Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

CT ScanArt : Aplikácia pre vizualizáciu a interpretáciu CT skenov v reštaurátorskej praxi

Bakalárska práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	RNDr. Zuzana Berger Haladová, PhD.

Bratislava, 2025
Klára Senderáková

Obsah

Zoznam obrázkov

Kapitola 1

Úvod do problematiky

1.1 Nedeštruktívne metódy v reštaurátorskom výskume

Využitie CT snímkov pri reštaurovaní historických objektov umožňuje detailnú analýzu vnútornej štruktúry objektov bez potreby ich poškodenia. Jedným z kľúčových aspektov tejto technológie je možnosť odhalenia skrytých poškodení, opráv a historických zásahov, ktoré nie sú viditeľné voľným okom. Týmto spôsobom je možné získať informácie o stave a histórii objektov, čo je neoceniteľnou informáciou pri ich reštaurovaní a konzervácii.[33] Využívanie nedeštruktívnych metód umožňuje zachovať integritu a funkčnosť skúšaných objektov, čo je obzvlášť dôležité pri historických a umeleckých dielach, kde je potrebné minimalizovať akékoľvek riziko poškodenia.[37] Tomografia presne rekonštruuje údaje o rozložení denzity materiálu v každom mieste a poskytuje digitálne spracovanie týchto údajov.[37]

1.2 Snímanie objektov prostredníctvom počítačovej tomografie

Pre základy počítačovej tomografie bol dôležitý objav nového druhu žiarenia, nazvaného röntgenové lúče, ktorý popísal držiteľ prvej Nobelovej ceny za fyziku C. W. Röntgen. Prvým zhotoveným snímkom a nahliadnutím dovnútra človeka bola snímka ruky jeho manželky, ktorá okrem jasných kontúr kostí a svalov zachytila aj objekt - snubný prsteň. Ďalším dôležitým bodom v histórii CT tomografu bola publikácia rakúskeho matematika J. K. A. Radona,[34] v ktorej opísal takzvanú Radonovu transformáciu, ktorá tvorí jeden z kľúčových podkladov pre tomografickú rekonštrukciu. Koncepcia Radonovej transformácie je v teórii tomografie nesmierne dôležitá. Výstupom z tomografu totiž nie je priamo obrázok prierezu snímaného objektu, ale jeho zjednodušený Radonov obraz. Teória Radonovej transformácie sa tu uplatňuje zvlášť pri tomografickej rekonštrukcii, keď sa na nameraný Radonov obraz aplikuje inverzná Radonova transfor-

mácia a získava sa snímka prierezu študovaného objektu. [34] Teória na rekonštrukciu tomografického rezu z viacerých sumačných snímok bola vypracovaná v roku 1963 Allanom M. Cormackom.[36] Vzhľadom na jej komplikovanú technológiu rekonštrukcie sa zhotovenie prvého tomografu uskutočnilo až takmer o desať rokov neskôr. Moderná vyšetrovacia metóda, ktorá mala základy vo vlastnostiach röntgenového žiarenia, bola predstavená v roku 1971 anglickým inžinierom Godfreym Hounsfieldom.[35] Podľa neho bola pomenovaná stupnica denzity v priestore - Hounsfieldova jednotka(HU). [34]

1.2.1 Princíp fungovania CT prístrojov

Základom tomografickej metódy je zber a počítačové spracovanie veľkého množstva údajov, popisujúcich hodnotu absorpcie röntgenového žiarenia. Objekt je snímaný prístrojom, v ktorom rotuje zdroj röntgenového žiarenia, nazývaný röntgenka a rad detektorov postavených oproti sebe.[34] Súčasťou CT prístroja sú aj dva vysokovýkonné počítače. Riadiaci počítač má na starosti synchronizáciu a koordináciu funkcií celého zariadenia. Druhým počítačom je zobrazovací počítač, ktorý prijíma údaje v číslicovej podobe, ktoré sa prevádzajú z absolútnych číselných hodnôt na relatívne hodnoty odtieňa sivej. Vytváranie snímok pomocou röntgenového žiarenia je založené na schopnosti tohto žiarenia prenikať cez pevné látky. Ako žiarenie prechádza cez materiál, dochádza k jeho pohlcovaniu a intenzita žiarenia sa znižuje. Na detektore sa zaznamenáva intenzita preniknutého žiarenia, ktorá je vždy nižšia ako pôvodná intenzita. Meria sa pokles žiarenia počas prechodu objektom, čo predstavuje absorpciu. [34] CT obraz je tvorený dvojrozmernou sieťou pozostávajúcou zo štvorcov - pixelov, ktoré reprezentujú trojrozmerné hranoly, označované ako voxely. Počítačový tomograf rozdelí skúmaný objekt na tisíce malých kvádrov a v každom z nich meria absorpciu žiarenia. Výkonný počítač má za úlohu rekonštruovať plošný rez výpočtom veľkého množstva rovníc, čo sa nazýva počítačová tomografia. Po spracovaní množstva získaných súmárnych číselných údajov sa vytvorí číselná sieť (matica). Veľkosť tejto siete určuje priestorové rozlíšenie CT snímky.[34]

1.3 Zobrazovanie obrazu vo freewaroch

Pre účely zobrazovania medicínskych tomografov bol v roku 1993 vyvinutý dátový formát DICOM (Digital Imaging and Communications in Medicine), ktorý sa stal najpoužívanejším formátom. Na internete existuje mnoho freeware programov, ktoré umožňujú prezeranie DICOM súborov. Tie sú v porovnaní s ich plnými zakúpenými verziami obmedzené na niektoré funkcie. Medzi takéto freeware programy patria napríklad TomoCon PACS,[1] Image J,[2] Syngo fastView, [3]OSIRIX, [4]Volume Graphics [5] a mnohé ďalšie. Image J umožňuje prehliadanie 3D vstupov, no má užšiu škálu

meracích nástrojov, najmä pre HU jednotky v porovnaní s TomoCon PACS programom. Prvotné dáta, ktoré sú predmetom reštaurátorského výskumu, sú RAW dáta nasnímané priamo z CT prístrojov. Systém vypočíta z týchto RAW dát multiplanárne rekonštrukcie a projekcie maximálnej intenzity žiarenia. Pri práci s RAW dátami je zároveň možné vytvárať 3D náhľady pomocou techniky generujúcej objem a priamo ich analyzovať. Celé série 2D obrázkov zo snímania spolu s 3D záznamami sa po skončení práce ukladajú na CD nosič, avšak bez RAW dát. [34] Multiplanárne rekonštrukcie sú najjednoduchšou a najinformatívnejšou metódou zobrazenia. Pri 2D zobrazovaní CT snímok je transverzálny rez základom, pretože poskytuje najpresnejšie výsledky. Pomocou multiplanárnych rekonštrukcií môžeme získať aj ďalšie roviny, ako sú koronálna, sagitálna a semikoronálna.[34]

1.4 Druhy zobrazenia v 2D

Prínos tomografickej zobrazovacej metódy spočíva vo viacerých možnostiach vizuálneho prieskumu diel. Jednotlivé možnosti zobrazení umožňujú sledovať rôzne parametre.

1. MPR - multiplanárne zobrazenie poskytuje najväčšie množstvo informácií o zložení vzorky. Tieto zobrazenia je možné prezerať v troch náhľadoch: axiálnom, koronálnom a sagitálnom, pričom najpoužívanejšie sú axiálne snímky. Medzi najdôležitejšie funkcie pri prehliadaní multiplanárnych snímok patrí zviditeľňovanie vybraných denzít, alebo používanie prednastavení, ktoré poskytuje každý freeware pre DICOM súbory. Napríklad zobrazenie pre kosti, ktoré na sochách ukazuje polychrómiu, alebo zobrazenie pre pľúca zviditeľňuje na sochách hustotu. Ďalšou dôležitou funkciou je využívanie meracích nástrojov. Najdôležitejšími meracími parametrami sú : miery poškodenia dreveného nosiča - rozsah prasklín, objem dutín, epoxidových tmelov a rozmery klincov. V multiplanárnom zobrazení sa meria denzita v Hounsfieldových jednotkách a toto zobrazenie poskytuje možnosti rôznych farebných nastavení podľa hustôt. Najpoužívanejšie je inverzné zobrazenie a zobrazenie v Rainbow invert. V nich sa hustotám priraďujú rôzne farby - čierna farba značí najhustejšie miesta, červená poukazuje na miesta s najmenšou hustotou a biela predstavuje vzduch.

2. MIP - Maximum Intensity Projection zobrazuje voxely s najvyššou mierou absorpcie RTG žiarenia. Tie majú najvyššie Hounsfieldovo číslo, čo umožňuje zobrazenie materiálov s veľkou hustotou. Protikladom k tomuto zobrazeniu je takzvaná Minimum Intensity Projection, ktorá zobrazuje najnižšie hustoty. Toto zobrazenie napomáha v identifikácii výskytu pôvodnych polychrómií, sekundárnych doplnkov či pigmentov na báze kovov.

3. DDR - je digitálne rekonštruovaný rádiogram. Predstavuje reprojekciu konvenčného 2D röntgenového lúča vytvoreného z CT údajov. Keďže niektoré časti sa v klasickom 2D röntgenograme prekrývajú, nie je možné identifikovať konkrétne detaily. [34] Zlepšiť viditeľnosť je možné funkciou posun okna a kontrastu, ktorá je dôležitá najmä v multiplanárnych rekonštrukciách. Zviditeľňuje oblasti s vybranou denzitou materiálov v historickom objekte. Posunom získame konkrétnejšie informácie o oblastiach s nižšou, strednou a vysokou denzitou. Toto zobrazenie umožňuje vidieť štruktúru dreva.[34]

1.5 Softvérové zobrazenie výstupov z CT snímkov

V softvéri, môžeme zvoliť rôzne pohľady. Pri detailnom zväčšení je možné veľmi presné meranie rozmerov, napríklad hrúbky zobrazených vrstiev, ako aj veľkosti vzorky, vrátane malých segmentov pri meraní rôzne hustých oblastí vo vzorke. Používané sú vizualizácie ako maximum intensity projection, volume rendering, inverzné zobrazenie a ďalšie. Farby vzoriek či rezov sa dajú nastaviť podľa individuálnych potrieb. Farby v týchto zobrazeniach sú len ilustratívne a neodpovedajú autentickej farebnosti vzoriek, pretože výstup z CT je vždy v čiernobielej škále.[34] Niektoré možnosti freewaru myVGL vykresľujúce určitý algoritmus zahŕňajú:

- Isosurface renderer: Tento algoritmus zobrazuje povrch vybraného objektu definovaný iso-levelom (červená čiara v manipulačnom priestore). Produkuje vysokokvalitné fotorealistické obrazy v interaktívnych rýchlostiach takmer nezávisle od veľkosti súboru dát. Najčastejšie sa isosurface používa na vykresľovanie aktuálne určeného povrchu predmetu.

- Scatter HQ: Je vhodný na vizualizáciu slabých rozdielov hodnôt sivej v rámci voxelových dát, ako aj na vizualizáciu povrchových detailov štruktúry.

- Scatter: Vykresľuje objem a vizualizáciu transparentných štruktúr.

- X-ray: Algoritmus X-ray vrhá jeden lúč na jeden pixel dátového súboru. Čím vyššia je integrovaná nepriehľadnosť voxela pozdĺž lúča, tým tmavší je zodpovedajúci pixel (opačný je algoritmus Sum along Ray: nepriehľadnejší voxel je svetlejší).

- Maximum projection: Maximálna intenzita voxela pozdĺž lúča určuje sivú hodnotu priradenú pixelu.[34]

Kapitola 2

Návrh

2.1 Dôvod vývoja nového softvéru

Existujúce riešenia v podobe softvérových aplikácií boli vytvorené pre narábanie s medicínskymi dátami v medicínskom prostredí. Tieto aplikácie nie sú praktickým nástrojom pre reštaurátorskú prax, keďže obsahujú veľké množstvo nevyužívaných funkcií, častokrát v neintuitívnom používateľskom rozhraní. Softvér, ktorý by spĺňal požiadavky reštaurátorov, na realizáciu ich výskumu neexistuje. Aplikácia, ktorá je predmetom práce bude mať tri hlavné funkcionality: zobrazovanie, úprava a interpretácia CT snímkov.

2.1.1 Výhody nového softvéru

Predmetom tejto práce bolo vytvoriť alternatívu oproti exitujúcim softvérovým aplikáciám, ktorá bude mať jednoduché používateľské prostredie, v ktorom bude môcť užívateľ pracovať so súbormi typu DICOM. V aplikácii je na rozdiel od iných dostupných riešení možné zobraziť aj viaceré formáty súborov, ako napríklad .jpg, .png a ďalšie, ktoré slúžia reštaurátorom na prezeranie fotografií objektu pred aj po procese reštaurovania v jednej aplikácii. Aplikácia bude slúžiť reštaurátorom, ktorí nie sú dôkladne zorientovaní v oblasti využívania CT snímkov pre reštaurátorskú prax nakoľko táto technika nie je ešte veľmi rozšírená. Vzhľadom na cieľovú skupinu, pre ktorú je táto aplikácia vyvíjaná, hlavnou požiadavkou bolo intuitívne, jednoducho ovládateľné a jednoznačné používateľské rozhranie.

2.2 Popis funkcionality systému

Systém ako celok slúži ako pomocný nástroj pri procese najmä výskumnej časti reštaurovania historických objektov, špecificky drevených sôch. Používateľ bude schopný zobrazovať a upravovať snímky formátu DICOM cez systém, načítaním konkrétneho súboru projektu z vlastného alebo externého zariadenia pripojeného na svoje osobné

zariadenie. Načítanie súborov nie je viazané na žiadnu databázu, ale obsah úložiska konkrétneho používateľa. Zobrazenie, rovnako aj úprava sa bude vykonávať priamo v aplikácii pre potreby používateľa. Systém umožní používateľovi načítať a zobraziť CT skeny a obrázky formátov .png, .jpg pre účel prezerania a vyhodnocovania obrazu. Systém bude obsahovať funkcie na úpravu obrazu, ako napríklad úprava jasu a kontrastu, rotovanie a približovanie. Systém bude vedieť vytvárať projekciu maximálnej intenzity (MIP) a digitálne rekonštruovaný rádiograf (DRR). Systém bude vedieť vytvoriť cyklus za sebou idúcich skenov, ktoré bude vedieť používateľ exportovať vo forme krátkeho videa rotácie objektu vo formáte .mp4. Takáto funkcionalita neexistuje pri medicínskych softvérových aplikáciach a je kľúčová pre reštaurátorskú dokumntáciu a prezentáciu.

2.3 Architektúra systému

Systém je navrhnutý tak aby bol spustiteľný a funkčný na zariadeniach s operačným systémom Windows 7 a vyšším bez potreby inštalácie. Spustenie aplikácie sa uskutočňuje otvorením súboru s príponou .exe, ktorá je charakteristická pre spustiteľné súbory operačného systému Windows. Všetky projekty zobrazované v aplikácii sa nachádzajú v úložisku používateľa aplikácie, z čoho vyplýva, že aplikácia nespolupracuje so žiadnou databázou.

2.3.1 Štruktúra priečinkov projektu

Štruktúra priečinkov projektu sa priamo odvíja zo štruktúry dát, ktoré reštaurátori majú k dispozícii od rádiológov, s ktorými spolupracujú. Cieľom bolo zjednodušiť im používanie aplikácie a zabrániť zbytočnému zaťaženiu, ktoré by vyplývalo zo zmeny štruktúry projektov.

```
<Názov projektu>/
|
|___ DICOM/                          Hlavný adresár pre projekt
    |___ 00000001/                   Hlavný adresár pre CT snímky
         |___ 00000001/              Adresár jednej série CT snímkov
         |    |___ 00000000          CT snímky konkrétnej časti objektu
         |    |___ 00000001
         |    |___ ...
         |___ 00000002/              Ďalší adresár série CT snímkov
         |    |___ 00000000
         |    |___ 00000001
         |    |___ ...
         |___ ....
```

```
|___ FOTO/                    Adresár pre súbory formátu .png, .jpg
     |___ socha.png           Príklad fotky objektu vo formáte .png
     |___ ...
|___ VIDEO/                   Adresár pre súbory formátu .mp4
     |___ cyneloop5.mp4    Príklad videa objektu vo formáte .mp4
     |___ cynesweep3.mp4
```

2.3.2 Popis priečinkov a súborov

Hlavný priečinok projektu (<Názov projektu>)

Hlavný priečinok obsahuje kompletnú štruktúru projektu:

- Hlavný priečinok pre projekt obsahujúci všetky priečinky projektu
- Hlavný priečinok pre CT snímky
- Priečinky CT snímkov pre jednotlivé časti reštaurovaného objektu
- Samotné CT snímky
- Priečinok pre súbory formátu .png, .jpg. a .mp4
- Fotky objektu
- Exportované videá z aplikácie

Priečinok DICOM/

Ponechaný z pôvodnej štruktúry priečinkov, ktoré využívajú reštaurátori na VŠVU, pre zjednodušenie integrácie.

Priečinok 00000001/

Ponechaný z pôvodnej štruktúry priečinkov, ktoré využívajú reštaurátori na VŠVU, pre zjednodušenie integrácie. Obsahuje všetky priečinky projektu obsahujúce CT snímky aj priečinok FOTO/ obsahujúci súbory formátov .png, .jpg a .mp4.

- **Podpriečinky 00000000/ , 00000001/ a podobne:**

  Ponechané z pôvodnej štruktúry priečinkov, ktoré využívajú reštaurátori na VŠVU, pre zjednodušenie integrácie. Obsahujú všetky CT skeny konkrétnych častí reštaurovaného objektu, prislúchajúce konkrétnemu priečinku.

- **Podpriečinok FOTO/:** Slúži na ukladanie súborov formátov .png, .jpg, ktoré bude používateľ zobraziť v aplikácii za účelom doplnenia informácii o reštaurovanom objekte a prezerania reálneho vzhľadu historického objektu.

- **Podpriečinok VIDEO/:** Slúži na ukladanie súborov formátov .mp4, ktoré používateľ exportuje cez aplikáciu. V aplikácii sa budú dať zobraziť len ako úvodný snímok. Názvy súborov v priečinku sa odvíjajú od názvu zvoleného módu *Cyne Loop* alebo *Cyne Sweep* a čísla priečinku CT skenov, z ktorého bolo video vytvorené.

  Príklad: cyneloop5.mp4, značí názov súboru pre zvolený *Cyne Loop* mód skenov piateho priečinka projektu.

  cynesweep3.mp4, značí názov súboru pre zvolený *Sweep Loop* mód skenov štvrtého priečinka projektu.

2.4 Charakteristika používateľov

Systém je určený pre reštaurátorov, primárne pre reštaurátorov drevenej sochy, ktorý pri svojej reštaurátorskej praxi v oblasti výskumu pracujú s CT snímkami historických objektov. Používatelia budú mať základné technické znalosti v oblasti prezerania a vyhodnocovania CT snímkov. Systém je vhodný aj pre reštaurátorov, ktorých technické poznatky v oblasti úpravy CT snímkov, nie sú na vrcholovej úrovni. Systém ponúka aj tejto skupine používateľov základné a jednoduché manipulovanie s DICOM súbormi.

2.5 Požiadavky na systém

Požiadavky pre systém vyplývajú priamo z návrhu systému a dohody s reštaurátorkami na katedre reštaurovania drevenej sochy na Vysokej Škole Výtvarných Umení v Bratislave.

2.5.1 Špecifikácia požiadaviek

Požiadavky pre systém sú dané nasledovnými bodmi:

1. Používateľ dokáže otvoriť projekt, teda súbor so snímkami objektu v systéme
2. Systém vytvorí priečinok FOTO (ak ho projekt neobsahuje), určený pre ukladanie súborov formátov . png a .jpg
3. Systém dokáže otvoriť nový projekt, aj počas prezerania starého projektu
4. Systém vie zobraziť aplikáciu na celú obrazovku
5. Systém vie načítať a zobraziť súbory formátu DICOM
6. Systém vie načítať a zobraziť súbory formátu .png, .jpg

7. Používateľ vie prepínať medzi priečinkami konkrétneho projektu
8. Systém vie zobraziť náhľady skenov
9. Používateľ sa vie preklikávať medzi jednotlivými skenmi pomocou zvolenia konkrétneho náhľadu
10. Používateľ sa vie preklikávať medzi jednotlivými skenmi pomocou šípok na klávesnici
11. Systém poskytuje informáciu o konkrétnom zobrazovanom skene, zvýraznením konkrétneho náhľadového obrázku
12. Systém umožňuje zobrazenie lišty s nástrojmi na úpravu zobrazovaného skenu
13. Používateľ dokáže upravovať intenzitu kontrastu a jasu DICOM súboru
14. Používateľ dokáže priblížiť a oddialiť zobrazovaný obraz
15. Používateľ vie ovládať jas a kontrast DICOM súboru, podľa konkrétnych hodnôt daných štandardnými zobrazeniami pre pľúca, mozog, brušnú dutinu a kosť
16. Používateľ vie rotovať zobrazovaným obrazom po x-ovej a z-ovej osi
17. Systém dokáže zobraziť sadu za sebou idúcich skenov, čím tvorí ilúziu videa
18. Používateľ vie nastaviť rýchlosť posúvania sa skenov
19. Systém umožňuje zobrazovať skeny v móde looping – teda v cykle od začiatku po koniec, až kým ho používateľ nezastaví
20. Systém umožňuje zobrazovať skeny v móde sweeping – teda v cykle od začiatku po koniec a naopak, až kým ho používateľ nezastaví
21. Používateľ vie spúšťať a zastavovať tento cyklus stlačením tlačidla
22. Systém pri pustenom cykle zobrazuje konkrétne číslo skenu, určené jeho poradím v priečinku
23. Používateľ vie exportovať krátke video cyklu obrazov vo formáte .mp4
24. Systém vie merať vzdialenosť medzi 2 bodmi v milimetroch, určenými klikom používateľa na obrazovku aplikácie
25. Systém vie merať bodovú denzitu v Hounsfieldových jednotkách, v bode určenom klikom používateľa na obrazovku aplikácie
26. Systém vie zobraziť namerané hodnoty hustoty a vzdialenosti na obrazovke

27. Systém dokáže zobraziť skupinu skenov v axiálnom, sagitálnom a koronálnom zobrazení

28. Systém dokáže vytvoriť projekciu maximálnej intenzity (MIP)

29. Systém dokáže vytvoriť digitálne rekonštruovaný rádiograf (DRR) v módoch Anterior, Top, Right

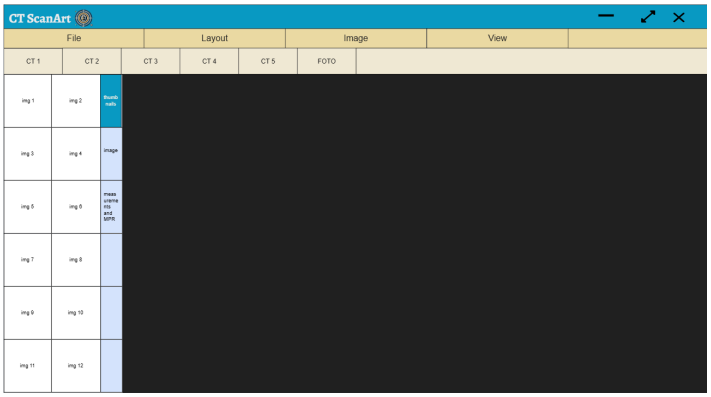
2.6 Používateľské rozhranie

Návrh používateľského rozhrania pred implementáciou sme vytvorili cez online prostredie moqups.com. Konkrétne návrhy jednotlivých častí a niektorých funkcionalít aplikácie sú dostupné na adrese:

`https://app.moqups.com/3LReBCGU1HqIN759hmicJdrKO4Zu6LbM/edit/page/a8549baa5`
`https://app.moqups.com/LDQ0t0eMY36gcvbstwuiAsZYIhMpPAHP/edit/page/ad64222d5`

2.6.1 Hlavné okno

Po spustení programu sa používateľovi zobrazí dialógové okno, cez ktoré si používateľ vyberie projekt, ktorý chce otvoriť. Po načítaní súborov zvoleného projektu sa zobrazí hlavná stránka aplikácie.

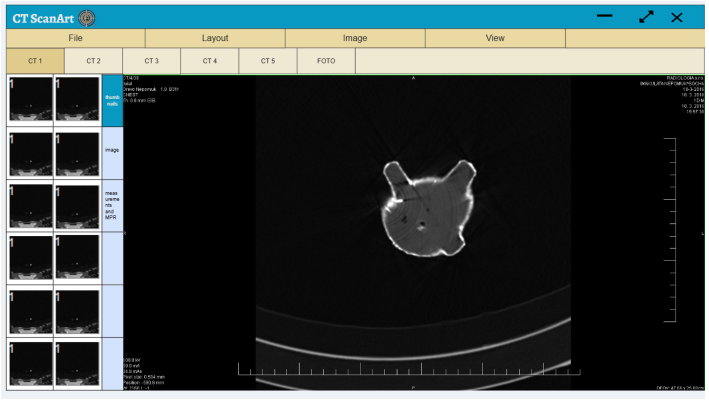


Obr. 2.1: Ukážka hlavného okna aplikácie. Okno sa zobrazí otvorením projektu, po spustení aplikácie.

2.6.2 Zobrazovanie skenu

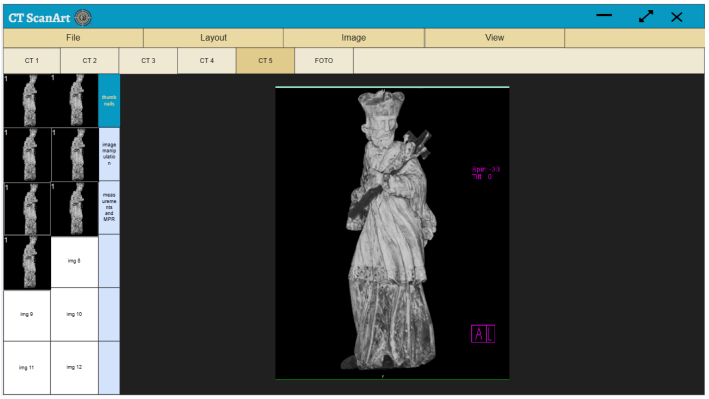
Používateľ stlačením na tlačidlo reprezentujúce konkrétny priečinok CT skenov v projekte zobrazí skeny zvoleného priečinka. Náhľady skenov v priečinku sú zobrazované v bočnej lište. Používateľ sa dokáže preklikávať medzi skenmi v priečinku pomocou šípok na klávesnici, alebo zvolením konkrétneho náhľadu skenu, ktorý sa po stlačení myšou

zobrazí na hlavnej obrazovke aplikácie. Cez lištu s názvami priečinkov sa používateľ vie dostať ku skenom zo všetkých priečinkov projektu.



Obr. 2.2: Ukážka zobrazenia skenu v hlavnom okne aplikácie a ich náhľadov v bočnej lište.

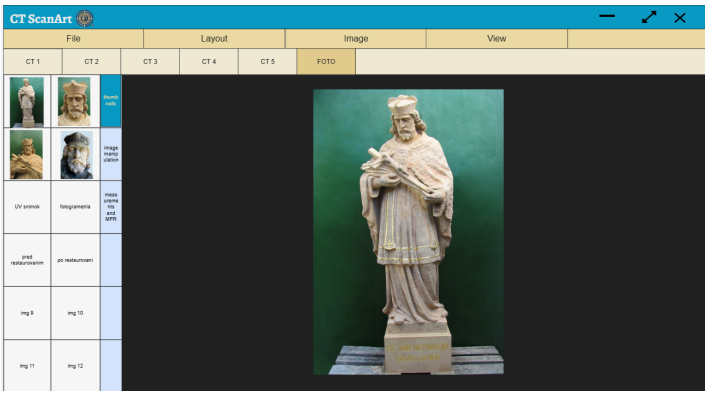
2.6.3 Zobrazovanie skenov 3D projekcie



Obr. 2.3: Ukážka zobrazenia 3D projekcie skenov v hlavnom okne aplikácie a ich náhľadov v bočnej lište.

2.6.4 Zobrazovanie fotografií

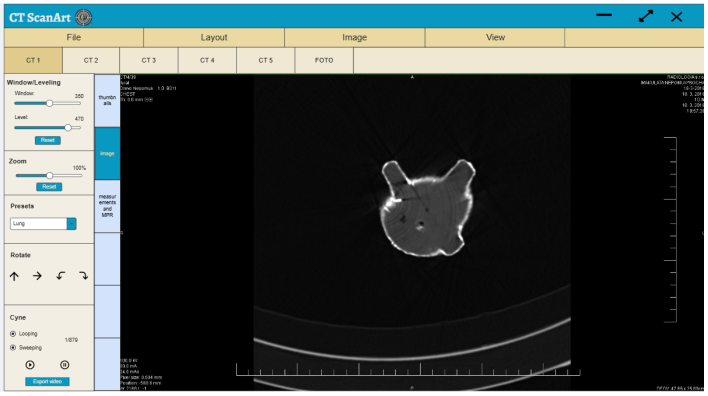
Systém umožňuje zobrazovanie aj formátov .png , .jpg, ktoré sú uložené v priečinku FOTO. Zobrazovanie funguje rovnako ako pri priečinkoch s DICOM súbormi.



Obr. 2.4: Ukážka zobrazovania fotografií v hlavnom okne aplikácie a ich náhľadov v bočnej lište.

2.6.5 Panel nástrojov

V paneli nástrojov používateľ má možnosť modifikovať jas a kontrast obrázka pomocou prvého panelu s názvom Window/Leveling. V druhej časti vie používateľ obraz približovať a odďaľovať. V tretej časti môže používateľ aplikovať konkrétnu hodnotu jasu a kontrastu, tak aby zodpovedala normám sledovanie pľúc, mozgu, kostí a brušnej dutiny. Štvrtý panel slúži na rotovanie obrazu. Posledná časť umožňuje používateľovi zobraziť animáciu Cyne Loop a Cyne Sweep, ktorú dokáže aj exportovať do videa .mp4 formátu.

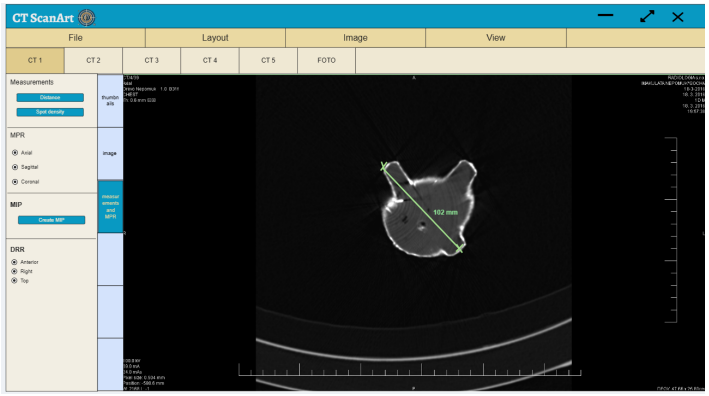


Obr. 2.5: Ukážka hlavného okna aplikácie s panelom nástrojov.

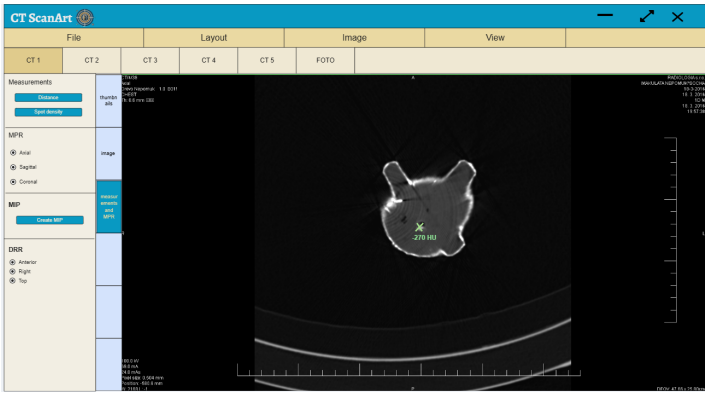
2.6.6 Panel meraní

Pomocou tohto panelu ,vie používateľ realizovať všetky potrebné merania a zobrazovať nové projekcie. V prvej časti *Measurements* sú dve tlačidlá. Prvým tlačidlom *Distance* systém vypočíta vzdialenosť dvoch bodov, ktoré používateľ určí klikom myši na hlavnú obrazovku, respektíve na sken. Rovnako stlačením tlačidla *Spot density*, systém vypočíta hustotu 2D skenu v bode určenom stlačením myši používateľom na hlavnú

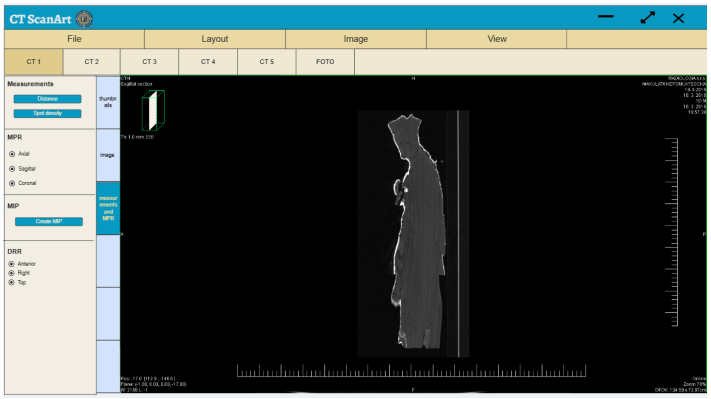
obrazovku v priestore objektu. V druhej časti sú tlačidla pre možnosti zobrazenia multiplanárneho zobrazenia: Axiálne, Sagitálne a Koronálne zobrazenie, ktoré sa vytvorí a zobrazí po stlačení príslušného tlačidla. V *MIP* časti sa nachádza len jedno tlačidlo, ktoré vytvára a zobrazuje projekciu maximálnej intenzity. V poslednej časti *DRR*, sa nachádzajú tri možnosti: *Anterior, Right, Top*. Systém vytvorí a zobrazí digitálne rekonštruovaný rádiograf, s konkrétnymi parametrami pomocou stlačenia príslušného tlačidla.



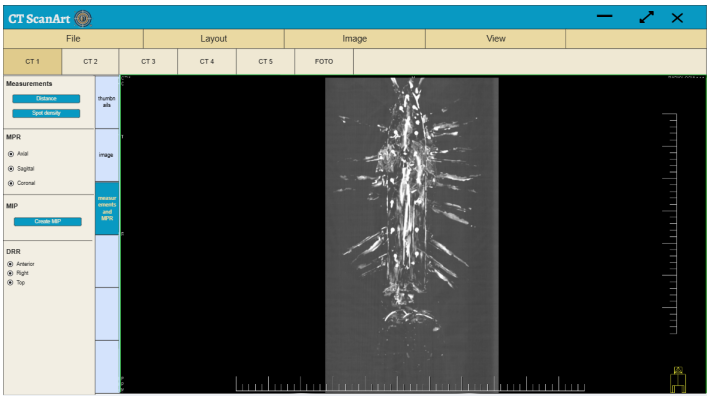
Obr. 2.6: Ukážka hlavného okna aplikácie s panelom meracích nástrojov, v ktorom je vypočítaná vzdialenosť 2 bodov.



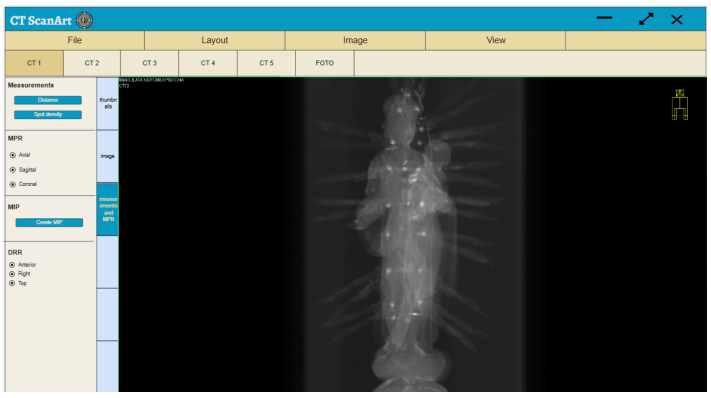
Obr. 2.7: Ukážka hlavného okna aplikácie s panelom meracích nástrojov, v ktorom je vypožítaná bodová hustota.



Obr. 2.8: Ukážka hlavného okna aplikácie s panelom meracích nástrojov, v ktorom je zobrazená sagitálna projekcia drevenej sochy.



Obr. 2.9: Ukážka hlavného okna aplikácie s panelom meracích nástrojov, v ktorom je zobrazená MIP drevenej sochy..



Obr. 2.10: Ukážka hlavného okna aplikácie s panelom meracích nástrojov, v ktorom je zobrazená DRR anterior projekcia drevenej sochy.

2.7 UML diagramy

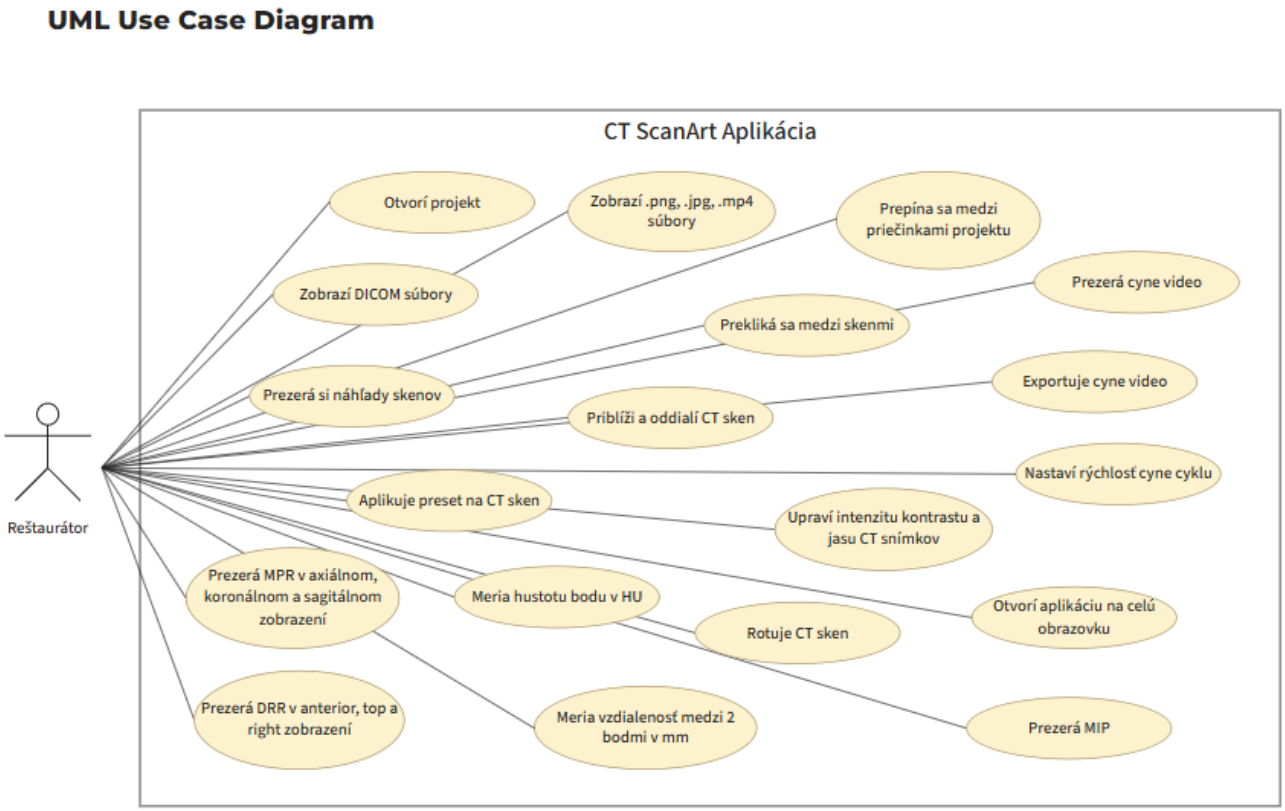
Jazyk UML (Unified Modeling Language) predstavuje štandardizovaný vizuálny nástroj na modelovanie procesov, analýzu, návrh a implementáciu softvérových systémov. Slúži ako spoločný komunikačný prostriedok pre softvérových architektov a vývojárov, ktorý umožňuje popísať, špecifikovať, navrhovať a dokumentovať štruktúru, správanie a procesy softvérových artefaktov – či už ide o nové alebo existujúce systémy. [6]

2.7.1 Use case diagram

Use Case diagram patrí medzi behaviorálne diagramy. Znázorňuje hlavné funkcionality aplikácie z pohľadu používateľa. Používa sa na opis súboru činností (používateľských prípadov). Každý používateľský prípad by mal poskytovať nejaký pozorovateľný a hodnotný výsledok pre subjekty alebo iné zainteresované strany systému. V tomto diagrame sú identifikované základné akcie, ktoré môže vykonávať koncový používateľ, v prípade našej softvérovej aplikácie, reštaurátor. Use Case diagram zároveň pomáha identifikovať hlavné požiadavky používateľa na systém a slúži ako základ pri návrhu používateľského rozhrania, plánovaní testov a validácií funkčnosti aplikácie. [7]

Diagram pokrýva nasledujúce oblasti použitia:

- Správa projektu: Otvorenie DICOM projektu z úložiska používateľa, načítanie CT dát a metadát.
- Zobrazenie a navigácia: Prezeranie DICOM skenov a náhľadov (thumbnails), prepínanie medzi skenmi a obrázkami. Navigácia medzi priečinkami projektu a CT skenmi.
- Manipulácia s obrazom: Nastavenie úrovne jasu a kontrastu (window leveling), priblíženie/oddialenie (zoom), otáčanie, výber prednastavených režimov zobrazenia.
- Merania a rekonštrukcie: Meranie vzdialenosti alebo hustoty, generovanie MPR (Multiplanárna rekonštrukcia), MIP (Projekcia maximálnej intenzity), a DRR (Digitálne rekonštruovaný rádiogram).
- Export a prehrávanie: Animácia obrázkov (tzv. "cyne") a export do videa vo formáte .mp4. Zmena rýchlosti prehrávania cyne animácie.
- Zobrazovanie iných formátov: Zobrazovanie nie len súborov formátu DICOM, ale aj .png, alebo .jpg.
- Navigácia v aplikácii: Zobrazovanie okna aplikácie na celú obrazovku.



Obr. 2.11: UML use case diagram, popisujúci používateľské prípady definujúce aplikáciu. Sú v nich zahrnuté všetky základné požiadavky a funkcionality aplikácie.

2.7.2 Triedny diagram

Triedny UML diagram znázorňuje štruktúru navrhovaného systému na úrovni tried. Zobrazuje hlavné triedy, ktoré tvoria grafické rozhranie, funkcionalitu celého softvéru a spracovanie DICOM obrazových dát. Súčasťou triedneho diagramu, sú vzťahy a obmedzenia medzi jednotlivými triedami. Tento model pomáha pri lepšom pochopení vnútorných väzieb systému a slúži ako základ pre ďalšie rozširovanie a údržbu aplikácie. [8]

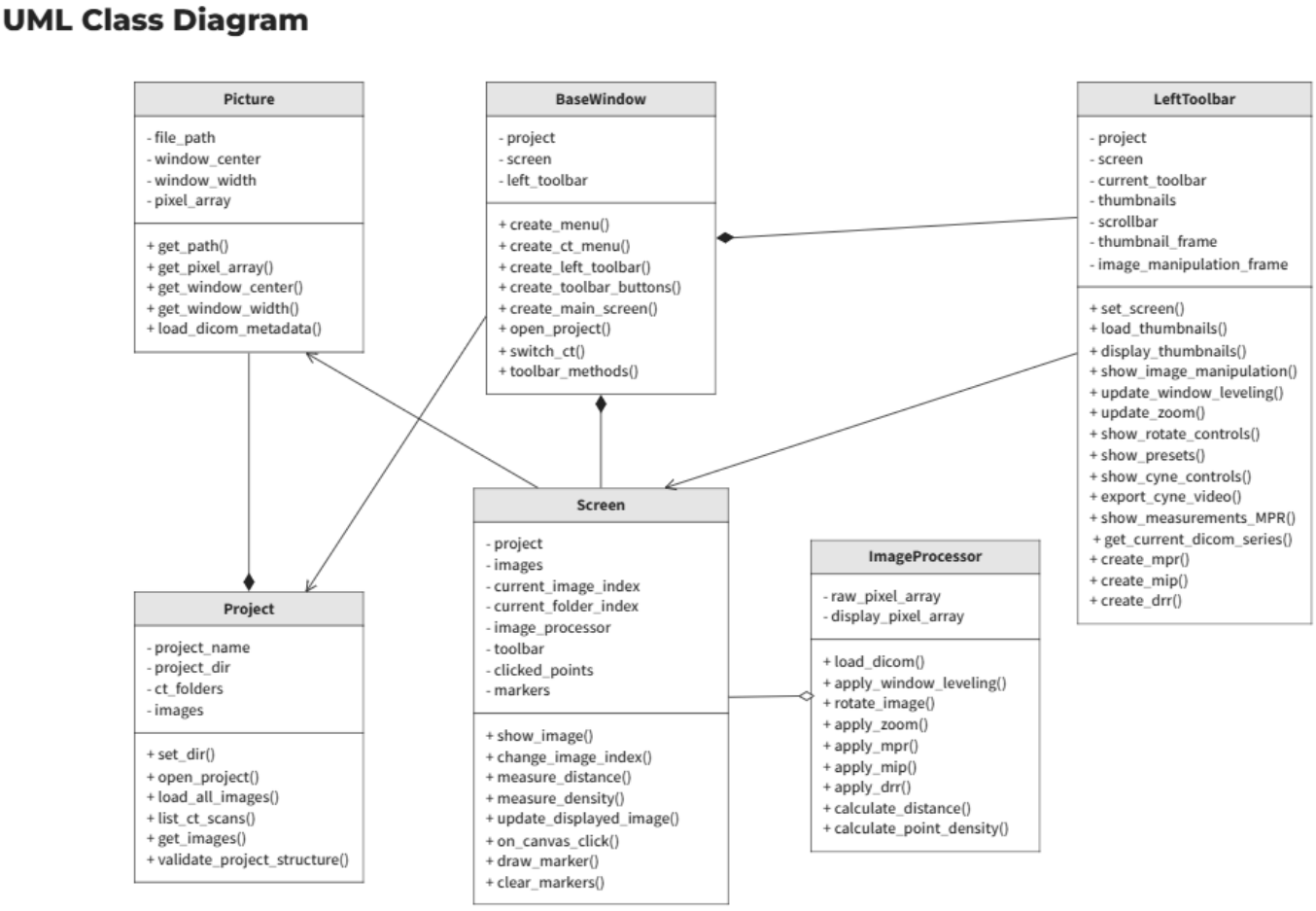
Diagram ukazuje:

- Komponenty používateľského rozhrania (BaseWindow, Screen, LeftToolbar), ktoré zabezpečujú vykreslenie, interakciu a nástroje pre manipuláciu s obrázkami.
- Triedu Project, ktorá reprezentuje otvorený reštaurátorský projekt a spravuje jeho zložky, obrázky a súbory.
- Obrázky typu DICOM, ktoré sú zapuzdrené v triede Picture. Táto trieda uchováva metadáta DICOM skenov.

- Triedu ImageProcessor, ktorá zabezpečuje manipuláciu s obrazovými dátami vrátane nastavovania jasu a kontrastu skenov, tvorby MPR, MIP a DRR. Zabezpečuje aj približovanie a oddiaľovanie skenov, otáčanie a kalkuláciu výpočtov vzdialenosti a hustoty.

Diagram tiež rozlišuje typy vzťahov medzi triedami:

1. Kompozícia (plný diamant) – trieda priamo vytvára a vlastní inštanciu inej triedy. [9]
2. Agregácia (prázdny diamant) – trieda používa inštanciu inej triedy bez silného vlastníctva. [10]
3. Asociácia (šípka) – popisuje voľné prepojenie alebo interakciu medzi objektmi. [11]



Obr. 2.12: UML triedny diagram, zjednodušene popisuje triedy programu a ich interakcie a vzťahy medzi sebou. Sú v nich zahrnuté všetky základné požiadavky a funkcionality aplikácie.

Kapitola 3

Implementácia

Proces implementácie softvérovej aplikácie prebiehal v súlade s návrhom a požiadavkami, ktoré boli vypracované pred začiatkom tejto fázy a spísane do predošlej kapitoly. Slúžili ako základ pre implementáciu jednotlivých komponentov a zabezpečili, že výsledná softvérová aplikácia spĺňa očakávania zadávateľa. Dodržiadanie týchto princípov pomáhalo udržiavať konzistentnosť a kvalitu aplikácie počas celého vývojového procesu.

3.1 Použité technológie

3.1.1 Python

Na programovanie nášho softvérového systému sme zvolili programovací jazyk Python, ktorý nám vďaka svojej prehľadnosti, širokej podpore knižníc a vhodnosti pre rýchly vývoj umožnil efektívne zrealizovať všetky požiadavky. Hlavnou motiváciou pre túto voľbu bola jeho knižnica DICOM, ktorá je určená na prácu s formátom DICOM. Ďalšou veľkou výhodou je jednoduchosť programovacieho jazyka a jeho popularita medzi vývojármi, ktorá bola nápomocná pri hľadaní riešení pre problematické oblasti kódu softvérovej aplikácie.

Grafické používateľské rozhranie

Knižnica, ktorú sme používali na tvorbu grafického používateľského rozhrania (GUI) je pythonovská knižnica tkinter. Túto grafickú knižnicu sme zvolili pre jej jednoduchosť, nakoľko intuitívne a jednoduché grafické rozhranie bolo jednou z hlavných požidaviek pre túto softvérovú aplikáciu. V aplikácii sme využili mnoho modulov, ktoré ponúka knižnica tkinter. Tieto moduly tvorili podstatnú časť používateľského rozhrania. Najvyužívanejšie sú: Scrollbar, Canvas, Scale, Button, Radiobutton, OptionMenu, Label, Entry, filedialog, Menu a Frame. [12]

Práca s obrazom

Hlavnou knižnicou, ktorú sme používali na prácu s DICOM súbormi je pythonovská knižnica pydicom. Na prácu s DICOM obrazovým materiálom bola v aplikácii použitá knižnica pydicom, ktorá umožňuje jednoduché načítavanie, úpravu a extrakciu metadát z DICOM súborov bez potreby proprietárneho softvéru.[13] V našej softvérovej aplikácii sú DICOM súbory spracovávané prostredníctvom triedy Picture, ktorá pri inicializácii načíta daný súbor pomocou pydicom.dcmread().

Distribúcia softvéru

Distribúciu systému reštaurátorom z ateliéru reštaurovania drevenej sochy sme realizovali vytvorením .exe súboru. Na jeho vytvorenie sme použili nástroj PyInstaller. PyInstaller je nástroj pre jazyk Python, ktorý umožňuje zabaliť a distribuovať Python aplikácie vo forme samostatného spustiteľného súboru. Používateľovi umožňuje spustiť zabalenú aplikáciu bez inštalácie prostredia Python alebo akýchkoľvek modulov. PyInstaller podporuje Python 3.8 a novšie verzie a správne balí mnoho hlavných balíkov Pythonu, ako sú numpy, matplotlib a ďalšie. PyInstaller dokáže vytvoriť spustiteľný súbor pre všetky hlavné operačné systémy vrátane Windows, macOS a Linuxu, čo ho robí mimoriadne flexibilným a užitočným pre distribúciu Python aplikácií. [14]

3.2 Backend

Aplikačné jadro, respektíve logickú vrstvu (backend) tvoria tri triedy. *ImageProcessor*, *Project* a trieda *Picture*. Tieto triedy pokrývajú dôležitú časť funkcionality celého softvéru. Súčasťou softvérovej aplikácie bolo niekoľko funkcií, ktoré boli nedostupné v iných aplikáciach. Tieto funkcionality zahŕňali napríklad export *cyne* animácií, alebo zobrazovanie iných fomrátov obrázkov priamo v aplikácii na vizualizáciu súborov formátu DICOM.

Nezanedbateľnou súčasťou všetkých výpočtov a úprav, ktoré boli v tejto vrstve realizované sú DICOM metadáta. Tieto rozsiahle metadáta, poskytujú kontextové informácie o objekte a jeho parametroch a ďalších aspektoch. Tieto metadáta sú kľúčové pre správnu interpretáciu a spracovanie CT obrazov, najmä pri pokročilých technikách, ktoré sme na tejto vrstve realizovali.[15][16]

3.2.1 Trieda ImageProcessor

Trieda *ImageProcessor* je hlavným nástrojom na manipuláciu so súbormi fomrátu DICOM. Umožňuje anipuláciu s metadátami DICOM súborov. Vystupuje ako backendová vrstva medzi obrazovými dátami (DICOM) a GUI komponentami (Screen). Cez ne do-

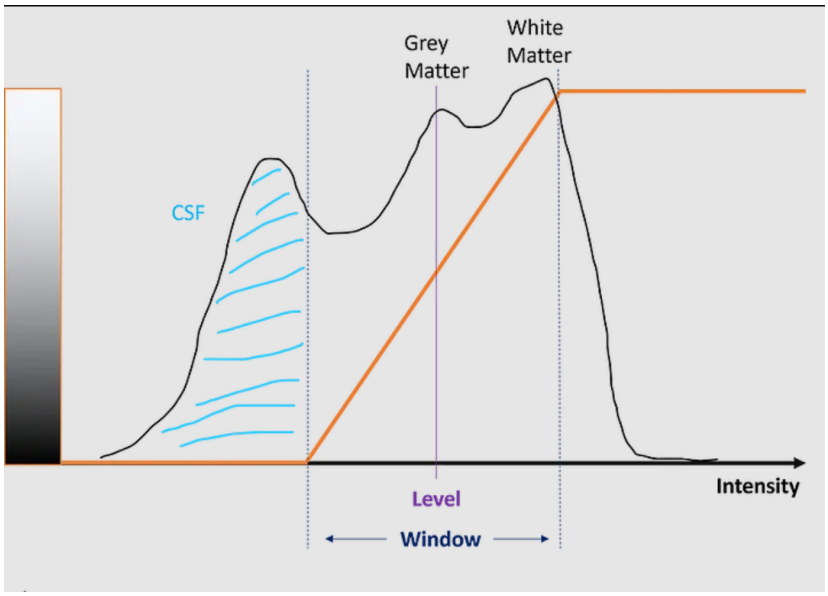
káže meniť napríklad jas a kontrast skenu, vypočítať bodovú hustotu, alebo vzdialenosť medzi dvoma bodmi. Obsahuje metódy na rôzne merania a manipuláciu s obrazom. Umožňuje rotáciu obrazu, jeho priblíženie a oddialenie, zmenu jasu a kontrastu konkrétneho obrazu manuálne, alebo pomocou prednastavených možností. Je hlavným nástrojom na realizáciu merania vzdialenosti dvoch bodov na snímkoch, aj kalkuláciu bodovej denzity . V tejto triede sa okrem meraní tvoria aj nové projekcie: Multiplanárne zobrazenie (MPR), Projekcia maximálnej intenzity (MIP) a Digitálne rekonštruovaný rádiogram (DRR).

Problematika úpravy jasu a kontrastu - Window Leveling

Jednou z hlavných problematík tejto triedy bolo takzvané *Windowing*. CT snímky merajú hodnoty sivej v Hounsfieldových jednotkách, ktoré majú rozsah od -1000 až po 3000, pričom pri niektorých materiáloch môžu byť tieto hodnoty ešte o niečo vyššie. Počítač má pixelovú hĺbku *(pixel depth)* v obmedzenom rozsahu 256, čo znamená, že vie zobraziť len 256 úrovní sivej. Namiesto toho, aby sa všetky hodnoty natlačili do tohto značne menšieho rozsahu sa využíva technika *window leveling*. Táto metóda umožňuje nastaviť strednú hodnotu *(level)* a šírku okna *(window width)*, ktoré určujú rozsah hodnôt, ktoré chceme zobrazovať. Hodnoty nachádzajúce sa mimo tento rozsah sa zobrazujú ako čierne ak sú na spodnej hranici, alebo ako biele ak sú na hornej hranici. Hodnoty, ktoré sa nachádzajú v zvolenom rozsahu sú rozdelené do odtieňov sivej. Táto technika umožňuje zamerať sa na špecifický kontrast tkanív, teda materiálov s rôznou hustotou. [17][16][18][19]

```
1 def apply_window_leveling(self, pixel_array, window_center,
    window_width):
2     ...
3     min_intensity = window_center - (window_width / 2)
4     max_intensity = window_center + (window_width / 2)
5     scaled_array = np.clip(pixel_array, min_intensity,
        max_intensity)
6     scaled_array = (scaled_array - min_intensity) / (
        max_intensity - min_intensity) * 255
7     return scaled_array.astype(np.uint8)
```

Algoritmus 3.1: Ukážka časti Python kódu - funkcia na aplikáciu window levelingu



Obr. 3.1: Uážka princípu window leveling-u. X-ová os zobrazuje pôvodné hodnoty pixelov v Hounnsfieldových jednotkách. Čierna krivka ukazuje histogram rozloženia hodnôt sivej prisluchajúcej rôznym hustotám materiálu. Hodnoty v časti šrafovanej modrou farbou sú nízke hodnoty. Fialová zvislá čiara udáva stred (level) intenzít, ktoré chceme vizualizovať na škále sivej. Rozsah daný modrými čiarami značí šírku okna (widow), ktoré definuje rozsah HU hodnôt, ktoré budú rozložené na škále sivej (0-255). Hodnoty pod oknom sa zobrazia ako čierne a hodnoty nad oknom sa zobrazia ako biele. Ľavý stĺpec zobrazuje zodpovedajúce odtiene sivej, ktoré budú použité na zobrazenie pixelov po aplikácii window levelingu. pozn: CSF je anglická skratka pre mozgomiechový mok.[17]

Meranie vzdialenosti

Ďalšou dôležitou funkcionalitou, ktorá bola súčasťou tejto triedy bola kalkulácia vzdialenosti dvoch bodov. Tento výpočet si vyžaduje nielen získanie presných obrazových súradníc, ale aj prevod na reálne fyzické rozmery. Tento prevod je realizovaný cez metadáta uložené v DICOM hlavičke. Rozlíšenie a skutočná fyzická veľkosť jedného pixelu sú definované v hlavičke súboru cez atribút: *PixelSpacing*, ktorý je reprezentovaný ako pole [dx, dy], kde:

- dx je fyzická šírka jedného pixelu (v mm),
- dy je fyzická výška jedného pixelu (v mm).

Pri práci s 3D objemom pri CT snímkoch sme využívali ak atribút *SpacingBetweenSlices*, ktorý predstavuje vzdialenosť medzi rezmi (v smere z-ovej osi). Pri 2D objeme, bol tento atribút nastavený na hodnotu *None*. Body získané z interakcie používateľa s grafickým plátnom (canvas), reprezentovaným triedou *Screen*, (ktorej sa budeme venovať

v nasledujúcej podkapitole) sú spolu s týmito dvomi atribútmi vstupnými argumentmi pre funkciu pre výpočet vzdielanosti. Na samotný výpočet vzdialenosti medzi dvomi bodmi sme použili euklidovský vzorec, ktorý zohľadňoval atribúty *PixelSpacing* a *SpacingBetweenSlices*. Hlavnú ideu pre algoritmus sme čerpali z pseudokódu: [20] [16]

```
delta = (pointA - pointB) * voxelspacing
distance = sqrt(delta.x^2 + delta.y^2 + delta.z^2);
```

Algoritmus 3.2: Pseudokód pre výpočet vzdialenosti medzi 2 bodmi

Konkrétna implementácia v triede *ImageProcessor* zohľadňovala aj 3D objem:

```
def calculate_distance(self, pointA, pointB, pixel_spacing,
    slice_spacing=None):
    if slice_spacing is None or slice_spacing == 0:
        voxel_spacing = [pixel_spacing[0], pixel_spacing[1]]
    else:
        voxel_spacing = [pixel_spacing[0], pixel_spacing[1],
            slice_spacing]

    delta = np.array(pointA) - np.array(pointB)
    voxel_spacing = np.array(voxel_spacing[: len(delta)])
    scaled_delta = delta * voxel_spacing
    distance = np.sqrt(np.sum(scaled_delta**2))
    return distance
```

Algoritmus 3.3: Ukážka Python kódu - funkcia na výpočet vzdialenosti dvoch bodov

Meranie hustoty

Hustota v kontexte CT snímkov sa vyjadruje ako Hounsfieldova jednotka (HU), ktorá je normalizovanou mierou röntgenovej denzity daného tkaniva v porovnaní s vodou a vzduchom. HU je základom pre analýzu štruktúr na CT snímkach a správna interpretácia je kľúčová pre výskum reštaurátorov.[21] Výpočet denzity v HU z DICOM dát je daný vzťahom :

$$HU = rawvalue * slope + intercept \tag{3.1}$$

Pre výpočet denzity sú potrebné *raw pixel data* (napr. 12-bitové alebo 16-bitové čísla), ktoré nie sú ešte HU hodnotami. Na kalkuláciu denzity, potrebujeme ďalšie dve hodnoty z DICOM hlavičky:

- RescaleSlope
- RescaleIntercept

Tieto údaje sú kľúčové pre výpočet denzity - ich nesprávna interpretácia môže viesť k chybným výsledkom. [22][16] Metóda na výpočet hustoty pracovala s parametrom *ds*, ktorý predstavuje DICOM dataset, načítaný pomocou príkazu pydicom.dcmread(). Tento dataset obsahuje metadáta, medzi nimi aj *RescaleSlope* a *RescaleIntercept*. Raw array so surovými hodnotami pixelov vo forme numpy poľa a súradnice zvoleného pixelu.

```
def calculate_point_density(self, ds, raw_array, x_pixel,
    y_pixel):
    ...
    intensity = raw_array[y_pixel, x_pixel]
    slope = getattr(ds, "RescaleSlope", 1)
    intercept = getattr(ds, "RescaleIntercept", 0)
    hu_value = slope * intensity + intercept
    return hu_value
```

Algoritmus 3.4: Ukážka časti Python kódu - funkcia na výpočet bodovej hustoty

Multiplanárna rekonštrukcia - MPR

Séria CT snímkov obsahuje sadu 2D obrazov (snímok), ktoré dohromady vytvárajú 3D objemový dataset. Tieto snímky sú zoradené pozdĺž jednej osi. V exisujúcich dátach ide o Z-ovú os, ktoré predstavujú sériu axiálnych (priečnych, horizontálnych) rezov. Pre vertikálny rez zboku - sagitálny rez, sa radia rezy pozdĺž X-ovej osi a pre koronálny (frontálny) rez sa využíva Y-ová os, pričom každý z nich je odvodený z pôvodného 3D objemu. [23] [24]

Funkcia na tvorbu MPR vytvorí 3D pole v tvare [rezy, výška, šírka], ktoré reprezentuje objekt v priestore. Z tohto poľa sa potom vyberú správne dáta, ktoré zodpovedajú zvolenému typu projekcie a výsledkom je výrez vo forme 2D poľa. Ten sa upraví a normalizuje na rozsah 0-255, aby sa mohol obraz správne zobraziť. Nakoniec sa obraz zmenší na vhodné rozlíšenie a výsledkom je numpy.array pripravený na vizualizáciu.

```
def apply_mpr(self, dicom_series, projection="Axial"):
    ...
    volume = np.stack(dicom_series, axis=0)

    if projection == "Axial":
        mpr_image = volume[volume.shape[0] // 2, :, :]
    elif projection == "Sagittal":
        mpr_image = volume[:, :, volume.shape[2] // 2]
    elif projection == "Coronal":
        mpr_image = volume[:, volume.shape[1] // 2, :]
```

```
11      else:
12          return None
13
14      mpr_image = (mpr_image - np.min(mpr_image)) / (np.max(
            mpr_image) - np.min(mpr_image)) * 255
15      pil_image = Image.fromarray(mpr_image)
16      width, height = pil_image.size
17      resized = pil_image.resize((int(width * 0.65), int(height *
            0.65)), Image.LANCZOS)
18      return np.array(resized)
```

Algoritmus 3.5: Ukážka časti Python kódu - funkcia na tvorbu MPR zobrazenia

Projekcia maximálnej intenzity - MIP

Projekcia maximálnej intenzity je technika zobrazovania používaná v CT zobrazovaní, ktorá z trojrozmerného objemu generuje dvojrozmerný obraz tak, že pre každú súradnicu XY zvolí pixel s najvyššou Hounsfieldovou hodnotou pozdĺž Z-ovej osi, takže v jednom dvojrozmernom obraze sa pozorujú všetky husté štruktúry v danom objeme. [25] Funkcia po výbere správnych pixelov preškáluje hodnoty do rozsahu 0-255, aby ich bolo možné zobraziť. Nakoniec podobne ako pri MPR, sa prispôsobí konečná veľkosť výsledného obrazu, ktorý sa zobrazí.

```
1 def apply_mip(self, dicom_series):
2     ...
3     volume = np.stack(dicom_series, axis=0)
4     mip_image = np.max(volume, axis=1)
5
6     mip_image = (mip_image - np.min(mip_image)) / (np.max(
          mip_image) - np.min(mip_image)) * 255
7     pil_image = Image.fromarray(mip_image)
8     width, height = pil_image.size
9     resized = pil_image.resize((int(width * 0.65), int(height *
          0.65)), Image.LANCZOS)
10    return np.array(resized)
```

Algoritmus 3.6: Ukážka časti Python kódu - funkcia na tvorbu MIP zobrazenia

Digitálne rekonštruovaný rádiogram - DRR

DRR je syntetický 2D röntgenový obraz, ktorý je počítačovo vygenerovaný z 3D objemu CT dát. Simuluje, ako by vyzerala klasická RTG projekcia, keby sa na objekt pozeralo z

určitého uhla – bez nutnosti reálneho ožiarenia historického objektu. Funkcia umožňuje rýchlu syntézu snímok podobných RTG snímkam z CT dát. Využíva jednoduchý model založený na priemerovaní pixelov v smere zvolenej projekcie.[26] Softvérová aplikácia umožňuje tvorbu DRR v troch rôznych módoch - *anterior* (pohľad spredu), podľa X-ovej osi, top (pohľad zhora) podľa Y-ovej osi a right (pohľad sprava) podľa Z-ovej osi.

```
def apply_drr(self, dicom_series, projection="Anterior"):
    ...
    volume = np.stack(dicom_series, axis=0)

    if projection == "Anterior":
        drr_image = np.sum(volume, axis=0)
    elif projection == "Top":
        drr_image = np.sum(volume, axis=1)
    elif projection == "Right":
        drr_image = np.sum(volume, axis=2)

    drr_image = (drr_image - np.min(drr_image)) / (np.max(
       drr_image) - np.min(drr_image)) * 255
    pil_image = Image.fromarray(drr_image)
    width, height = pil_image.size
    resized = pil_image.resize((int(width * 0.65), int(height *
       0.65)), Image.LANCZOS)
    return np.array(resized)
```

Algoritmus 3.7: Ukážka časti Python kódu - funkcia na tvorbu DRR zobrazenia

3.2.2 Trieda Project

Trieda *Project* slúži na reprezentáciu jednotlivých reštaurátorských projektov, ktoré sú zobrazované v aplikácii. Uchováva informáciu o všetkých skenoch a obrázkoch v projekte. Zaisťuje aby otvorené projekty mali správu štruktúru. V prípade, že projekt neobsahuje priečinky FOTO a VIDEO, program sám tieto priečinky vytvorí na správnom mieste projektu. Ak projekt tieto priečinky obsahuje, systém nevytvára žiadne zmeny. Táto tvorba priečinkov bola realizovaná pomocou pythonovského modulu os, ktorý umožňuje interakciu so súborovým systémom. Tento modul bol využívaný aj na overovanie existencie priečinkov a normalizáciu ciest, čo zabezpečovalo správnu interpretáciu ciest na rôznych operačných systémoch.

3.2.3 Trieda Pictue

Trieda *Picture* reprezentuje každý DICOM sken. Zodpovedá za načítanie záchladných metadát z DICOM hlavičky, ku ktorým pristupuje trieda *ImageProcessor*. Trieda načítava aj raw pixelové pole, ktoré slúži pre ďalšie spracovanie, napríklad pre výpočet denzity, alebo vizualizáciu samotného skenu.

Medzi metadáta, ktoré načítava a uchováva trieda *Picture* patrí *window center* a *window width*, ktoré slúžia na definovanie rozsahu hodnôt pixelov, vizualizovaných na škále hodnôt sivej. Tieto hodnoty sú využívané vo funkcii na aplikáciu *window leveling-u* v triede *ImageProcessor*. Ďalšou dôležitou informáciou sú *pixel data*, ktoré obsahujú surové obrazové dáta a sú ďalej upravované pomocou metadát *rescale slope* a *rescale intercept*. Slúžia na prepočet raw pixelových hodnôt na Hounsfieldove jednotky (HU) - meranie denzity v CT obrazoch. Tieto raw pixelové dáta sú využívané pri kalkulácii vzdialeností dvoch bodov, tvorby MPR, MIP a DRR. Okrem týchto hodnôt sú pre kalkuláciu vzdialenosti dôležité metadáta *Pixel Spacing* a *Spacing Between Slices*. Tie umožňujú prevod pixelovej vzdialenosti na skutočné jednotky (milimetre) a sú nevyhnutné pre presné merania.[27][16]

3.3 Frontend

Používateľské rozhranie a prostredie prie interakciu používateľa so softvérovou aplikáciou (frontend) je tvorené tromi triedami. *BaseWindow*, *LeftToolbar* a trieda *Screen*, ktoré tvoria celé interaktívne rozhranie softvéru. Cez tieto triedy používateľ interaguje so systémom, prezerá si snímky, ovláda manipuláciu obrazu a vytvára nové projekcie a merania. Táto časť tvorí neoddeliteľnú súčasť celeho projektu.

3.3.1 Trieda BaseWindow

Hlavné okno v aplikácii je tvorené triedou *BaseWindow* a je základom pre celú funkcionalitu aplikácie. Do tejto triedy sú vsadené všetky komponenty používateľského rozhrania, medzi ktoré patria *LeftToolbar* a *Screen*. Trieda úzko spolupracuje aj s triedou *Project*, pri načítavaní celého porjektu. S triedou Screen interaguje pri vizualizácii skenov a iných obrazov. Medzi hlavné komponenty tejto triedy patrí horné menu, ktoré ponúka základné operácie ako napríklad otvorenie projektu. Ďalšou dôležitou súčasťou tejto triedy je horná CT lišta, cez ktorú používateľ dokáže navigovať medzi rôznymi priečinkami projektu, ktoré obsahujú skeny iných častí historického diela. Neopomenuteľnou interaktívnou časťou je lišta na prepínanie medzi 3 variantami ľavého panelu nástrojov. Tieto varianty sú dôkladnejšie komentované v popise triedy *LeftToolbar*.

3.3.2 Trieda LeftToolbar

Ľavý panel nástrojov je reprezentovaný triedou *LeftToolbar*. Je to rozšírenie tk.Frame modulu a tvorí bočný panel GUI časti aplikácie. Ten sa zobrazuje v 3 módoch, ktoré plnia rôzne funkcie: zobrazovanie náhľadov snímok (Thumbnails), manipuláciu s DICOM skenmi (Image processor) a výkon meraní a rekonštrukčných projekcií (Measurements and projections). Umožňuje automatizované prezeranie obrázkov vo forme *cyne animácie* a aj jej samotný export do formátu .mp4.

Načítanie náhľadov snímok - thumbnails

Medzi zložitejšie procesy, ktoré sú súčasťou tejto triedy patrí tvorba thumbnails s predspracovaním. DICOM súbory sú pred vytvorením samotného náhľadu normalizované pomocou škálovania pixelov do rozsahu 0-255, čím sa vytvorí vizualizovateľný obraz. V tomto procese sa načíta zoznam pixelových dát, ktoré sú kovertované pomocou normalizácie intenzity (min-max scaling) na uint8. [28][29]

```
def load_thumbnails(self):
    ...
    ds = pydicom.dcmread(image_path)
    pixel_array = ds.pixel_array
    if pixel_array.max() == pixel_array.min():
        pixel_array = np.zeros_like(pixel_array, dtype="uint8")
    else:
        pixel_array = (pixel_array - pixel_array.min()) / (
           pixel_array.max() - pixel_array.min()) * 255
        pixel_array = pixel_array.astype("uint8")
    img = Image.fromarray(pixel_array)
    img.thumbnail((50, 50))
    ...
```

Algoritmus 3.8: Ukážka časti Python kódu - načítavanie náhľadov snímok

Cyne animácia a export videa

Ďalšou zaujímavou funkcionalitou, ktorou sme obohatili softvérovú aplikáciu bola tvorba a export cyne animácie. Samotná tvorba spočívala v prechádzaní sekvenciou skenov v dvoch variantoch: *Looping* a *Sweeping*. Na samotný export videa vo formáte .mp4 sme využili knižnicu OpenCV a triedu cv2.VideoWriter.
Cestu pre uloženie videa sme tvorili podľa zvoleného variantu *sweep* alebo*loop* a čísla adresára, ktorého skeny budú exportované do videa. Po určení cesty pre uloženie videa, sme iniciovali VideoWriter pomocou fourcc (four-character code), ktorý určuje

kompresný algoritmus mp4v pre mp4 video. Nastavili sme fps - počet snímok za sekundu. Iterované snímky načítané ako NumPy pole sa prevedú do OpenCV formátu a zapíšu do videa pomocou príkazu out.write(frame). Ukončenie zápisu do videa sa realizuje cez príkaz out.release(), ktorý zatvorí súbor a dokončí zápis. [30][31][32]

```
def export_cyne_video(self):
        ...
         if self.cyne_mode == "Looping":
             video_path = os.path.join(foto_dir, "cyneloop"+str(
                folder_index+1)+".mp4")
         elif self.cyne_mode == "Sweeping":
             video_path = os.path.join(foto_dir, "cynesweep"+str(
                folder_index+1)+".mp4")
         ...
         first_image_pil = Image.fromarray(first_image)
         frame_width, frame_height = first_image_pil.size
         fourcc = cv2.VideoWriter_fourcc(*'mp4v')
         fps = 1000 // self.cyne_speed
         out = cv2.VideoWriter(video_path, fourcc, fps, (
            frame_width, frame_height))
         ...
             image = self.screen.image_processor.load_dicom(
                image_path)
             if image is not None:
                 image_cv = np.array(image)
                 image_cv = cv2.cvtColor(image_cv, cv2.
                    COLOR_RGB2BGR)
                 image_cv = cv2.resize(image_cv, (frame_width,
                    frame_height))
                 out.write(image_cv)
         out.release()
```

Algoritmus 3.9: Ukážka časti Python kódu - export cyne animácie

Načítavanie DICOM série snímok

Pre tvorbu rekonštrukcií ako napríklad MPR, MIP a DRR je potrbná 3D dátová štruktúra, s ktorou *ImageProcessor* manipuluje podľa konkrétnej potreby. Táto séria je súbor 2D snímok, ktoré sú spojené pomocou np.stack() pozdĺž novej osi, čím vznikne 3D objemová štruktúra. Jej 3 dimenzie sú : počet snímkov, výška a šírka. [**?**]

```
def get_current_dicom_series(self):
         folder_index = self.screen.current_folder_index
```

```
3         ...
4         slice_arrays = []
5         for pic in self.screen.images[folder_index]:
6             ds = pydicom.dcmread(pic.get_path())
7             slice_arrays.append(pic.pixel_array)
8         volume = np.stack(slice_arrays, axis=0)
9         return volume
```

Algoritmus 3.10: Ukážka časti Python kódu - načítavanie DICOM série snímok

3.3.3 Trieda Screen

Trieda *Screen* predstavuje jeden z hlavných grafických komponentov aplikácie. Jej hlavnou úlohou je vizualizácia DICOM snímok a aj obrázkov formátov .png a .jpg. Relizuje interaktívne prepínanie medzi snímkami a významne napomáha pri interaktívnych procesoch ako je meranie vzdialeností a získavanie hodnôt HU denzity. Sprostredkuváva prepojenie medzi vizuálnym GUI a výpočtovými funkcionalitami realizovanými cez triedy *ImageProcessor* a *LeftToolbar*.

Prepočet súradníc

Najzložitejšou časťou tejto triedy bola transformácia obrazových GUI súradníc plochy na originálne pixelové súradnice DICOM obrázku. Táto funkcionalita bola využívaná na identifikáciu bodov pre určovanie vzdialeností a hustoty. Kliknutím na bod na obrazovke sa v aplikácii zaznamenajú jeho súradnice v súradnicovom systéme plátna. Tieto súradnice sú transformované na súradnice v rámci reálne zobrazeného obrázka. Tie boli potom prepočítané na základe odsadenia zobrazovaného obrázku, ktorý bol vopred definovaný.

```
x_rel = x_screen - image_offset_x
y_rel = y_screen - image_offset_y
```

Toto bolo možné pretože bol obrázok zarovnaný na stred a menší ako plátno, na ktorom sa zobrazovalo. Ak sa tieto súradnice nenachádzali mimo rozmerov obrázka, údaje boli prepočítané pomocou škálovania, ktoré vyplývalo z rozmerov zobrazovaného obrázka.

```
scale_x = image_width / displayed_width
scale_y = image_height / displayed_height
x_pixel = int(x_rel * scale_x)
y_pixel = int(y_rel * scale_y)
```

Príklad pre ilustráciu:
Pôvodný obrázok má veľkosť 512 × 512 px, ale zobrazený na plátne je ako 300 × 300

px obrázok. Používateľ klikne na pozíciu na obrazovke: `x_screen = 350, y_screen = 200`. Ak je odsadenie obrázka v plátne: `image_offset_x = 100, image_offset_y = 50`, tak relatívna pozícia bude: `x_rel = 250, y_rel = 150`. Mierka je vypočítaná z originálnej a zobrazovacej veľkosti obrázka: 512 / 300 = 1.71. Takto sa prevedú súradnice na: `x_pixel = 250 * 1.71 = 427 a y_pixel = 150 * 1.71 = 256`.

Literatúra

[1] Dostupné z https://tatramed.sk/tomocon-pacs/.

[2] Dostupné z https://imagej.net/ij/.

[3] Dostupné z https://www.siemens-healthineers.com/digital-health-solutions/syngo-fastview/download-disclaimer.

[4] Dostupné z https://www.osirix-viewer.com/.

[5] Dostupné z https://www.volumegraphics.com/.

[6] Dostupné z https://www.uml-diagrams.org/.

[7] Dostupné z https://www.uml-diagrams.org/use-case-diagrams.html.

[8] Dostupné z https://www.uml-diagrams.org/class-diagrams-overview.html.

[9] Dostupné z https://www.uml-diagrams.org/composition.html?context=class-diagrams.

[10] Dostupné z https://www.uml-diagrams.org/aggregation.html?context=class-diagrams.

[11] Dostupné z https://www.uml-diagrams.org/association.html??context=class-diagrams.

[12] Dostupné z https://docs.python.org/3/library/tkinter.html.

[13] Dostupné z https://pydicom.github.io/pydicom/stable/guides/user/viewing_images.html#introduction.

[14] Dostupné z https://pyinstaller.org/en/stable/.

[15] Dostupné z https://www.intelerad.com/en/2023/02/23/handling-dicom-medical-imaging-data/?utm_source=chatgpt.com.

[16] Dostupné z https://www.dicomlibrary.com/dicom/dicom-tags/.

[17] Dostupné z https://medium.com/@pohaneparikshit7/medical-image-enhancement-window-leveling-and-contrast-adjustment-2a6fe78bf797.

[18] Dostupné z https://radiopaedia.org/articles/windowing-ct.

[19] Dostupné z https://stackoverflow.com/questions/24664359/dicom-image-window-and-level-problems-sometimes.

[20] Dostupné z https://stackoverflow.com/questions/8847632/how-to-measure-the-distance-in-dicom.

[21] Dostupné z https://www.wikiskripta.eu/w/V%C3%BDpo%C4%8Detn%C3%AD_tomografie_a_Hounsfieldovy_jednotky.

[22] Dostupné z https://www.linkedin.com/pulse/generating-hounsfield-units-from-dicom-files-nathan-reilly.

[23] Dostupné z https://www.sciencedirect.com/science/article/abs/pii/S1053077022007972.

[24] Dostupné z https://www.postdicom.com/en/blog/advanced-image-processing.

[25] Dostupné z https://radiopaedia.org/articles/maximum-intensity-projection?lang=us.

[26] Dostupné z https://link.springer.com/article/10.1007/s10278-025-01406-9.

[27] Dostupné z https://medium.com/@ashkanpakzad/reading-editing-dicom-metadata-w-python-8204223a59f6/.

[28] Dostupné z https://www.geeksforgeeks.org/python-pil-image-thumbnail-method/.

[29] Dostupné z https://www.geeksforgeeks.org/python-writing-to-video-with-opencv/.

[30] Dostupné z https://www.geeksforgeeks.org/python-writing-to-video-with-opencv/.

[31] Dostupné z https://opencv.org/blog/reading-and-writing-videos-using-opencv/.

[32] Dostupné z https://stackoverflow.com/questions/30509573/writing-an-mp4-video-using-python-opencv.

[33] Jan Žemlička Daniel Vavřík. *Příspěvek technických věd k záchraně a restaurování památek*. Ústav teoretické a aplikované mechaniky AV ČR, v. v. i., Praha, 2015. Dostupné z `https://www.itam.cas.cz/miranda2/export/sitesavcr/utam/publications/10.21495/49-9/49-9.pdf`.

[34] Jana Dušková Dušana Ondreková. *Využitie počítačovej tomografie v oblasti výskumu umeleckých diel pri reštaurovaní*. Vysoká škola výtvarných umení v Bratislave, Katedra reštaurovania, 2019.

[35] M Hložek and P Krupa. Využití výpočetní tomografie (ct) při przkumu archeologickỳch a historickỳch artefakt. In *Sborník z konference konzervátor a restaurátor Cheb*, pages 36–41, 2006.

[36] FERDA J., Novák M., and KREUZBERG B. Výpočetní tomografie, 1. vyd. praha: Galén, 2002. 663s.

[37] Bc. Juraj Čapelja. Aplikácia počítačovej tomografie v procese reštaurovania historických stolárskych nástrojov. Master's thesis, Technická Univerzita vo Zvolene, Drevárska fakulta, 2013. Dostupné z `https://opac.crzp.sk/?fn=docview2ChildO1QFI1&record=BF2E02CB30820396A8E6366A3158&seo=CRZP-Prehliadanie-pr%C3%A1c`.