dostupné na <<https://doi.org/10.54294/1f8u75>>
- [23] DICOM image sample sets OsiriXDicom [online], [cit. 14. 05. 2023], dostupné na <<https://www.osirix-viewer.com/resources/dicom-image-library>>

Seznam symbolů a zkratek

PET	Pozitronová emisní tomografie
DICOM	Digital Imaging and Communications in Medicine
PACS	Picture Archiving and Communications System
ROI	Region of interest
VTI	Vertebral Tortuosity Index
AIAA	Visualization Toolkit
XNAT	Extensible Neuroimaging Archive Toolkit
NIH	National Institutes of Health
ITK	Insight Toolkit
VTK	The Visualization Toolkit
fMRI	funkční magnetická rezonance
MRI	magnetická rezonance
CT	computed tomography
ROI	region of interest
MRML	Medical Reality Modeling Language

Seznam příloh

A	Ukázky zdrojového kódu	49
B	Elektronická příloha	51
B.1	HeartSegmentation.py	51
B.2	Heart-WatershedSegmentation	51
B.3	PřidáníModulu.docx	51
B.4	Segmentace	51
B.5	Watershed.png	51

A Ukázky zdrojového kódu

A.1: Funkce *initiateSegmentEditor* v logické části

```
1 def initiateSegmentEditor(self, inputVolume):
2     self.segmentationNode = slicer.mrmlScene.AddNewNodeByClass(
3         ↪ "vtkMRMLSegmentationNode", "preHeart_□Segmentation")
4     if self.segmentEditorWidget == None:
5         self.segmentEditorWidget = slicer.
6             ↪ qMRMLSegmentEditorWidget()
7         self.segmentEditorWidget.setMRMLScene(slicer.mrmlScene)
8         self.segmentEditorNode = slicer.mrmlScene.
9             ↪ GetSingletonNode("SegmentEditor", "
10             ↪ vtkMRMLSegmentEditorNode")
11         self.segmentEditorWidget.setMRMLSegmentEditorNode(self.
12             ↪ segmentEditorNode)
13
14     self.segmentEditorWidget.setSegmentationNode(self.
15         ↪ segmentationNode)
16     self.segmentEditorWidget.setSourceVolumeNode(inputVolume)
17
18     self.segmentationNode.CreateDefaultDisplayNodes()
19
20     CurrentSegmentation = self.segmentationNode
21     return CurrentSegmentation
```

A.2: Příklad připojení widgetů

```
1 self.ui.SegmentsInNode.connect("currentIndexChanged(QString)",
2     ↪ self.updateParameterNodeFromGUI)
3
4 self.ui.NodeBoxROI.connect("currentNodeChanged(vtkMRMLNode*)",
5     ↪ self.updateParameterNodeFromGUI)
6
7 # connections - spinBox
8 self.ui.samplingSpinBox.connect("valueChanged(double)", self.
9     ↪ updateParameterNodeFromGUI)
10 self.ui.ListofFilters.connect("currentIndexChanged(int)", self.
11     ↪ updateParameterNodeFromGUI)
12
13 # connections - checkBox
14 self.ui.filtercheckBox.connect("toggled(bool)", self.
15     ↪ updateParameterNodeFromGUI)
16 self.ui.DownsamplingcheckBox.connect("toggled(bool)", self.
17     ↪ updateParameterNodeFromGUI)
```


A.3: Funkce volaná tlačítkem Start Segmentation

```
1
2 def startSegmentation(self):
3     self.ui.CollapsiblePreprocessingButton.collapsed = True
4     self.ui.groupBox.hide()
5     self.ui.CancelButton.enabled = True
6     self.ui.StartSegmentationButton.hide()
7     self.activeSegments()
8
9     ActiveSegments = self.\_parameterNode.GetParameter("
10     ↪ activeSegments")
11     if ActiveSegments == "00000":
12         ActiveSegments = "10000"
13     int\_list = [int(s) for s in ActiveSegments]
14     for i in range(0,5):
15         if int\_list[i]==1:
16             if i == 0:
17                 NameOfSegment = "Unspecified"
18                 ActiveSegments = "00000"
19                 self.\_parameterNode.SetParameter("activeSegments",
20                 ↪ ActiveSegments)
21             if i == 1:
22                 NameOfSegment = "LeftVentricle"
23                 self.activeSegmentChangeStatus(1,"0")
24             if i == 2:
25                 NameOfSegment = "RightVentricle"
26                 self.activeSegmentChangeStatus(2,"0")
27             if i == 3:
28                 NameOfSegment = "LeftAtrium"
29                 self.activeSegmentChangeStatus(3,"0")
30             if i == 4:
31                 NameOfSegment = "RightAtrium"
32                 self.activeSegmentChangeStatus(4,"0")
33
34     # calls logic function, set up segment editor for use
35     CurrentSegmentation = self.logic.initiateSegmentEditor(
36     ↪ self.ui.inputSelector.currentNode())
37     self.\_parameterNode.SetNodeReferenceID("Segmentation",
38     ↪ CurrentSegmentation.GetID())
39     # creates first segment
40     self.logic.InputForSegments(NameOfSegment,
41     ↪ CurrentSegmentation)
42     break
```

B Elektronická příloha

B.1 HeartSegmentation.py

Obsahuje zdrojový kód modulu. Funkce ve zdrojovém kódu jsou stručně popsány. Obsahuje kód pro vytvoření a nastavení GUI modulu a logické funkce určující funkcionalitu modulu.

B.2 Heart-WatershedSegmentation

Jedná se o složku vytvořeného rozšíření. Obsahuje nezbytné soubory pro správné načtení a fungování ve 3D sliceru včetně zdrojového kódu.

B.3 PřidáníModulu.docx

Obsahuje návod jak přidat modul do 3D sliceru a instrukce k používání.

B.4 Segmentace

Složka obsahuje již vyhotovené segmentace. Referenční obrazy nejsou přítomny.

B.5 Watershed.png

Obsahuje diagram procesu segmentace.