

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY



BAKALÁRSKA PRÁCA

Študijný program: informatika

Jozef Dráb

Aplikácia pre automatickú kvantifikáciu dát

Application for automatic data quantification

Vedúci práce: prof. Ing. Vitaly Levashenko, PhD.

Registračné číslo práce: 1316/2020

Ministerské číslo práce: 28360220211231

Žilina, 2021

ABSTRAKT

DRAB JOZEF: *Softvérová aplikácia pre analýzu a kvantifikáciu dát, vyhľadávanie a spracovanie rôznych závislostí* [Bakalárska práca]

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

Vedúci: doc. Ing. Vitaly Levashenko, PhD.

Stupeň odbornej kvalifikácie: Bakalár v odbore Informatika. Žilina.

Fakulta riadenia a informatiky Žilinskej univerzity v Žiline.

Cieľom práce je vytvorenie softvéru pre spracovanie dát schopné analyzovať posun stredovej osi mozgových štruktúr z CT obrazu. Tie sa získavajú z vyšetrení pacientov pomocou CT prístrojov. Aplikácia následne automaticky analyzuje a vyhodnocuje dáta a vyhľadáva rôzne závislosti pre veľkosť posunu za jeho stredovú os. Aplikácia je orientovaná na použitie ako podpora mladým lekárom a praktikantom pri vyhodnocovaní urgentných prípadov.

Kľúčové slová: CT mozgu, Posun stredovej osi mozgu, Model anatomického popisovača, Python

ABSTRACT

DRAB JOZEF: *Software application for data analysis and quantification, searching and processing of various dependencies* [Bachelor thesis]

University of Žilina, Faculty of management Science and Informatics, Department of informatics.

Tutor: doc. Ing. Vitaly Levashenko, PhD.

Qualification level: Bachelor in field Informatics Žilina

FRI ŽU in Žilina.

The aim of this work is to create software for data processing capable of analyzing the shift of the central axis of brain structures from a CT image. These are obtained from examinations of patients using CT devices. The application then automatically analyzes and evaluates the data and looks for various dependencies for amount of displacement its central axis. This application is oriented to be used as a support for young doctors and trainees in the evaluation of emergencies.

Keywords: CT brain, Brain midline shift, Brain Anatomical marker model, Python

ČESTNÉ PREHLÁSENIE

Čestne vyhlasujem, že som prácu vypracoval samostatne s využitím dostupnej literatúry a vlastných vedomostí. Všetky zdroje použité v bakalárskej práci som uviedol v súlade s platnými predpismi.

Súhlasím so zverejnením práce a jej výsledkov.

V Žiline, dňa

.....
Meno Priezvisko

POĎAKOVANIE

Moje poďakovanie patrí prof. Ing. Vitaly Levashenko, PhD. za odborné vedenie, cenné rady a pripomienky pri vypracovaní bakalárskej práce. Ďakujem aj zamestnancovi Ing. Martinovi Kaveckému, spoločnosti Siemens Healthineers za sprístupnenie informácií, aktívnu komunikáciu a pomoc pri nejasnostiach, ktoré sa pri písaní bakalárskej práce vyskytli.

OBSAH

Zoznam obrázkov.....	6
Zoznam tabuliek.....	7
Zoznam skratiek.....	8
Úvod do problematiky	9
1 Súčasný stav riešenej problematiky	11
1.1 Algoritmus využívajúci anatomicke značkovacie modely	11
1.1.1 Časť vyhľadania riadiacich bodov	13
1.1.2 Detekčná časť.....	14
1.2 Algoritmus vyhodnocujúci posun na základe symetrie.....	16
1.3 Algoritmus vyhodnocujúci posun pomocou váženej krivky	17
2 Vývojové prostriedky a tvorba používateľského prostredia.....	18
2.1 Výber vhodného programovacieho jazyka.....	18
2.2 Implementácia GUI - grafického rozhrania	20
2.3 Implementácia objektov	20
3 Popis aplikácie a implementácie.....	22
3.1 Autorotácia	23
3.2 Segmentácia lebečnej časti.....	25
3.3 Vyhľadanie stredu (ťažiska) lebečnej časti	28
3.4 Spracovanie CSF časti.....	30
3.5 Clustering (zhlukovanie)	33
3.6 Detekcia riadiacich bodov A, B	39
3.7 Detekcia riadiacich bodov C, D	41
3.8 Kvantifikácia posunu	43
4 Experimentovanie a vyhodnotenie výsledkov	44
Záver	47
Zoznam použitej literatúry	48
Zoznam príloh.....	51

ZOZNAM OBRÁZKOV

Obr. 1.1 Segmenty CT obrazu mozgu	13
Obr. 1.2 Hľadané riadiace body	14
Obr. 2.1 Grafické okno užívateľa	20
Obr. 2.2 Objekty použité v aplikácii	21
Obr. 3.1 Diagram algoritmu používajúci značkovacie modely	22
Obr. 3.2 Auto-rotácia v diagrame	23
Obr. 3.3 Výsledok autorotácie	24
Obr. 3.4 Fragment kódu autorotácie	25
Obr. 3.5 Výber lebečnej časti v diagrame	26
Obr. 3.6 Výber lebečnej časti	27
Obr. 3.7 Fragment kódu výberu lebečnej časti	28
Obr. 3.8 Vyhľadanie stredu obrazu v diagrame	29
Obr. 3.9 Fragment kódu vyhľadania stredu obrazu	30
Obr. 3.10 Diagram spracovania CSF časti	31
Obr. 3.11 Obraz po spracovaní CSF a hlukovaní	32
Obr. 3.12 Fragment kódu spracovania CSF časti	32
Obr. 3.13 Histogramy intenzít	Chyba! Záložka nie je definovaná.4
Obr. 3.14 Priemerný počet pixelov v intenzitách	35
Obr. 3.15 Zhlukovanie v diagrame	37
Obr. 3.16 Obrazy po metóde zhlukovania	38
Obr. 3.17 Fragment kódu zhlukovania	38
Obr. 3.18 Detekcia bodov A a B v diagrame	40
Obr. 3.19 Výpočet vzdialeností maxím a miním	41
Obr. 3.20 Detekcia bodov C a D v diagrame	42
Obr. 3.21 Porovnanie lekárskeho zakreslenia vs. algoritmického	43
Obr. 4.1 Vzdialenosť posunu z obrazov	46

ZOZNAM TABULIEK

Tab. č. 1 Porovnanie desktopových programovacích jazykov	19
Tab. č. 2 Početnosť intenzít jednotlivých vzoriek	35
Tab. č. 3 Porovnanie výsledkov aplikácie a lekára	45
Tab. č. 4 Porovnanie úspešnosti vyčíslenia	45

ZOZNAM SKRATIEK

MLS	midline shift	posun stredovej osi
CT	computed tomography	počítačová tomografia
IML	ideal midline	ideálna stredová os
DML	deformed midline	deformovaná stredová os
TBI	traumatic brain injury	traumatické poranenie mozgu
ICH	intra-cerebral hemorrhage	vnútro-mozgové krvácanie
CSF	cerebrospinal fluid	mozgovo-miechový mok
ICP	intra-cranial pressure	vnútro-mozgový tlak
SH	subdural hemorrhage	subdurálne krvácanie
3DV	3rd ventricles	vnútro-mozgová komora
FH	frontal horn	čelný roh (komora)
DH		dolná hranica
HH		horná hranica

ÚVOD DO PROBLEMATIKY

Počítačová tomografia (CT) je v súčasnej medicíne nevyhnutná na zobrazenie vnútorností ľudského tela. Pojem „počítačová tomografia“ označuje počítačový röntgenový snímací postup, pri ktorom je úzky lúč röntgenových lúčov zameraný na pacienta, rýchlo sa otáča okolo tela a produkuje prierezové obrázky tela [Ropper1986]. Tieto rezy sa nazývajú tomografické obrázky a obsahujú podrobnejšie informácie, než bežné röntgenové lúče. Len čo počítač stroja zhromaždí množstvo po sebe nasledujúcich rezov, sú digitálne skladané dohromady a vytvárajú tak trojrozmerný obraz pacienta, ktorý umožňuje ľahšiu identifikáciu a lokalizáciu základných štruktúr, ako aj možných nádorov alebo abnormalít [Tromberg2019].

Čoraz dôležitejšie je v zdravotníctve digitálne spracovanie obrazu, vďaka využívaniu systémov priameho digitálneho zobrazovania. Okrem pôvodne digitálnych metód, ako je napríklad počítačová tomografia (CT) alebo magnetická rezonancia (MRI), spočiatku analógové zobrazovacie modalitty, ako endoskopia alebo rádiografia sú dnes vybavené digitálnymi senzormi. Digitálne obrázky sa skladajú z jednotlivých pixelov, ktorým sú priradené samostatné hodnoty jasu alebo farieb [Sullivan2008]. Sú efektívne spracované, objektívne vyhodnotené a sprístupnené na mnohých miestach súčasne, vhodnými prostriedkami. Na základe digitálnych zobrazovacích techník je vo svete použiteľné celé spektrum spracovania medicínskych digitálnych obrazov [Harris2009]. Bežne používaný pojem „spracovanie biomedicínskeho obrazu“ znamená ustanovenie digitálneho spracovania obrazu pre biomedicínske vedy. Všeobecne digitálne obrazové spracovanie pokrýva hlavné oblasti, ako tvorba obrazu, ktorá zahŕňa všetky kroky od zachytenia obrázku až po formovanie matice digitálneho obrazu. To si vyžaduje znalosti o povahe a obsahu obrazu, ktorý musí byť integrovaný do algoritmov na vysokej úrovni abstrakcie. Proces analýzy obrazu je teda veľmi špecifický a vyvinuté algoritmy je možné zriedka prenášať priamo do inej aplikácie [Boone2007].

Na rozdiel od analýzy obrazu, ktorá sa často označuje aj ako analýza na vysokej úrovni, spracovanie na nízkej úrovni označuje manuálne, alebo automatické techniky, ktoré je možné realizovať bez znalostí o konkrétnom obsahu obrázkov. Tento typ algoritmov má podobné účinky bez ohľadu na obsah obrázkov. Napríklad natiahnutie histogramu röntgenového snímania zvyšuje kontrast ako na akejkol'vek dovolenkovej fotografii. [Deserno2011]

Vyšetrenie mozgu pomocou CT, je nevyhnutné na hodnotenie mozgových abnormalít, najmä pri akútnych poraneniach, ako je traumatické poranenie mozgu (TBI). To je definované ako akékoľvek štrukturálne poranenie lebky so zmenami mozgovej fyziológie v dôsledku externej sily [Uribe2013]. Ročný výskyt TBI v Spojených štátoch je 200 prípadov na 100 000 ľudí [Segun2021]. V krajinách s

nízkym a stredným príjmom je percento postihnutých chorobami TBI pomerne vysoké a málo dostupných údajov neidentifikuje rozsah problému [Byass2010].

Rýchla a presná interpretácia obrazov CT mozgu je teda prvým kľúčovým krokom k dosiahnutiu lepšej neurologickej prognóze [Silver2005].

Ľudská hlava je zhruba obojstranne symetrická. Aj keď existujú funkčné rozdiely medzi hemisférami mozgu, hrubá morfológia sa riadi pravidlom, že mozog aj mozoček je symetrický s lalokmi, komorami a jadrá sú podobnej veľkosti a tvaru v oboch hemisférach. Jemná štrukturálna asymetria nehrá v klinickej diagnostickej rádiológii žiadnu úlohu [Qiu2006]. Z patologických vyšetrení lekári vedia, že intrakraniálne krvácanie v mozgu (ICH) môže spôsobiť posun mozgových štruktúr, po ktorom nasleduje kompresia mozgového kmeňa a následne smrť. Preto sa spoliehajú na to, že posun štruktúr stredovej osi napomáha diagnostike od samého počiatku zobrazovania mozgu pomocou CT obrazov. [Liao2006].

Aj keď vizuálna kontrola a manuálne meranie posunu mozgových štruktúr (MLS) sa už stalo štandardom a rutinná úloha pre skúsených lekárov (najmä odborníci, ktorí sa často stretávajú s interpretáciou obrazov mozgu vrátane neurorádiológov, neurochirurgov a neurológov), presné meranie a interpretácie CT mozgu môže byť pre ostatných stále náročné [Quattrocchi1991]. Lekári na pohotovosti sú zvyčajne prví, ktorí sa podieľajú na vyšetrení pacientov s traumatickým poranením mozgu (TBI) a spontánnym vnútro-mozgovým krvácaním (ICH), ale spoľahlivosť ich vyšetrení mozgu je často spochybňovaná [Rincon2019].

Vďaka pokroku vo výpočtovej technike veríme, že pomocou počítača môže mať diagnostika potenciál na zlepšenie presnosti a rýchlosti spracovania interpretácie CT mozgu. Je ale ťažké formulovať lekárske znalosti tak, aby ich bolo možné priamo a ľahko integrovať do automatických algoritmov spracovania obrazu. [Wang2017]. V literatúre sa to označuje ako sémantická medzera, čo znamená rozpor medzi kognitívnou interpretáciou diagnostického obrazu lekára (vysoká úroveň) a jednoduchá štruktúra diskretných pixelov, ktorá je používaná v počítačových programoch na predstavenie obrazu (nízka úroveň). [Liu2006]

1. SÚČASNÝ STAV RIEŠENEJ PROBLEMATIKY

Vďaka moderným programovacím jazykom, umožňujúcim spracovanie obrazu, nastal rozvin aplikácií pre automatickú detekciu a kvantifikáciu posunu mozgových štruktúr. Väčšina z nich sú priamo používané v nemocniciach, implementované súkromnými vývojárskymi spoločnosťami. K týmto aplikáciám sa dá priblížiť vďaka verejným publikáciám. Pristúpiť sa dá k spôsobom riešenia, výsledkom danej aplikácie ako aj k štatistickému spracovaniu. Aplikácie sa líšia robustnosťou algoritmov, ako aj prístupom ku vstupným údajom, čo je v podstate obraz CT mozgu [Kazdal2015].

Najrozsiahlejšie programy zahrňujú široké spektrum informácií o pacientovi, jeho zdravotnom stave a diagnózach. Sú plne zautomatizované, čo znamená, že program si z CT prístroja sám vyberá obrazy. Na základe implementácie vyberie vhodný obraz, z ktorého je schopný vyčíslieť posun mozgových štruktúr a podľa veľkosti posunu vyhodnotiť, v akom kritickom stave sa pacient nachádza. Takéto aplikácie vytvárajú vývojárske spoločnosti, ktoré úzko spolupracujú s nemocnicami, ktoré im poskytujú dáta pacientov, pomocou ktorých je aplikácia vyvíjaná a následne testovaná [Gong2007]. V nasledujúcej podkapitole sa budeme podrobnejšie zaoberať tromi typmi riešení. Sú to algoritmy používajúce anatomické značkovacie modely (podkapitola 1.1), algoritmus vyhodnocujúci posun na základe symetrie (podkapitola 1.2) a algoritmus vyhodnocujúci posun pomocou váženej krivky (podkapitola 1.3).

1.1 ALGORITMUS POUŽÍVAJÚCI ANATOMICKÉ ZNAČKOVACIE MODEL Y

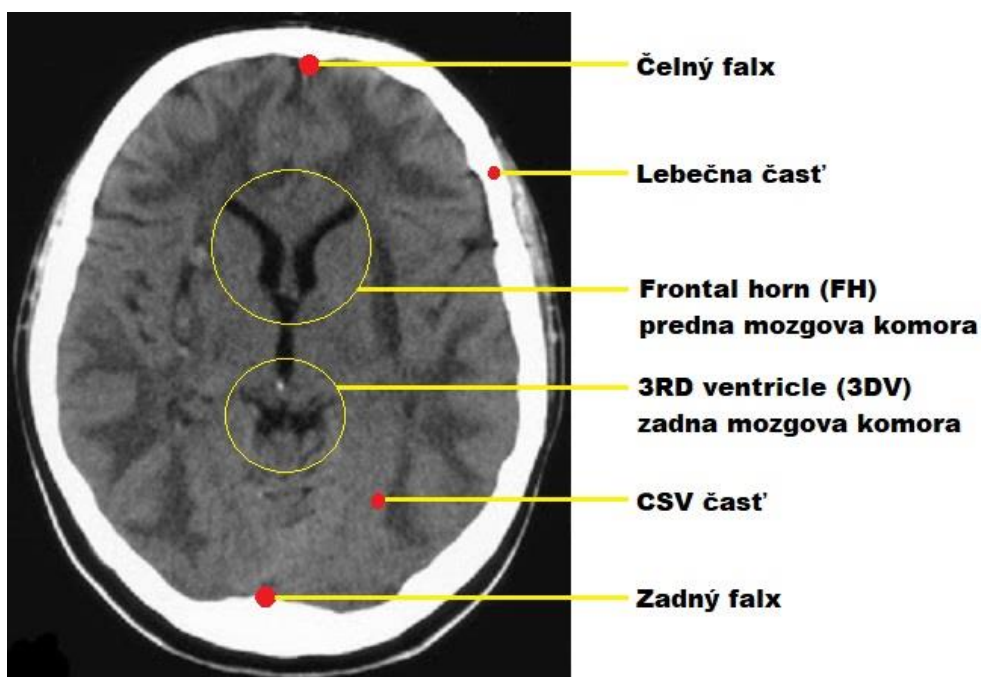
Algoritmus využívajúci značkovacie modely pristupuje k obrazu tak, že sú oddelené jednotlivé časti obrazu do segmentov. Dané časti obrazu sú na základe vyhodnocovania a výpočtov označené ako anatomické značky. Pomocou nazbieraných anatomických značiek, ich vzájomných vzťahov a symetrických vlastností sú vo fáze kvantifikácie určené ako riadiace body. Tie majú za následok určiť tvar deformovanej krivky (DML).

Pomocou týchto bodov a ich vzájomných vzťahov je v procese kvantifikácie vedená ideálna a deformovaná krivka. Na základe ich vzdialeností a pozícií je vyčíslený posun stredovej osi a pomer oboch mozgových hemisfér (polovice mozgu). Experimentálne výsledky ukazujú, že navrhnutá práca efektívne vyhľadáva riadiace body a kvantifikuje na základe ich vzťahov, posun mozgových štruktúr. V porovnaní so súčasnými metódami táto metóda významne znižuje chybu kvantifikácie, zväčšuje robustnosť aplikácie, ktorá úspešne kvantifikuje posun aj pri veľkej rôznosti vstupných obrazov. Veľkou výhodou tejto metódy je budovanie nadstavieb, kde pridaným ďalších anatomických značiek sa zvýši pravdepodobnosť úspešnej detekcie a kvantifikácie posunu mozgových štruktúr [Ruizhe2014].

Uviedli sme si, že algoritmus využíva anatomické značkovacie modely. Vstup aplikácie predstavuje manuálne zadanie obrazu CT mozgu, ktorý najviac spĺňa predpoklady pre zobrazenie zadnej mozgovej komory (3DV) a čelnej mozgovej komory (FH) (obr. 1.1). Pri normálnom vyšetrení CT prístroj vytvorí približne 20 obrazov od nosovej dutiny až po vrch lebky. Obraz má tieto vlastnosti: ohraničenie plochy lebečnou časťou a výbežky falxov (obr. 1.1). Z obrazu sa vyseparujú také časti, ktoré sú z hľadiska detekcie kľúčové. Patrí medzi nich lebečná časť a vnútro-mozgová časť (CSF časť)(obr. 1.1).

Obe časti sú z hľadiska skúmania dôležité, pre nájdenie riadiacich bodov. Tie určujú tvar deformovanej krivky (DML). Výsledkom procesu detekcie má byť ideálna úsečka (IML), predeľujúca vnútro-mozgovú časť na dve symetrické polovice a deformovaná krivka (DML). Ideálna úsečka je daná počiatočným a koncovým bodom, v našom prípade čelný a zadný falx (obr. 1.1). Ideálna úsečka má slúžiť ako indikátor korektnosti, resp. v akej pozícii by sa mali mozgové štruktúry nachádzať. Znamená to, že mozog zdravého jedinca bez akejkoľvek deformácie má IML v rovnakej polohe, ako DML. Tá má úchyt počiatočného a koncového bodu rovnako, ako IML, v hornom a dolnom falxe.

Rozdiel medzi ideálnou a deformovanou krivkou je ten, že deformovaná krivka je aproximovaná riadiacimi bodmi C a D, určené pomocou detekčného algoritmu vo vnútro-mozgovej časti. Ten vyberie do zoznamu kandidátov viaceré body, ktoré spĺňajú predpoklad, žeby sa mohlo jednať o zadnú mozgovú komoru (3DV) a čelnú mozgovú komoru (FH). Na základe porovnávania týchto kandidátov, ich veľkostí, vzdialeností od ideálnej úsečky (IML) a symetrie sú zo zoznamu kandidátov vybraní najvhodnejší kandidáti, označení ako riadiace body C a D. Pomocou algoritmu kvantifikácie a vyhodnotenia výsledkov sú výstupom údaje, ktoré reprezentujú maximálnu vzdialenosť ideálnej úsečky (IML) od deformovanej krivky (DML) a pomery plôch v ideálnom a deformovanom prípade. Tento algoritmus obsahuje 2 časti: časť vyhľadania riadiacich bodov (podkapitola 1.1.1) a detekčná časť (podkapitola 1.1.2)

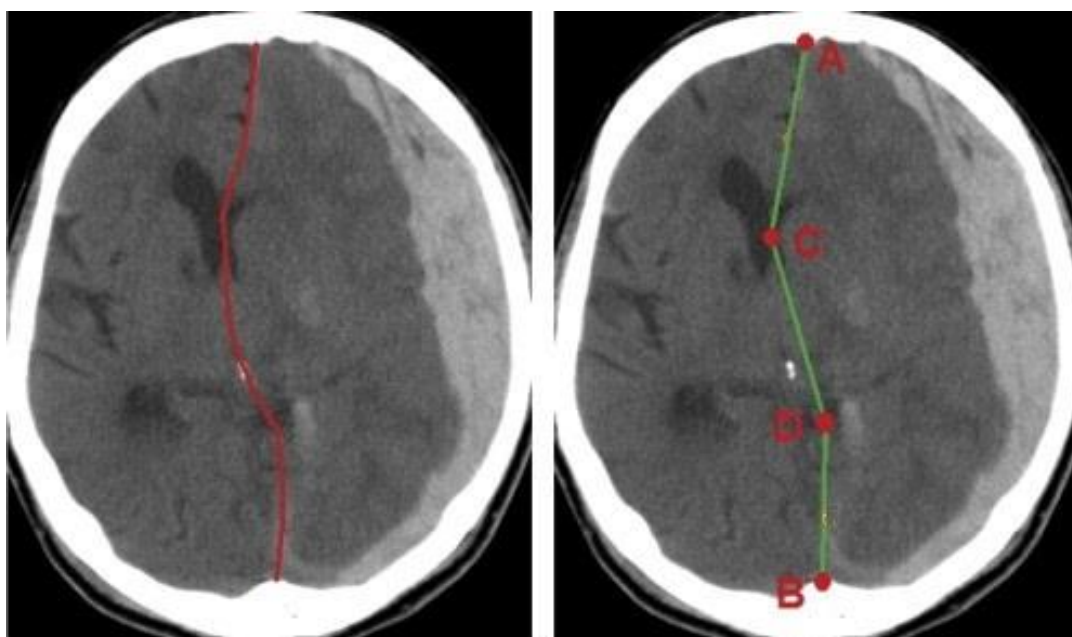


Obr. 1.1, Segmenty CT obrazu mozgu

1.1.1 Časť vyhľadania riadiacich bodov

Algoritmus je prispôsobený k tomu, aby bol výpočtový čas čo najmenší a dokázal vyhľadať riadiace body aj v nesymetrickom prípade. V prípade nízkej kvality obrazu, alebo neschopnosti nájsť súčasne zadnú mozgovú komoru (3DV) a čelnú mozgovú komoru (FH), sa aplikácia ukončí bez výsledku. Ako je uvedené vyššie, deformovaná krivka (DML)(obr. 1.2) prechádza riadiacimi bodmi C a D. Pevne stanovené sú tie, v ktorých DML začína a končí. Sú to úchyty čelného a zadného falxu, označme si ich ako body A a B. Tak ako ideálna úsečka (IML) je daná dvomi riadiacimi bodmi, DML je formovaná posunom 3DV a FH, teda nám medzi riadiace body pribudnú ďalšie dva body. Označme si ich ako body C a D. Tieto body sú kľúčové pre modelovanie DML.

Určili sme si množinu riadiacich bodov A, B, C, D (obr. 1.2 (b))



a)

b)

Obr. 1.2, Hľadané riadiace body : (a) DML vytvorená lekárom (b) DML vedená riadiacimi bodmi

1.1.2 Detekčná časť

Na základe princípu riadiacich bodov posunu stredovej osi postupujeme takto: Z každého obrazu CT (rezu) sa v zadanom intervale vyčíslí posun. Algoritmus detekcie riadiacich bodov nájde vhodných kandidátov pre daný bod posunu stredovej osi v strede CSF časti. Potom sa z množiny najlepších kandidátov vyberú na základe informácií o vnútro-mozgovej časti takí kandidáti, ktorí sú vhodní pre sadu riadiacich bodov. Na záver sa deformovaná krivka (DML) aproximuje na základe týchto riadiacich bodov a kvantifikuje rozdiel medzi ideálnou úsečkou (IML) a deformovanou (DML).

Spracovanie spočíva v segmentácii zvoleného CT obrazu do dvoch úrovní. Segmentujú sa vnútro-mozgové štruktúry a lebka, na základe intenzít pixelov. Avšak je ťažké segmentovať oblasti na základe konštantných hodnôt, vzhľadom k rôznosti obrazov vyšetrených pacientov. Zhlukovaním dokážeme segmentovať pixely rovnakej intenzity do skupín a tým ich odlišiť od ostatných zhlukov. Deformovaná krivka (DML) je určená riadiacimi bodmi, pričom každý riadiaci bod má predom určenú oblasť, kde sa môže nachádzať. Je ňou lebka (body A a B) a komory (body C a D). Princíp spočíva v dvoch detektoroch, kde každý detektor prehľadáva iný segment vstupného obrazu CT.

Riadiace body reprezentujúce falxy (body A a B) sú dva body umiestnené vo výbežkoch v dolnej a hornej časti lebky. K ich nájdeniu algoritmus hľadá najmenšiu vzdialenosť od stredu lebky a okraja lebky medzi dvoma lokálnymi maximálnymi vzdialenosťami z ľavej a pravej strany. Pre ich nájdenie

je potrebná správna rotácia lebky do vertikálnej polohy. Aby sme správne určili lokálne maximum, algoritmus prechádza z centrálnej vertikálnej osi smerom doľava a doprava, kde kontroluje vzdialenosť lebky od centrálného bodu lebky (ťažisko obrazu). Presný popis algoritmu sa nachádza v ďalšej sekcii implementácie aplikácie.

Detekcia komôr (bodov C a D) pracuje pomocou metódy zhukovania, kde je celý vnútorný obsah lebky podrobený hľadaniu zhukov s najtmavšou intenzitou. Každý zhuk je považovaný za kandidáta. Pre určenie, ktorý kandidát je najvhodnejší, je pripravený rozhodovací strom, ktorý na základe vzdialeností zhukov od ideálnej osi (IML), symetrie, veľkosti zhukov a vzdialeností medzi nimi, určí najvhodnejšieho kandidáta. Takýmto spôsobom sú definitívne určené riadiace body C a D.

Predpokladá sa, že každý klaster (zhuk) ktorý hľadáme ako kandidáta pre riadiace body C a D, obsahuje pixely určitej intenzity v intervale $\langle a, b \rangle$. Obraz transformovaný do intenzity šedej sa zmení z 3-channel do 1-channel obrazu. Interval farieb je teda $\langle 0, 255 \rangle$. Keďže hľadáme zhuky tmavej intenzity, dolnú hranicu intervalu určíme ako najmenšiu hodnotu (intenzitu) všetkých pixelov nachádzajúcich sa vnútri lebky. O to náročnejšie je určiť hornú hranicu intervalu, teda takú intenzitu pixelov, ktorá bude najvyššia prípustná pri detekcii zhukov. Algoritmus to nemôže odhadovať konštantne, vzhľadom k rôznosti obrazov. Konkrétny prístup k odhadu hornej hranice je bližšie popísaný v kapitole implementácie. Takto je vyrátaný interval $\langle a, b \rangle$. [Ruizhe2014].

Možnosťou pre zefektívnenie detekcie by bolo hľadanie kandidátov pre riadiace body E a F. Tie sú dva koncové body mozgových falxov. Bod E je dolný roh (lokálne minimum) čelného falxu, bod F je horný roh (lokálne maximum) zadného falxu. Cípy falxov sú reprezentované ako dva priame segmenty, umiestnené okolo hornej centrálnej a dolnej centrálnej oblasti mozgu. Problémom je detekcia cípov z CT snímok. Vzhľadom k tomu, že cípy sa vždy posúvajú s deformovanou osou mozgu, je ľahšie odhadnuteľná deformovaná krivka (DML). Problém vyplýva z nejasného výskytu cípov v CT obrazoch. Ako je znázornené na obr. 1.2, cípy sú ťažko viditeľné. Hranica falxu nie je zreteľne viditeľná a koncové body cípov sa v mozgovej hmote rozmazávajú a zmenšujú. Úspešná metóda detekcie cípov by mala byť schopná nájsť falx aj v nízkom kontraste a vysokom šume obrazu, ako aj v CT obrazoch. Metóda detekcie by mala byť časovo nenáročná a robustná aj v prípade závažnej patologickej deformácie. V súčasnosti bohužiaľ neexistuje práca, na automatickú detekciu falxov a súčasne cípov mozgu z CT obrazov. Tradičná metóda extrakcie je napríklad Houghova metóda transformácie a vektorizácie [Ramachandran1980]. Tá si však nevyžaduje prepojenie s falxom v porovnaní s metódou vektorizácie. Avšak pri výpočte a detekcii predného a zadného falxu má

Houghová metóda vysoký výpočtový výkon a časovú náročnosť [Duda1972]. Vektorizácia je ďalšia široko používaná metóda na detekciu. Jedným dobre známym algoritmom vektorizácie je jednosmerný reťazec (DSCC). [Zheng2002]. Avšak v tejto práci sa ňou nebudeme zaoberať.

Účelom výberu viacerých kandidátov a výberu najlepšieho vhodného kandidáta je zlepšenie robustnosti algoritmu a zvýšenie funkcionality aj pri obrazoch s nízkou pravdepodobnosťou úspechu.

V predchádzajúcich krokoch sme získali riadiace body A a B a kandidátov pre riadiace body C a D. Upozornili sme, že z dôvodu chýbajúcej zadnej mozgovej komory (3DV) a čelnej mozgovej komory (FH) z nejasného obrazu, môžu chýbať riadiace body C a D. V prípade chýbajúcich bodov C a D je možnosťou si určiť priestorové vzťahy medzi kandidátmi, ktoré charakterizujú geometrické vlastnosti značiek. Vzhľadom k tomu, že krvácanie ovplyvňuje priestorové vzťahy a pozície mozgových štruktúr. Krvácanie je dôležitým faktorom. Napríklad veľké krvácanie v pravej strane mozgu deformuje a posúva komory vľavo, takže sa očakáva, že riadiace body C a D budú v ľavej polovici mozgu. V tejto práci ale nemodelujem vplyv krvácania na posun stredovej osi mozgu, keďže nie každý obraz CT obsahuje subdurálne krvácanie mozgu (SH).

1.2 ALGORITMUS VYHODNOCUJÚCI POSUN NA ZÁKLADE SYMETRIE

V algoritmoch založených na vyhodnocovaní symetrie sa vyhľadáva deformovaná stredová os (DML) a jej vzdialenosť od ideálnej stredovej osi (IML), ktorá sa vyhodnocuje ako posun mozgových štruktúr (MLS). Rozdiel oproti iným metódam je v detekcii deformovanej osi. Aplikácia fungujúca na princípe vyhodnocovania symetrie funguje tak, že rozdeľuje deformovanú krivku (DML) na tri segmenty: horný a dolný priamy segment predstavujúci časti falxov (výbežkov vo vnútornej stene lebky (obr. 1.2), umiestnené vpredu a vzadu), ktorý oddeľuje dve mozgové hemisféry a centrálny zakrivený segment reprezentovaný takzvanou „Bézierovou“ krivkou, ktorá znázorňuje mäkké mozgové tkanivo. Autori predpokladajú, že deformovaná krivka má najideálnejšiu bilaterálnu symetriu, vypočítanú minimalizáciou sčítaných štvorcových rozdielov medzi všetkými pixelmi stredovej osi v horizontálnej rovine (medzi ľavou a pravou hemisférou). Pre zjednodušenie riešenia je horný a dolný falx považovaný za statický pri opakovanom spúšťaní výpočtu. Ten sa opakuje 3x s maximálnymi povolenými parametrami, ktoré určujú vzdialenosť symetrických pixelov v milimetroch. Ak sú výsledky stabilné, posun je pohotovo určený na základe centrálného segmentu. Inak sa považuje vyhodnotenie za neúspešné. [Chun-Chin2018]

1.3 ALGORITMUS VYHODNOCUJÚCI POSUN POMOCOU VÁŽENEJ KRIVKY

Ide o metódu vypočítania posunu stredovej osi na základe pomeru posunu (MLS) oproti maximálnej šírke intrakraniálnej (vnútro-mozgovej) oblasti. Vážená stredová os sa vykresľuje na základe jednotlivých intenzít pixelov s vyššou váhou vzhľadom na tmavšie časti obrazu. Posun (MLS) sa potom meria ako vzdialenosť medzi váženou stredovou osou a ideálnou stredovou osou (IML). Priemerný čas spracovania, od spustenia až po automatický výstup údajov trval 10 sekúnd. Celková presnosť je 90%, ak sú vstupné obrazy CT manuálne kalibrované do správnej vertikálnej polohy. Tento spôsob je veľmi presný a jednoduchý, ale nezohľadňuje automatickú rotáciu vstupného obrazu. Taktiež je určený len pre obrazy, ktoré ilustrujú vnútro-mozgové krvácanie. Stupeň posunu je meraný len v jednej úrovni, nazvanej „framen Monro“, nachádzajúcej sa v hornej polovici skúmaného obrazu. Z toho hľadiska je stupeň kalibrácie obrazu v ideálnej vertikálnej polohe veľmi dôležitý. V tomto prípade je algoritmus výberu vhodného rezu nepotrebný, pretože je manuálne vopred vybraný taký rez, ktorý je z hľadiska vlastností najlepší. V tomto spôsobe vyhľadávania ideálnej stredovej osi je počiatočný a koncový bod položený ináč, než predchádzajúce metódy. Počiatočný bod je umiestnený v čelnom falxe (obr. 1.2), koncový bod v ťažisku lebečnej časti, teda centre vnútro-mozgovej časti. Táto upravená definícia ideálnej osi je vybraná na základe jednoduchosti a robustnosti. Nájdenie hmotnostného centra zahŕňa maticu, čo algoritmus zjednodušuje a zefektívňuje. Tento spôsob detekcie sa odvoláva k tomu, že detekcia falxov znižuje robustnosť algoritmu, keďže predpokladá neporušenú lebečnú časť. Taktiež vzniká pravdepodobnosť nepresného určenia pozície falxu. Nájdenie stredu vnútro-mozgovej časti spôsobuje nízku výpočtovú záťaž. Výhodou tohto prístupu je rýchlosť a spoľahlivosť úspešnosti aplikácie. Nevýhodou je manuálna kalibrácia obrazu do správnej roviny a obmedzenie na obrazy s viditeľným vnútro-mozgovým krvácaním. [Huan2017]

Po analýze problematiky bol následne naformulovaný cieľ práce. Cieľom práce je vytvorenie softvéru pre automatické spracovanie dát, schopné analyzovať posun stredovej osi mozgových štruktúr z CT obrazu. Vstupné dáta predstavujú medicínske obrazy CT mozgu, získané z vyšetrení pacientov pomocou CT prístrojov. Pomocou navrhnutých algoritmov budú dáta podrobené analýze a vyhľadaniu rôznych závislostí a ďalej vyhodnotená veľkosť posunu za jeho stredovú os. Výstupné hodnoty posunu a pomeru hemisfér budú štatisticky spracované, čo povedie k vyhodnoteniu úspešnosti výslednej aplikácie.

Pridaním používateľského rozhrania aplikácia zobrazí okrem číselných výsledkov aj grafické zobrazenie posunu. To napomôže lekárom a praktikantom pri vyhodnocovaní urgentných prípadov.

Na dosiahnutie cieľov tejto práce vznikla nutnosť vyriešiť nasledovné úlohy:

1. Výber správneho programovacieho jazyka a implementačného nástroja
2. Vytvoriť grafické rozhranie pre zobrazovanie výsledkov
3. Implementovať detekčné algoritmy
4. Implementovať kvantifikačné algoritmy
5. Porovnať výsledky aplikácie s výsledkami lekára

2. VÝVOJOVÉ PROSTRIEDKY A TVORBA POUŽIVATEĽSKÉHO PROSTREDIA

2.1 VÝBER VHODNÉHO PROGRAMOVACIEHO JAZYKA

Rozhodovanie pre výber vhodného programovacieho jazyka prebiehal na základe možností a vhodných knižníc, ktorými daný jazyk disponuje. Nasledujúca tabuľka (tab. 1) popisuje možnosti troch desktopových a voľne dostupných jazykov, vhodných pre danú prácu. Konkrétne sa jedná o jazyky Python, Java a C++. Mnoho aplikácií pre automatické spracovanie obrazov, je naprogramovaných v jazyku Python. Ten disponuje mnohými grafickými knižnicami. Je viac vhodnejší pre strojové učenie, oproti porovnávaným programovacím jazykom. Vďaka produktivite, ktorou Python disponuje (schopnosti zostrojiť algoritmus v menšom počte riadkov kódu) je v porovnaní s Java a C++ taktiež výhodnejší. Nevýhodou je čas kompilácie, v tomto prípade je najrýchlejší C++. Na základe odporúčaní zamestnanca Siemensu sme nepovažovali rýchlosť spracovania za prioritu a ako programovací jazyk pre danú prácu bol vybraný jazyk Python. Vzhľadom k jeho popularite, nízkemu záberu pamäte, čitateľnosti kódu (ktorá je potrebná v budúcom vývoji aplikácie) a schopnosti daného programovacieho jazyka bežať multi-platformovo oproti porovnávaným jazykom.

Tab. č. 1, porovnanie desktopových programovacích jazykov [web1]

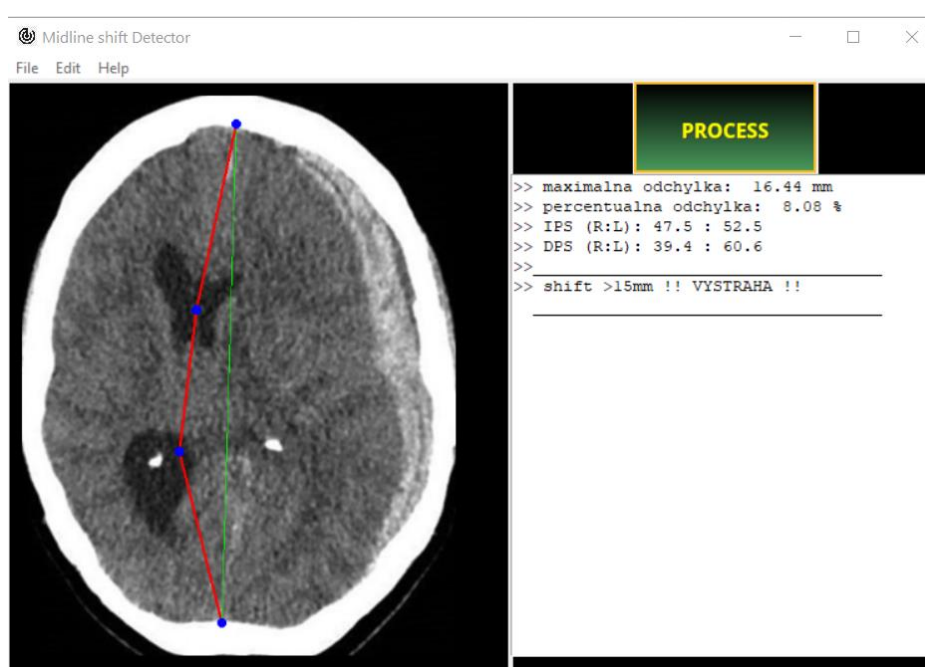
	Java	C++	Python
čitateľnosť kódu
pravidla skriptovania	statické, silné	statické, silné	zásadné
platforma	Windows, Solaris, Linux, OS X	Linux, MacOS, Solaris	multi-platformový
záber pamäte (MB)	40	6	14
popularita	19.10%	5.90%	29.90%
trend	-2.20%	-0.40%	4.60%
beh programu (milisekúnd)	1.89	1.56	7.19
určené pre	multiplatformové aplikácie	herné nástroje	AI, strojové učenie
produktivita	menej produktívny (premenné predom definované)	menej produktívny (premenné predom definované)	vysoká produktivita v malom množstve riadkov kódu
licencia	zadarmo	zadarmo	zadarmo

Programovanie prebiehalo v IDE PyCharm. V aplikácii sú zahrnuté tieto knižnice:

1. **Knižnica CV2** určená na riešenie problémov s obrazmi. Knižnica je multiplatformná a bezplatná. Obraz uloží ako maticu pixelov, ktorú je možné spracovať pomocou vstavaných funkcií. V tejto práci som použil funkcie na konverziu obrazu do 1-kanálového obrazu, v sivej škále a funkciu gaussian blur, pre identifikáciu hraníc.
2. **Knižnica TIME** meria čas, počas ktorého aplikácia spracúva sekvenciu funkcií. Kontrola času slúžila ako indikátor zložitosti jednotlivých algoritmov, ktoré boli následne optimalizované aby čas spracovania bol minimalizovaný.
3. **Knižnica NUMPY** pridáva podporu pre veľké mnohorozmerné polia (matice), s rozsiahlou zbierkou matematických funkcií na vysokej úrovni a minimálnej časovej náročnosti pre prácu s poľami a viacrozmernými poľami. V prípade navrhovanej práce pre spracovanie matic obrazov.

2.2 IMPLEMENTÁCIA GUI – GRAFICKÉHO ROZHRAVIA

Grafické plátno slúži na reprezentáciu výsledkov, jak číselných, tak aj grafických. Je rozdelené na dve polovice, v ľavej strane sa nachádza obraz vloženého obrazu, ktorý po spracovaní zobrazí ideálnu úsečku (IML) zelenou farbou, deformovanú krivku (DML) červenou. V pravej časti plátna sa nachádza tlačidlo spracovania a konzola výpisu výsledkov. Užívateľ má v ponuke zvoliť nový obraz, nastaviť preferovaný jazyk, z pomedzi jazykov: slovensky, anglicky. Nastaviť šírku lebky v milimetroch, uložiť modifikovaný obraz alebo uložiť aktuálne výsledky do súboru. Taktiež je mu poskytnutý užívateľský manuál. Ukážka grafického plátna je zobrazená na obrázku 2.1.



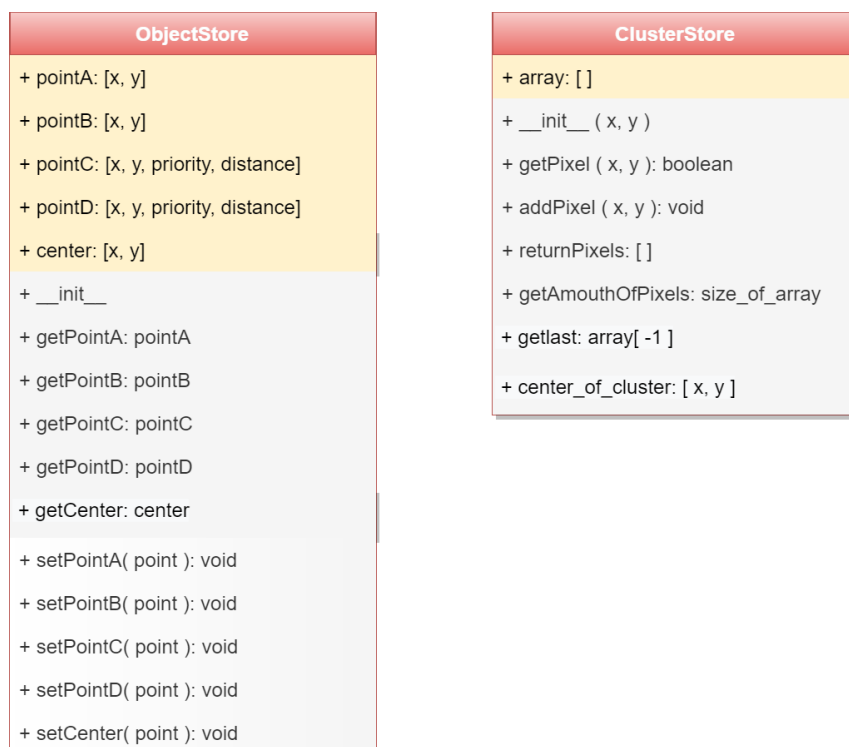
Obr. 2.1, Grafické okno užívateľa

2.3 IMPLEMENTÁCIA OBJEKTOV

Program obsahuje dvojicu objektov, do ktorých sa zapisujú radiacie body a zhľuky potrebné pre identifikáciu radiacích bodov C a D. K prístupu k objektom disponuje hlavná radiaca funkcia, ktorá objekty pridieľuje pomocným funkciám pre detekciu radiacích bodov a zhľukov a následným zapísaním bodov a zhľukov do objektov. Diagram tried je zobrazený na obrázku 2.2.

ObjectStore – obsahuje atribúty riadiacich bodov a verejných metód, vďaka ktorým sa dá pristupovať k riadiacim bodom, zapisovať a vyberať ich. Taktiež obsahuje bod označujúci stred lebečnej časti obrazu (center) obsahujúci hodnoty súradníc. Atribúty pre riadiace body A a B (pointA, pointB) sú typu dvojrozmerného poľa, ukladajúci hodnotu na x-ovej a y-ovej osi. Narozdiel od bodov C a D (pointC, pointD), ktoré okrem hodnôt osí si ukladajú aj prioritu zhuku a vzdialenosť od ideálnej úsečky. Metódy get a set pristupujú k týmto atribútom, umožňujú zapisovať a čítať tieto atribúty.

ClusterStore – obsahuje jediný atribút typu pole (array). Do poľa sú zapisované také pixely, ktoré patria do aktuálneho zhuku. K zhuku sa dá prístupit' pomocou verejných metód. Umožňujú pridávanie nového pixelu (addPixel), kontrolu, či sa daný pixel nachádza v zhuku (getPixel), počet pixelov v zhuku pre kontrolu veľkosti zhuku (getAmouthOfPixels), ako aj metódu pre vrátenie kópie poľa (returnPixels). Objekt taktiež disponuje metódou, ktorá nájde stred zhuku (center_of_cluster) a tiež metódou pre vrátenie posledného pixelu v poli (getlast). Túto metódu využíva funkcia zhukovania pre kontrolu posledného nájdeného pixelu v prehľadávaní.

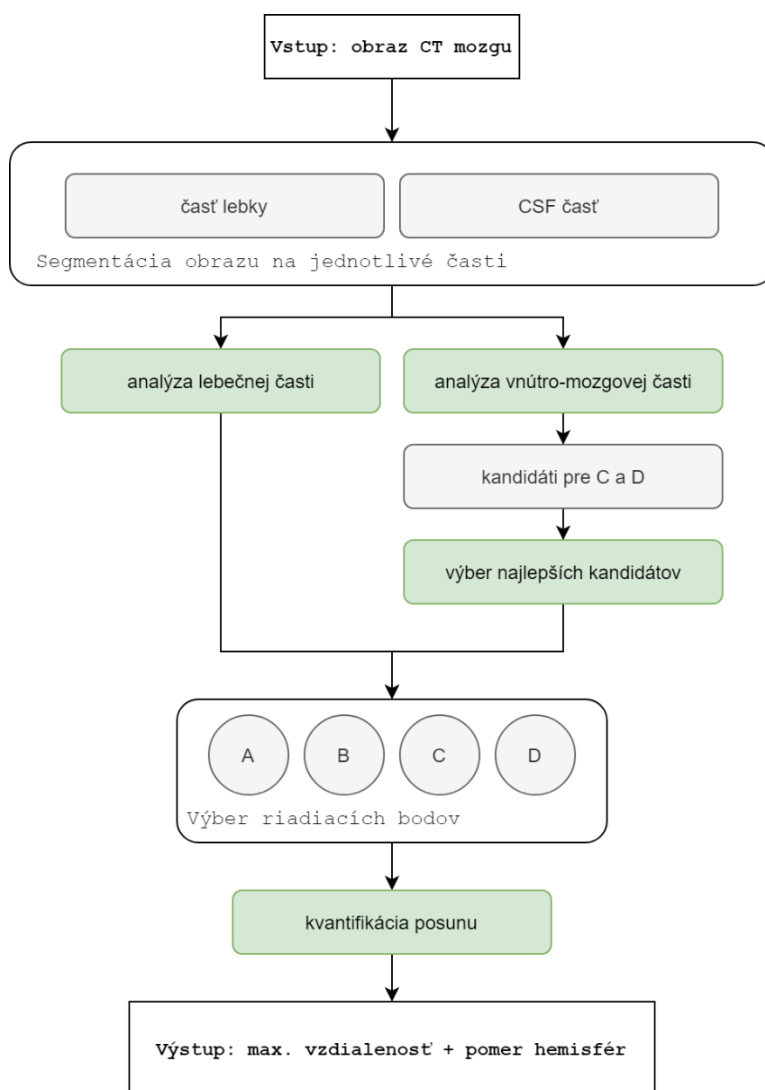


Obr. 2.2, Objekty použité v aplikácii

Uviedli sme si návrh celej aplikácie, výber programovacieho jazyka a vývojového prostredia. Vytvorili sme používateľského rozhranie pre zobrazovanie výsledkov a implementovali objekty pre ukladanie riadiacich bodov a zhlukov. V ďalšej kapitole si bližšie popíšeme jednotlivé časti aplikácie a jej prístupu k spracovávanému obrazu.

3. POPIS APLIKÁCIE A IMPLEMENTÁCIE

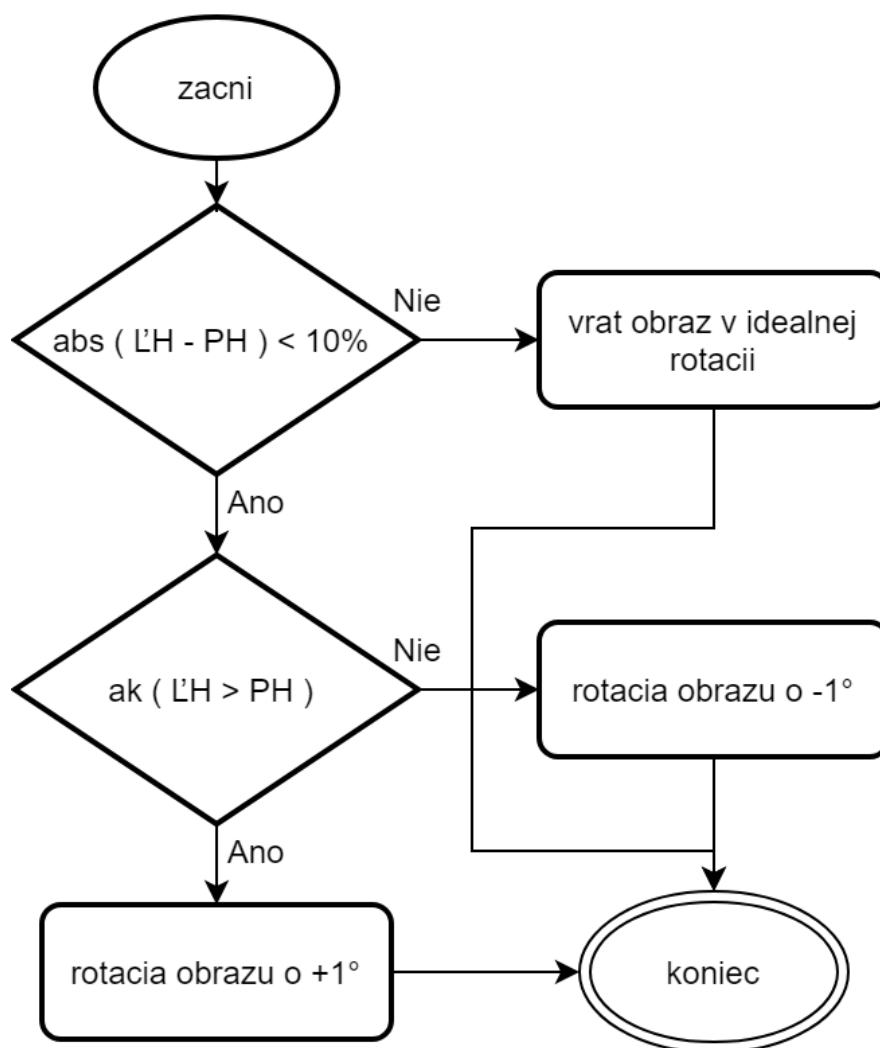
Beh aplikácie postupuje sekvenčne v poradí ôsmich krokov. Diagram na obr. 3.1 zobrazuje proces od vstupu obrazu CT mozgu až k výstupu číselných hodnôt posunu mozgových štruktúr.



Obr. 3.1, Diagram algoritmu používajúci značkovacie modely

3.1 AUTOROTÁCIA

Pre správne spracovanie obrazu je potrebné dosiahnuť symetrický stav CT obrazu mozgu podľa vertikálnej osi. Spôsob, ako to dosiahnuť, je rotáciou obrazu. To umožní odstrániť stav, kedy je obraz v nesprávnom uhle. Ako výsledok je oblasť lebky znázorňujúca elipsu, ktorá bude mať dva najvzdialenejšie body vo vertikálnej osi.

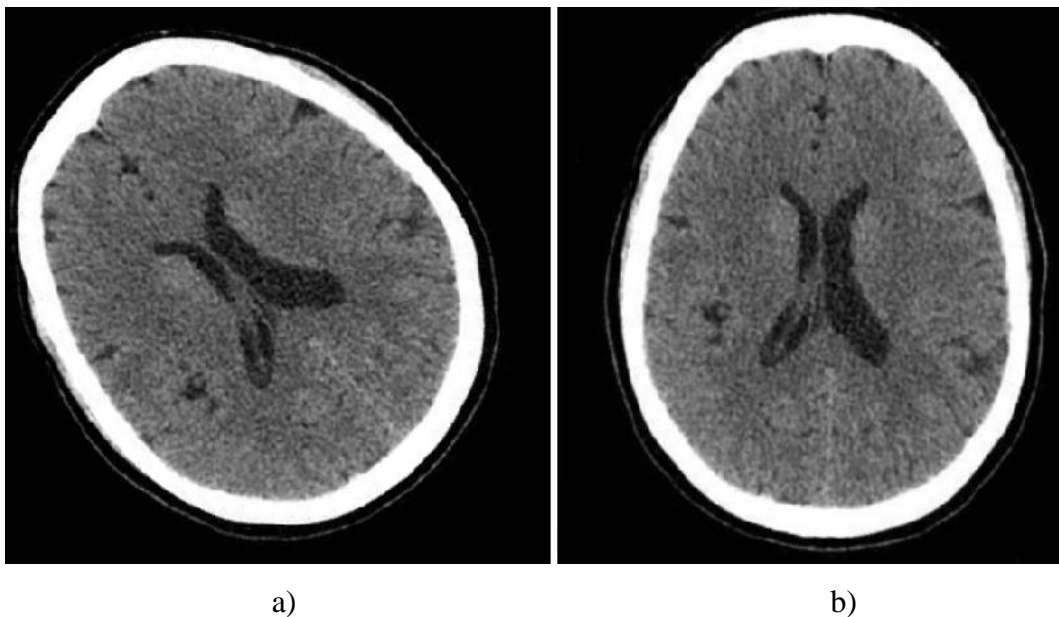


Obr. 3.2, Autorotácia v diagrame

Princíp autorotácie spočíva v rozdelení plochy ohraničenej lebkou na štyri kvadranty. Stred je určený ako ťažisko vnútro-mozgovej plochy. Po vypočítaní veľkosti plochy sa porovnávajú kvadranty medzi sebou, každý so svojim horizontálnym susedom. Ak sa zistí veľký rozdiel, obraz sa rotuje do tej strany, v ktorom je plocha rozdielna (závisí podľa porovnávania horných kvadrantov, alebo dolných kvadrantov, ľavého a pravého). V prípade, že je horný, ľavý kvadrant menší, ako jeho susedný pravý,

je obraz rotovaný o $+1^\circ$. V opačnom prípade je obraz rotovaný o -1° . Postup sa opakuje s novým zrotovaným obrazom. Cyklus prebieha dovtedy, kým nie je odchýlka medzi veľkosťami plôch susedných kvadrantov minimálna z hľadiska bezpečnosti pred zacyklením programu. Výsledkom nie je obraz v dokonalej vertikálnej polohe, ale stupeň vychýlenia, pretože funkcia rotácie degraduje obraz natoľko, že po desiatich rotáciach je obraz priam nečitateľný. Z toho dôsledku vráti funkcia auto-rotácie stupeň rotácie, ktorým je následne riadiaci obraz zarotovaný a tým aj kvalita obrazu zachovaná.

V prípade takého obrazu, v ktorom je lebka nesymetrická a neforemná, je obraz rotovaný do nie ideálnej vertikálnej polohy. To má však na výpočtový algoritmus v nasledujúcich fázach mierny efekt. Dôležité je ale to, v akom sklone sa nachádza lebka v počiatočnom stave. V prípade že je lebka v správnom uhle alebo v prípade nesymetrickej lebky, algoritmus nespustí auto-rotáciu, ak je rozdiel plôch susedných kvadrantov minimálny. Z hľadiska skúmania a pozorovania je konštanta pre veľkosť rozdielov daných kvadrantov, pre rozhodovanie o auto-rotácii, nastavená empiricky na hodnotu 10.



Obr. 3.3, výsledok autorotácie : (a) vstupný obraz (b) obraz po automatickej rotácii

```

def auto_rotation(image, search):
    image_org = np.copy(image)

    ideal = False
    center = [0,0]
    gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    blur = cv2.GaussianBlur(gray, (7, 7), 0)
    res = select_skull(blur, image, search)
    skull_image = res[0]
    canny = cv2.Canny(skull_image, 0, 255)
    pre_center = [image.shape[0], image.shape[1]]

    # vyhľadaj stred lebky a prepíš stred na stred obrázy canny
    while pre_center != center:
        pre_center = center
        center = move_to_center(canny)

    angle = 0
    i = 0

    # vyzerá nebezpečne ale k zacykleniu nedojde
    while ideal == False:

        size = (int)((canny.shape[0] * canny.shape[1])/1000)

        distance_left = 0
        distance_right = 0
        # vyrataj veľkosť ploch ľaveho horného a pravého horného kvadrantu
        for y in range(center[1], 0, -20):
            pre_center[1] = y
            if canny[y, center[0]] > 240 or center[2] > y:
                break
            for x in range(center[0], 0, -1):
                if canny[y, x] > 240:
                    distance_left += calc_distance(x, y, pre_center)
                    break

        for x in range(center[0], canny.shape[1]-1, 1):
            if canny[y, x] > 240:
                distance_right += calc_distance(x, y, pre_center)
                break

        # porovnaj s tým že ich veľkosti musia byť +- 10%
        if abs(distance_right - distance_left)/size < 0.10:
            ideal = True
        else:
            if distance_left > distance_right:
                skull_image = rotate_image(skull_image, -1)
                angle -= 1
            else:
                skull_image = rotate_image(skull_image, 1)
                angle += 1

        canny = cv2.Canny(skull_image, 0, 255)
        pre_center = [image.shape[0], image.shape[1]]

        # opat nakalibruj stred lebky
        while pre_center != center:
            pre_center = center
            center = move_to_center(canny)

        i += 1

    image_org = rotate_image(image_org, angle)
    return image_org

```

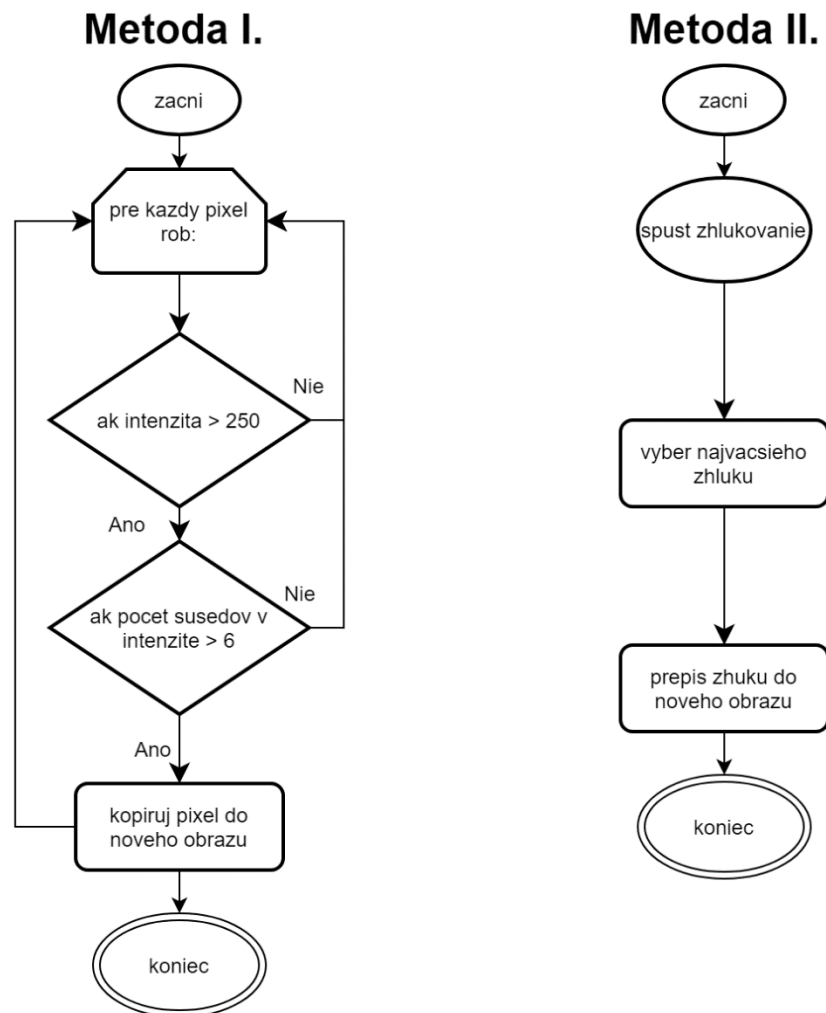
Obr. 3.4, Fragment kódu autorotácie

V cykle sú počítané veľkosti kvadrantov, ak sú ich veľkosti $\pm 10\%$, cyklus sa ukončí s príslušným uhlom rotácie, ktorým sa následne zarotuje vstupný obraz (pozri obr. 3.4).

3.2 SEGMENTÁCIA LEBEČNEJ ČASTI

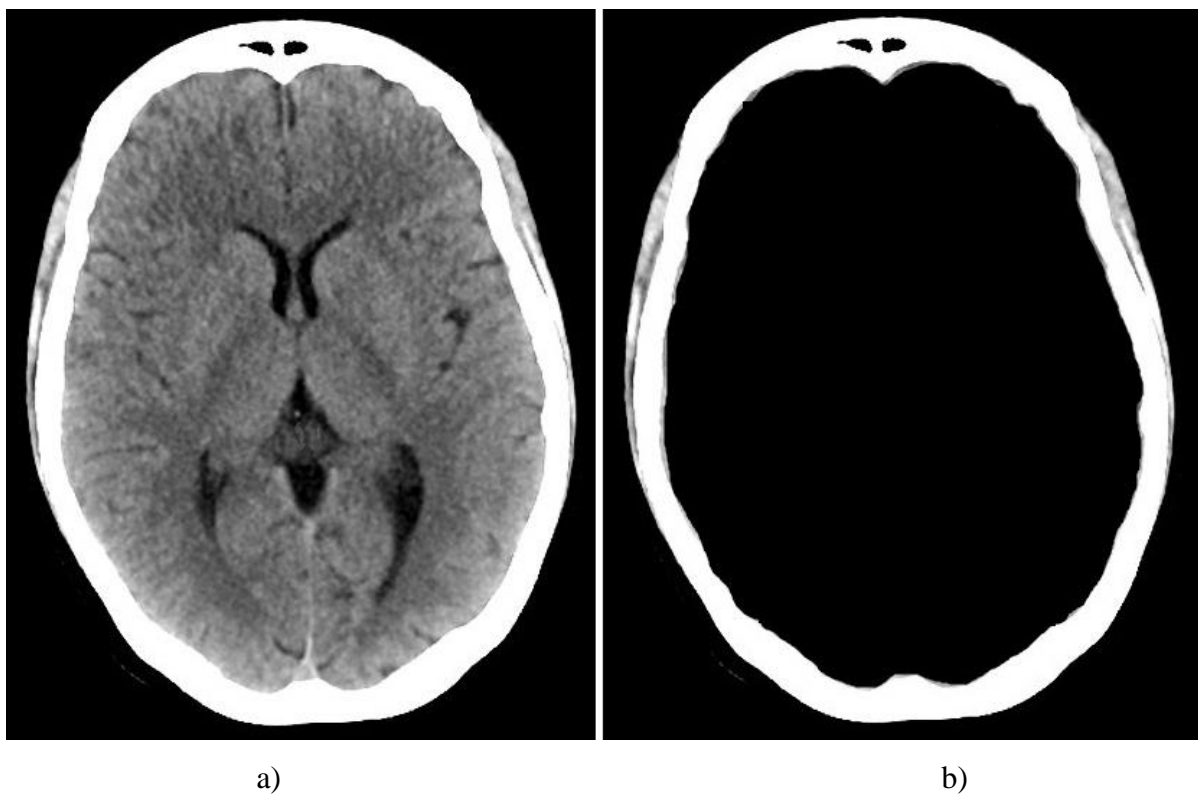
Až je obraz v správnej vertikálnej polohe, je vyčlenená oblasť lebky. Ako vidíme na obrázku č. 3.6 (b), lebka tvorí obvod zreteľne viditeľný bielu farbou. To je dôležitý faktor pri hľadaní tohto segmentu a následnom prepise do pomocného obrazu, ktorý posluží v ďalších funkcionalitách. Výber lebečnej časti funguje na báze hľadania susedných pixelov rovnakej bielej intenzity. Obraz je v type 1-channel, teda odtieni šedej. Ak má prehládavaný pixel vo svojom okolí viac ako 7 susedných pixelov rovnakej intenzity, predpokladá sa, že je súčasťou lebečnej časti. V takomto prípade je pixel zakreslený do nového pomocného obrazu. Princíp výberu viacerých susedov rovnakej intenzity je navrhnutý z dôvodu viacerých faktorov, ktoré by mohli negatívne ovplyvniť algoritmus výberu lebečnej časti. Výhodou je vysoká pravdepodobnosť úspešnej detekcie lebečnej časti a nízka časová náročnosť. V prípade zlyhania metódy výberu susedných pixelov, kvôli nedostačujúcej hrúbke lebky, je v ponuke metóda zhlukovania, ktorá vyberie najväčší zhluk o intenzite bielej farby. Táto metóda má omnoho vyššiu pravdepodobnosť správnej detekcie lebky, priam až 100% (nie celková úspešnosť). Nevýhodou je jej časová zložitosť, kedy presiahne metódu hľadania susedov rovnakej intenzity o viac ako 50-násobok. (Priemerný výpočtový čas detekcie lebečnej časti pre metódu hľadania susedných pixelov je

0.6s, pre metódu zhlukovania 34s pri obraze 300x400px). Nevýhodou prvej metódy je strata krajných pixelov, ktoré v zriedkavých prípadoch majú vplyv na výsledok aplikácie.



Obr. 3.5, Výber lebečnej časti v diagrame

Algoritmus výberu lebečnej časti má napomôcť nie len pri hľadaní lebečnej časti, ale taktiež nájsť štyri hraničné body pre následné orezanie obrazu. Sú nimi maximum a minimum na osi X a osi Y. Znázorňujú súradnice pre najľavejšiu a najpravejšiu stranu lebky, najvyššiu a najspodnejšiu časť. Slúžia ako riadiace body pre strih obrazu z hľadiska optimalizácie pre výpočtový čas. Centrom skúmania je vnútro mozgovej časti ohraničené lebkou, preto plocha mimo nej je zanedbateľná a z hľadiska algoritmov vyčerpávajúca pre jej prechádzanie.



Obr. 3.6, Výber lebečnej časti : (a) Obráz lebky pred segmentáciou (b) obraz po vyselektovaní lebečnej časti

Prehľadávanie vnútornej časti, teda plochy vnútri lebečnej časti, je náročná úloha. V ďalšej sekcii si ale povieme, aký postup prehľadávania je použitý a akú rolu v tom má lebečná časť, ako ohraničenie vnútro-mozgovej časti.

```

try:
    # prehlada kazdy pixel a ak ma vo svojom okruhu viac ako 6 pixelov rovnakej intenzity,
    # tak ho preklopi do dalsieho pomocneho obrazu
    for x in range(0, blur.shape[0] - reserve):
        for y in range(0, blur.shape[1] - reserve):
            count = 0
            lane2_image[x, y] = 0
            # if blur[x, y] == [ 0, 111, 0]:
            if blur[x, y] > 250:
                for m in range (x-1, x+reserve):
                    for n in range (y-1, y+reserve):
                        if blur[m, n] >= 230:
                            count += 1
            if count >= 7:
                lane2_image[x, y] = (255, 255, 255)
                if most_up == 0:
                    most_up = x
                most_down = x
                if most_left > y:
                    most_left = y
                if most_right < y:
                    most_right = y
except:
    err += 1

```

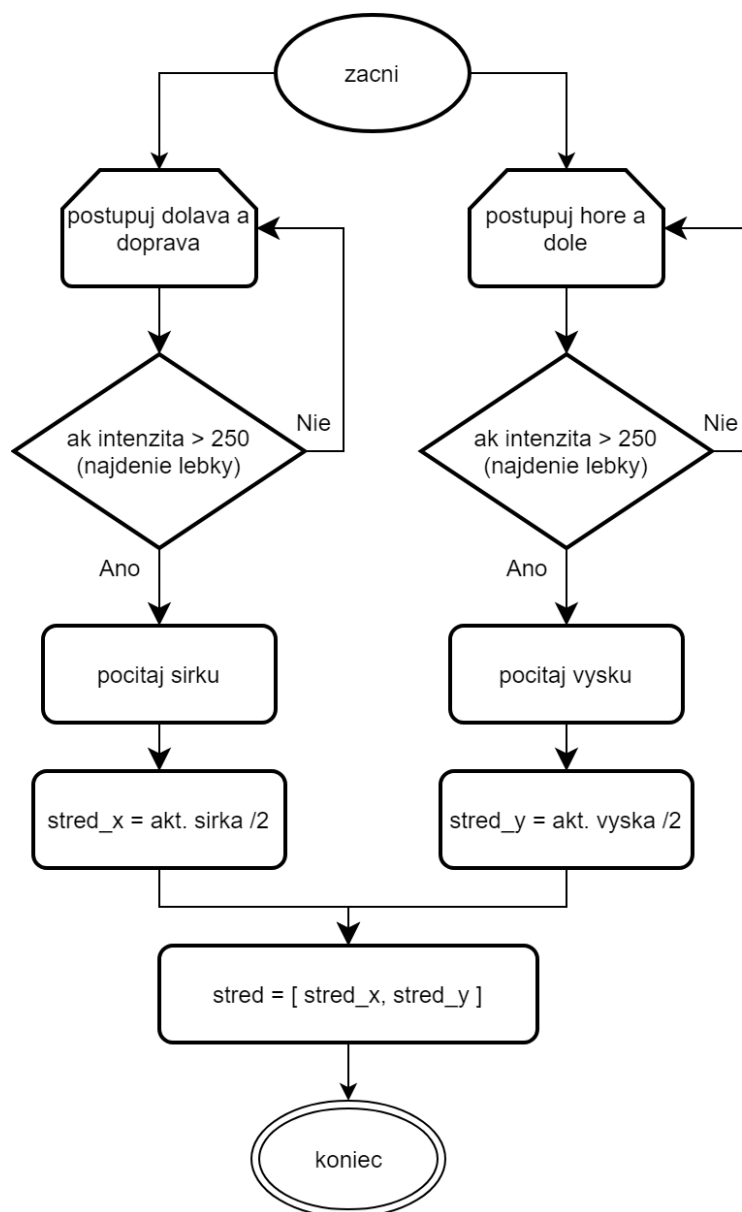
Obr. 3.7, Fragment kódu výberu lebečnej časti

Algoritmus sa pokúsi pre každý pixel obrazu nahliadnuť na jeho intenzitu a v prípade vyhovujúcej intenzity kontroluje susedné pixely. Pri zadanom množstve susedných pixelov rovnakej intenzity je tento riadiaci pixel prekopírovaný do nového obrazu, slúžiaci ako lebečný segment obrazu (pozri obr. 3.7)

3.3 VYHLADANIE STREDU (ŤAŽISKA) LEBEČNEJ ČASTI

Mnoho metód v sekvencii spracovania potrebuje svoj štartovací bod, alebo os. Ten ale nemôže byť ľubovoľným bodom vo vnútro-mozgovej časti. Pre správne prehľadávanie je potrebné nájsť ťažisko vnútornej časti, čo je triviálna úloha. Jeho správnu polohu získame prehľadávaním dĺžok v horizontálnej a vertikálnej rovine. Ako štartovací bod sa vyberie stred obrazu, teda polovica šírky a dĺžky obrazu. Označme si ho ako bod $M = [h, w]$, kde h je výška obrazu / 2 a w šírka / 2. Vzhľadom k ideálnej vertikálnej rovine obrazu určíme stredovú os, ktorá prechádza bodom M . Jej počiatkový bod je v ľavej strane lebečnej časti, jej koncový bod v pravej strane lebečnej časti. Jej koncový bod získame prehľadávaním sekvencie pixelov smerom do ľavej a pravej strany, pričom ak narazí na pixel bielej intenzity, teda lebečnú časť, považuj tento bod za svoj hraničný. Po objavení týchto bodov je vypočítaná vzdialenosť a určený stred osi. V rovnakom prípade počítania vzdialenosti medzi najvyšším a najnižším bodom je použitý rovnaký mechanizmus preklopený z horizontálneho prehľadávania na vertikálne a následne určený stred osi. Tento postup zaistí nájdenie ideálneho centra vnútro-mozgovej časti v každom prípade. Treba podotknúť, že štartovací bod sa nachádza v strede

obrazu. Ak sa štartovací bod nenachádza v ohraničení lebečnej časti, je stred vyčíslený nesprávne a nasledujúca sekvencia funkcií predom neúspešná. V tomto prípade je nevyhnutné, aby bol objekt skúmania v obraze v takej polohe, aby stred obrazu bol v prieniku s vnútro-mozgovou časťou.



Obr. 3.8, Vyhľadanie stredu obrazu v diagrame

```

# prechadzaj dolava a doprava a meraj vzdialenost, potom sa posun do jej stredu
for x in range(windowWidth // 2, 0, -1):
    if canny>windowHeight // 2, x] > 250:
        leftCoord = x
        break

for x in range(windowWidth // 2, canny.shape[1], 1):
    if canny>windowHeight // 2, x] > 250:
        rightCoord = x
        break

distanceM = rightCoord - leftCoord
x_center = distanceM//2 + leftCoord

# ----- for height
# pto iste pre vysku, meraj vzdialenost a posun sa do stredu
for y in range(windowHeight // 2, 0, -1):
    if canny[y, x_center] > 250:
        upCoord = y
        break

for y in range(windowHeight // 2, windowHeight, 1):
    if canny[y, x_center] > 250:
        downCoord = y
        break

distanceM = downCoord - upCoord
y_center = distanceM // 2 + upCoord

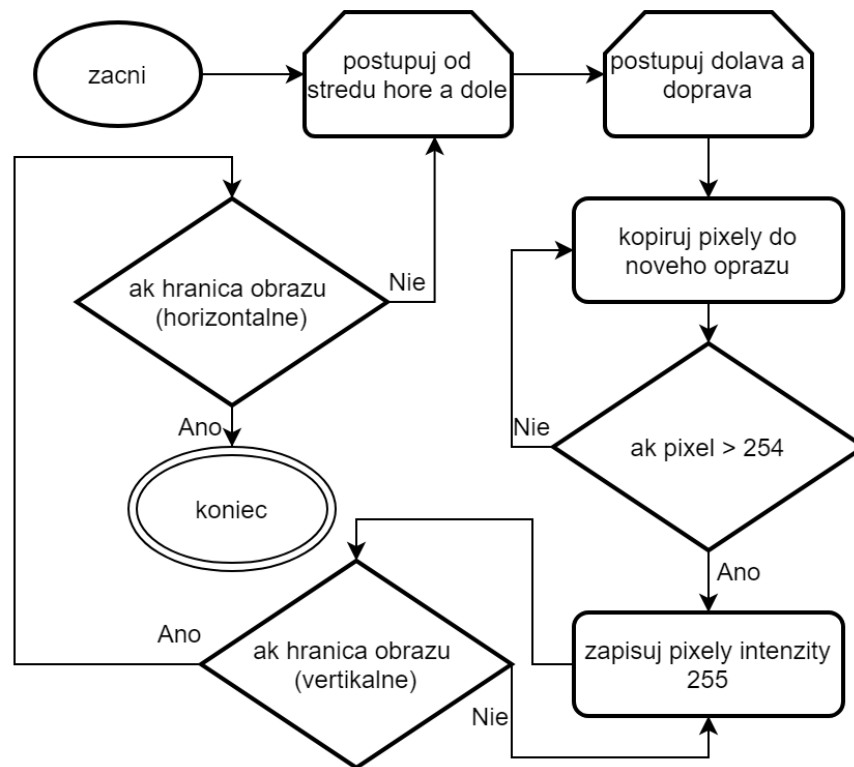
```

Obr. 3.9, Fragment kódu vyhľadania stredu obrazu

Pixely sú vyberané tak, že algoritmus prechádza od stredu obrazu, do strán (doleva, doprava, hore a dole). Každý aktuálny pixel je podrobený kontrole, či jeho intenzita rovnaká ako intenzita lebky. V prípade rovnakej intenzity je vyčíslená vzdialenosť pixela od stredu ťažiska, sčítaná s protichodnou vzdialenosťou. Stred tejto vzdialenosti je vyčíslený ako polovica vzdialenosti a bod ležiaci na nej je označený ako stred úsečky (pozri obr. 3.9)

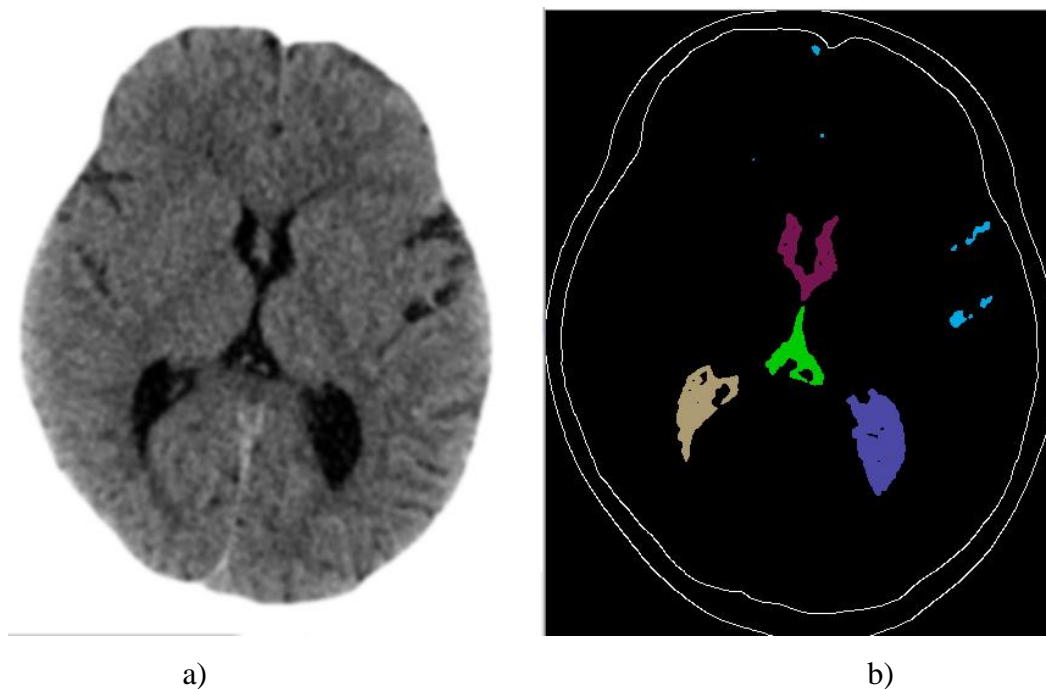
3.4 SPRACOVANIE CSF-ČASTI

Najdôležitejšou plochou celého skúmania tohto algoritmu je vnútro-mozgový priestor ohraničený lebkou, nazvime ju CSF časťou (cerebrospinal fluid map). Túto časť k správne spracovaniu nasledujúcich algoritmov je potrebné ponechať a odstrániť plochu mimo nej. Pre rýchlejšie spracovanie je potrebné zvoliť jednotnú farbu pixelov, ktorá bude počas spracovania ignorovaná. Vieme to, že vnútro-mozgová časť v 1-kanálovom type obrazu je v odtieňoch šedej, teda z intervalu $\langle 1, 254 \rangle$. Vzhľadom k tomu, že časť lebky je prekopírovaná do pomocného obrazu v intenzite bielej farby, konkrétne hodnotu 255, celé pozadie mimo plochy skúmania transformujeme do rovnakej bielej farby s hodnotu 255, ako farba lebky.



Obr. č. 3.10, Diagram spracovania CSF časti

V tejto práci je problém riešený takýmto spôsobom: začni v strede obrazu, postupuj dvoma smermi, doľava a doprava. Kopíruj postupne všetky pixely, pokiaľ nenarazíš na pixel o intenzite 255, čo značí, že sme došli k hranici vnútro-mozgovej časti s lebkou. Tento postup sa vykoná cyklicky pre každú úroveň výšky obrazu. Pre jednoduchší spôsob postupu je z hľadiska náročnosti jednoduchšie, ak algoritmus kopíruje všetky pixely od začiatku cyklu, než začínať na nulových súradniciach.



Obr. 3.11, Obráz po spracovaní CSF a zhlukovaní (a) CSF časť (b) zhluky po procese clusteringu

```

for invert in range(2):
    for y in range(center[1], direction, step):
        left_was = False
        right_was = False
        for x in range(center[0], 0, -1):
            if left_was:
                csf_image[y, x] = (255, 255, 255)
                continue
            csf_image[y, x] = image[y, x]
            if canny[y, x] > 240:
                csf_image[y, x] = (255, 255, 255)
                left_was = True

        for x in range(center[0], canny.shape[1] - 1, 1):
            if right_was:
                csf_image[y, x] = (255, 255, 255)
                continue
            csf_image[y, x] = image[y, x]
            if canny[y, x] > 240:
                csf_image[y, x] = (255, 255, 255)
                right_was = True

direction = canny.shape[0]
step = 1

```

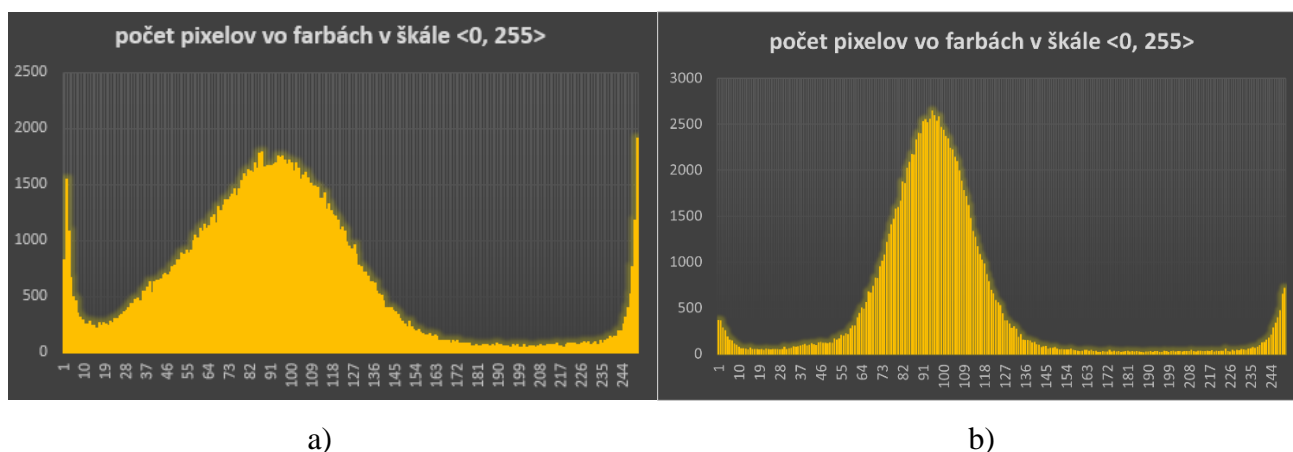
Obr. 3.12, Fragment kódu spracovania CSF časti

V cykle algoritmus prechádza od horizontálnej stredovej osi a kopírujú sa pixely do novej mapy. Po nájdení časti lebky sú v rovine prehľadávania do novej mapy vkladané pixely bielej intenzity. (pozri obr. 3.12)

3.5 CLUSTERING (ZHLUKOVANIE)

Proces zhľukovania je časovo najnáročnejšia metóda, akú aplikácia využíva pre identifikáciu kľúčových vlastností v CSF časti. K tomu, aby sme dokázali nájsť v objekte skúmania zadnú mozgovú komoru (3DV) a čelnú mozgovú komoru (FH), je potrebné si uvedomiť ako vyzerajú vo vstupnom obraze. Z pozorovaní vieme intuitívne odhadnúť ich pozíciu v obraze. Vzhľadom k tomu, že algoritmus nie je schopný intuitívne určiť ich pozíciu, je potrebné určiť správnu polohu v obraze matematickými metódami. Z každého CT obrazu vieme s istotou povedať, že tieto objekty sú v intenzite čiernej farby. Aby sme dokázali určiť správny interval hľadania hornej a dolnej hranice intenzity zhľukovania, je nutné vyčísliť tieto hranice.

Dolná hranica je určená ako najtmavší pixel CSF časti. Horná hranica je o čosi zložitejšia, preto je potrebné určiť hornú hranicu intenzity tmavých pixelov CSF časti tak, aby v procese zhľukovania nebola rátaná intenzita mimo komorových plôch. Tie by boli algoritmicky súčasťou zhľuku, čo by prinieslo irelevantný výsledok. Spôsob výberu intervalu intenzity pixelov, definovaného ako dolná a horná hranica vyhľadávania zhľukov, v procese vývoja bola určená primárne ako $<0; 60>$ na základe empirického pozorovania. Vzhľadom k rôznosti obrazov rozličnej intenzity pixelov vnútro-mozgovej časti nebude vhodné určiť interval intenzity pixelov ako konštantný. Bola by to predom znížená úspešnosť výpočtu, ktorá by fungovala len na obrazy podobnej intenzity.



Obr. 3.13, Histogramy intenzít: v škále 0 až 255 dvoch rôznych CT obrazov

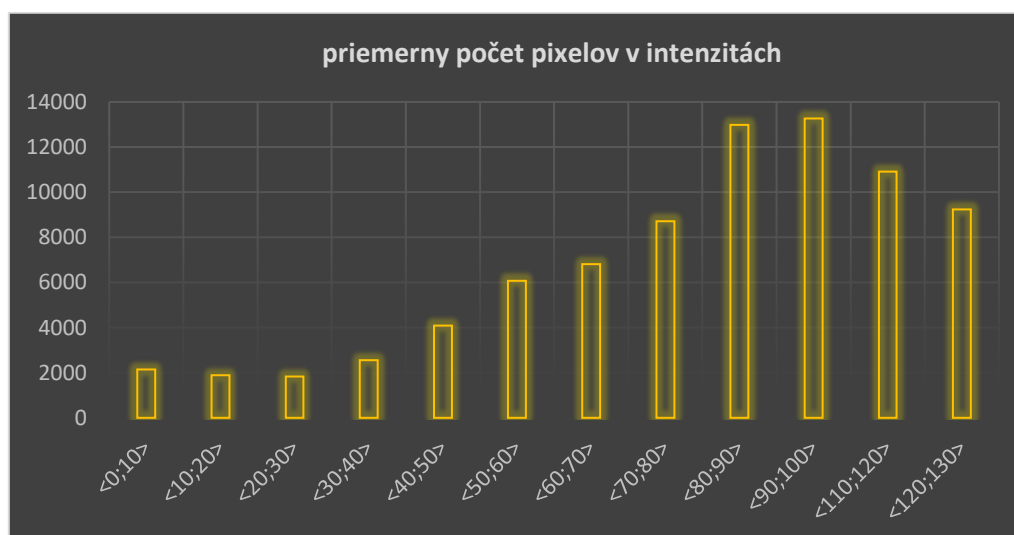
Ako vidíme na obr. 3.13, priemerne v intenzite 95 sa nachádza vrchol počtu pixelov danej intenzity. Hľadanie dolnej hranice je triviálna úloha, ktorá nájde najtmavší pixel v oblasti ohraničenej lebkou. Horná hranica je z hľadiska pozorovaní číslo v intervale $\langle 50; 100 \rangle$. K jej nájdeniu je potrebné si uvedomiť, čo daný graf predstavuje. Vzorky v najnižších intenzitách nízkeho počtu pixelov predstavujú hľadajú zadnú mozgovú komoru (3DV) a čelnú mozgovú komoru (FH) a pixely vo vnútro-mozgovej časti, ktoré sú v procese zhlukovania vyradené z hľadiska skúmania. Prudko rastúca funkcia znázorňuje rastúci počet pixelov v danej intenzite, čo predstavuje celú zvyšnú mozgovú časť mimo mozgových komôr. Čo potrebujeme oddeliť, je táto mozgovú časť od 3DV a FH.

Algoritmus v intervale $\langle 50; 100 \rangle$ hľadá takú intenzitu, ktorej počet pixelov danej intenzity bude o 7-násobok počtu pixelov menej, než pôvodná. Táto metóda zabezpečí nájdenie vysokého rozdielu počtu pixelov v malom úseku. Na základe nájdenia takej hodnoty, podľa zadanej podmienky, algoritmus túto hodnotu použije v procese zhlukovania ako hornú hranicu detekcie zhlukov. Nasledujúca tabuľka zobrazuje vzťah jednotlivých vzoriek a počet pixelov v jednotlivých intervaloch intenzít.

Tab. č. 2, Početnosť intenzít jednotlivých vzoriek

	<0;10>	<10;20>	<20;30>	<30;40>	<40;50>	<50;60>	<60;70>	<70;80>	<80;90>
v 1	1130	906	782	875	1122	1520	2568	4445	7221
v 2	3467	7315	8238	12654	34236	54317	45271	22527	9253
v 3	8478	1493	2897	5728	5095	5960	12194	23284	35357
v 4	958	7721	3129	2902	2871	3775	5122	6766	8077
v 5	3029	1779	1246	931	2149	6775	3737	2437	2432
v 6	2715	3782	3594	4618	7557	16919	20693	23844	23432
v 7	2458	2085	1633	1373	4267	4428	5743	22787	70874
v 8	1187	529	460	818	1117	1160	1564	4002	7538
v 9	1409	873	680	787	938	1375	2351	3929	5930
v 10	1839	871	1213	2254	2079	2477	4091	7173	9131
v 11	2181	1070	1583	3004	2877	2623	3440	5615	8246
v 12	1935	1431	2096	3173	3557	3761	5040	6650	7933
v 13	1924	957	1123	1494	1663	1751	2617	4696	8055
v 14	1520	798	847	1240	1773	2459	4117	6539	9185
v 15	1995	968	825	1034	1135	1408	2474	4949	8780
v 16	1359	1386	2055	2910	3218	3046	2658	2510	3009
v 17	1586	1152	1217	1468	1390	1348	1995	3824	6937
v 18	1217	1314	1771	2164	2505	3366	5383	8036	10689
v 19	1290	899	846	1125	1400	1884	2977	5417	8915
v 20	1240	532	459	635	803	1121	2151	4849	8644
priem	2145.85	1893.05	1834.7	2559.35	4087.6	6073.65	6809.3	8713.95	12981.9

Na obrázku č. 3.14 tak vidíme koreláciu medzi vzorkami z tab. č. 2.



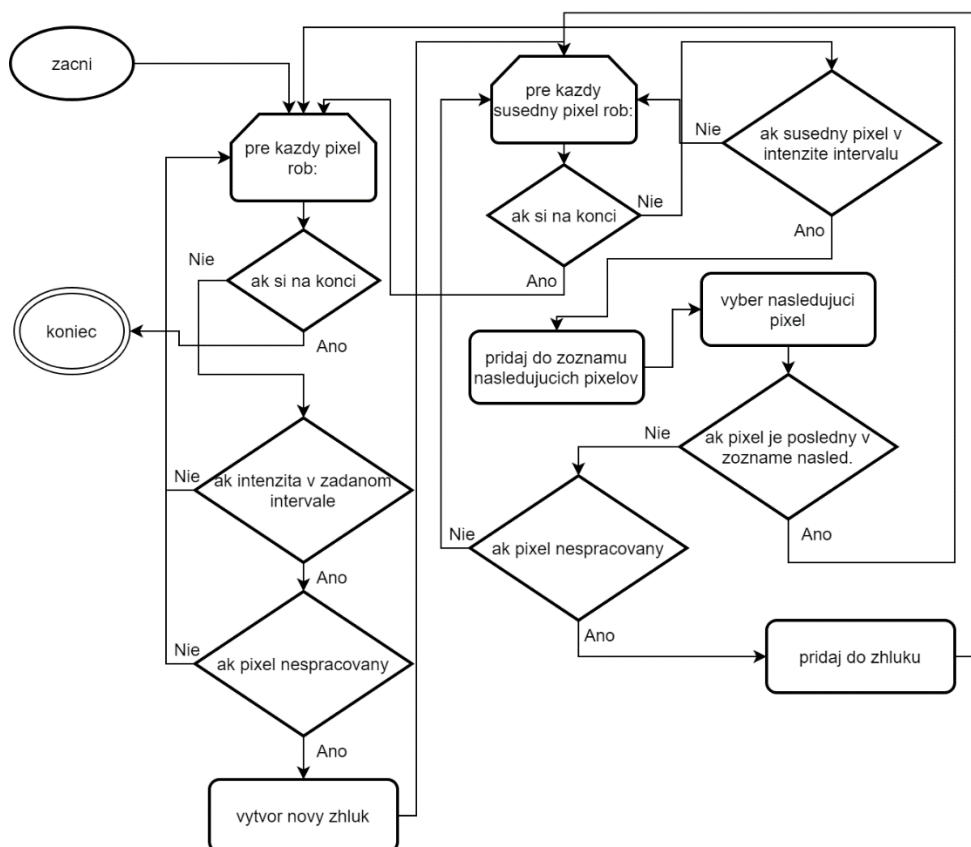
Obr. č. 3.14, Priemerný počet pixelov v intenzitách

Po správnom určení hornej a dolnej hranice postupujeme k spracovaniu každého pixelu nasledovne: prechádzaj v dvojrozmernom poli (matici) každý pixel, dokým nenaraziš na pixel s intenzitou v intervale $< DH, HH >$ (kde DH je dolná hranica, HH horná hranica). Po jeho objavení je pixel prekopírovaný do zoznamu spracovaných pixelov. Následne sa spusti metóda hľadania suseda o rovnakej intenzite z intervalu $< DH, HH >$. Každý aktuálny nespracovaný pixel je zapísaný do zoznamu spracovaných pixelov a priradený do zhluku, v ktorom sa práve nachádza. Ak je v poradí nespracovaný pixel, ktorý nemá spracovaného suseda, je vytvorený nový zhluk a pixel s každým jeho nespracovaným susedom o intenzite spomínaného intervalu priradený do zhluku. Proces priradovania do zhluku končí, až je aktuálny pixel posledný v spracovávanom zozname. Ďalej sa pokračuje rovnakým princípom ako na začiatku: pokračuj ďalej v prehľadávaní, kým nenájdeš pixel o intenzite spomínaného intervalu, ktorý ak je spracovaný, pokračuj ďalej. Až je skontrolovaný každý pixel obrazu, zhluky prejdú filtráciou.

Filtrácia zhlukov je proces, pri ktorom sú odstránené zhluky s nízkym počtom obsahovaných pixelov. Odstránené sú z toho dôvodu, aby nevznikol veľký počet zanedbateľných zhlukov, zvýšila sa rýchlosť prehľadávania dôležitých zhlukov a znížil šum, ktorý sú zanedbateľné zhluky schopné generovať počas rozhodovacieho procesu o kandidátoch. Zoznam spracovaných pixelov bol v priebehu času nahradený pomocným obrazom, do ktorého boli kopírované použité pixely z dôvodu optimalizácie. Z hľadiska časovej náročnosti je rýchlejšie prehľadávať susedné pixely v obraze s aktuálnym platným pixelom, než prehľadávať zoznam použitých pixelov. Ten by sa pri obraze veľkosti 400x600px dokáže naplniť do veľkosti desiatok tisíc prvkov v zozname, čo vedie k enormným počtom iterácií pri prehľadávaní zoznamu pre každý pixel. Pri prehľadávaní pixelov vo veľkom zhluku sa počet iterácii dokázal vyšplhať na 10^{10} , čo činilo operačný čas na niekoľko desiatok minút. Pri zmene kontroly použitých pixelov sa operačný čas rapídne znížil na niekoľko sekúnd.

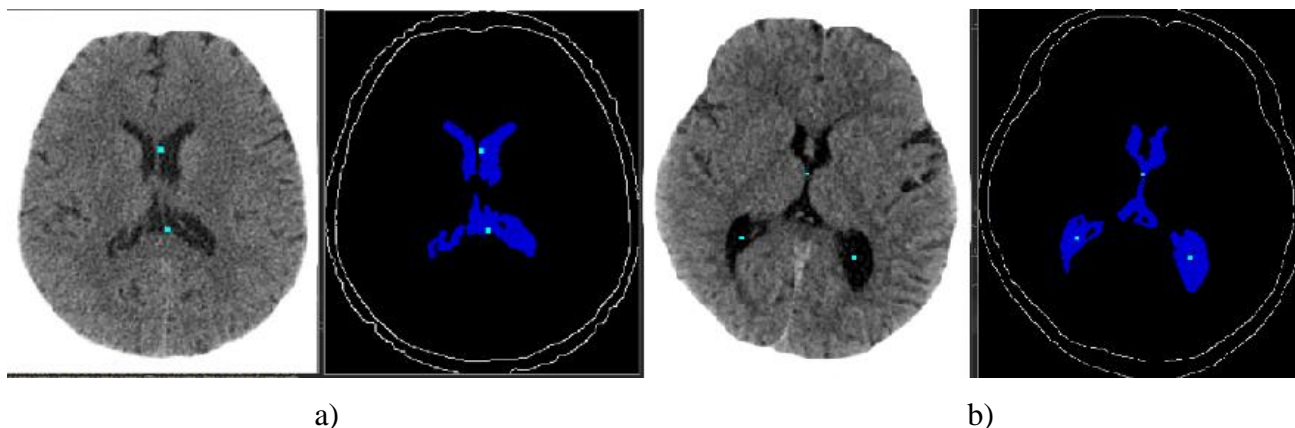
Výsledkom clusteringu sú zhluky, viditeľné na obr. 3.11 (b) kde je každému zhluku pridelená osobitná farba z dôvodu odlíšenia od ostatných zhlukov. V tomto kroku je vyčíslený stred zhluku, ktorý má pôsobiť ako identifikačný bod daného zhluku pri výbere vhodných kandidátov pre radiace body C a D.

Proces zhlukovania je využiteľný aj pri detekcii lebečnej časti, ktorá je z hľadiska porovnávania spoľahlivejšia, než metóda výberu susedov rovnakej intenzity. Ako je uvedené vyššie, je to ale časovo vyčerpávajúca metóda, ktorá by z hľadiska veľkosti lebečnej časti v obraze predĺžila výpočtový čas o niekoľko desiatok sekúnd.



Obr. č. 4.15, Zhlukovanie v diagrame

V prípade, ak je obraz nedostatočnej kvality, nesprávne identifikované hranice intenzity zhukov, alebo výberu takého obrazu, v ktorom sa nenachádzajú zadná mozgová komora (3DV) a čelná mozgová komora (FH) je proces clusteringu ukončený bez výsledku. To má za následok neschopnosť nájdenia riadiacich bodov v nasledujúcich metódach, pre nájdenie vhodných kandidátov riadiacich bodov C a D. V takomto prípade je program ukončený s chybovou hláškou o nevyhovujúcom vstupnom obraze.



Obr. 3.16, Obrazy po metóde zhlukovania: výsledný proces zhlukovania s centrálnym bodom zhuku v porovnaní s CSF mapou

```
# prechadza všetky pixely v obraze
for x in range(1, csf_image.shape[1] - 2):
    for y in range(1, csf_image.shape[0] - 2):
        # podľa zadaneho parametra sa pyta ci je pixel bielej intenzity
        if (pixelWhite(csf_image, x, y) == is_white):
            # a ci pixel patri do intenzity intervalu ktory bol prideleny funkcii v parametroch
            if index_min < csf_image[y, x] and csf_image[y, x] < index_max:
                # nasledne sa spusti metoda zhlukovania ktora je popisana nizsie
                res = clustering(x, y, csf_image, c_check_image, list_pixel_processed, clusters, index_min, index_max)

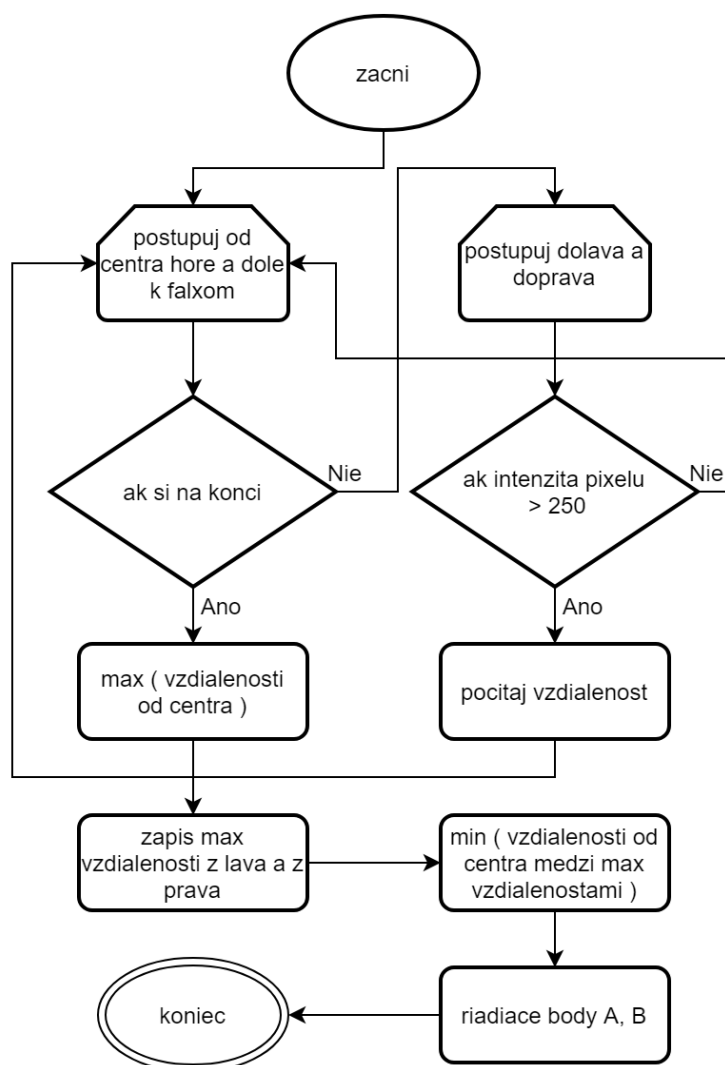
                list_pixel_processed = res[1]
                clusters = res[2]
```

Obr. č. 3.17, Fragment kódu zhlukovania

Každý pixel obrazu prejde kontrolou, či spĺňa predpoklady na to, aby mohol byť zhukom. V danom prípade je pixel predaný funkcii, ktorá vytvorí zhuk so všetkými jeho susedmi rovnakej intenzity (pozri obr. 3.17)

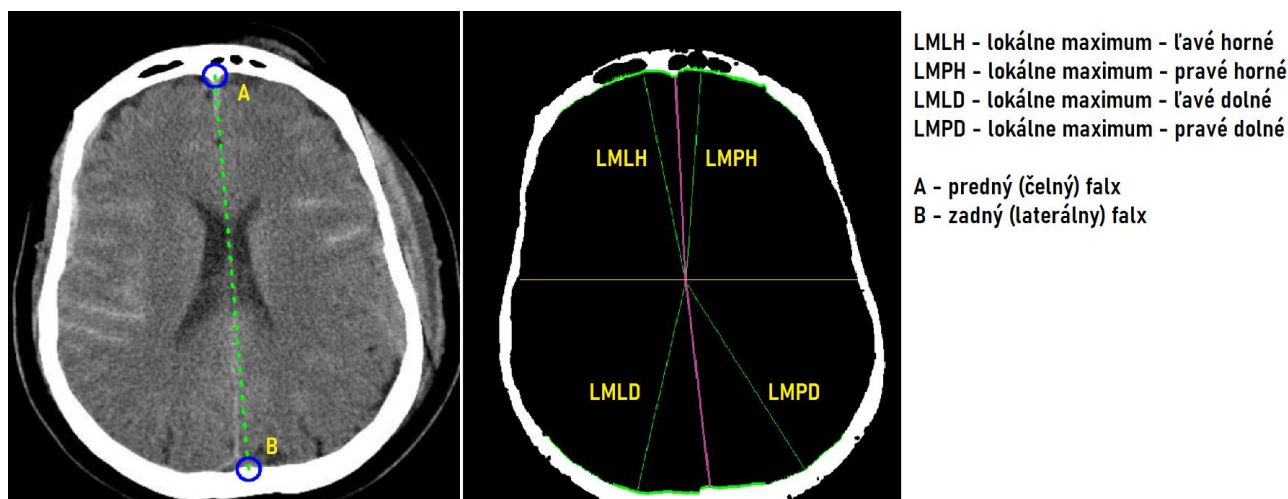
3.6 DETEKCIA RIADIACÍCH BODOV A, B

Vyhľadávanie riadiacich bodov A, B prebieha pre každý bod v osobitných funkciách, vzhľadom na pozíciu, v ktorých sa lokálne maximum nachádza. Ďalej využíva spoločnú funkciu pre nájdenie lokálneho minima, medzi vzdialenosťou definovanou lokálnymi maximami. Algoritmus prechádza od centrálnej vertikálnej osi smerom doľava a doprava, kým nenájde hranicu lebečnej časti obrazu. V tejto časti označí nájdený pixel hraničiaci s lebkou ako koncový bod úsečky, začínajúcej v centre lebky. Počíta sa vzdialenosť, pričom najväčšia vzdialenosť je uložená v pamäti pre neskoršie využitie. Súradnice nájdeného hraničného pixelu sú v ďalšej iterácii prehľadávania využité v podmienke, či daný pixel susedí s novým pixelom. To zaisťuje spoľahlivosť, ktorá by mohla byť narušená náhodným bielym pixelom vnútri mozgovej časti obrazu. Táto detekcia je optimalizovaná spôsobom štartujúcej pozície, ktorá namiesto začínania v priamom strede lebky začína v 70% vzdialenosti medzi stredom a falxom. Z vertikálneho prehľadávania v určitom bode sa algoritmus prekllopí na horizontálne prehľadávanie. Zmena je vytvorená kvôli tvaru, aký má falx. V prípade vertikálneho prehľadávania by hrozilo vynechanie určitých častí ktoré môžu byť z hľadiska vyhľadávania maxim kľúčové.



Obr. 3.18, Detekcia bodov A,B v diagrame

Algoritmus vyhľadávania lokálnych maxim končí, až sa stretne ukazateľ na prehľadávaný pixel z ľavej a pravej strany v jednom bode. Vyhodnotí sa najväčšia vzdialenosť z ľavej a pravej strany. Následne spustí algoritmus pre hľadanie najmenšej vzdialenosti v bodoch medzi ľavým a pravým maximom. To zabezpečí nájdenie takých bodov, ktoré považujeme za body umiestnené v čelnom a zadnom falxe. V tomto prípade platí, čím je obraz v najideálnejšej vertikálnej polohe a čím je lebka najideálnejšie symetrická, tým spoľahlivejší je tento algoritmus.

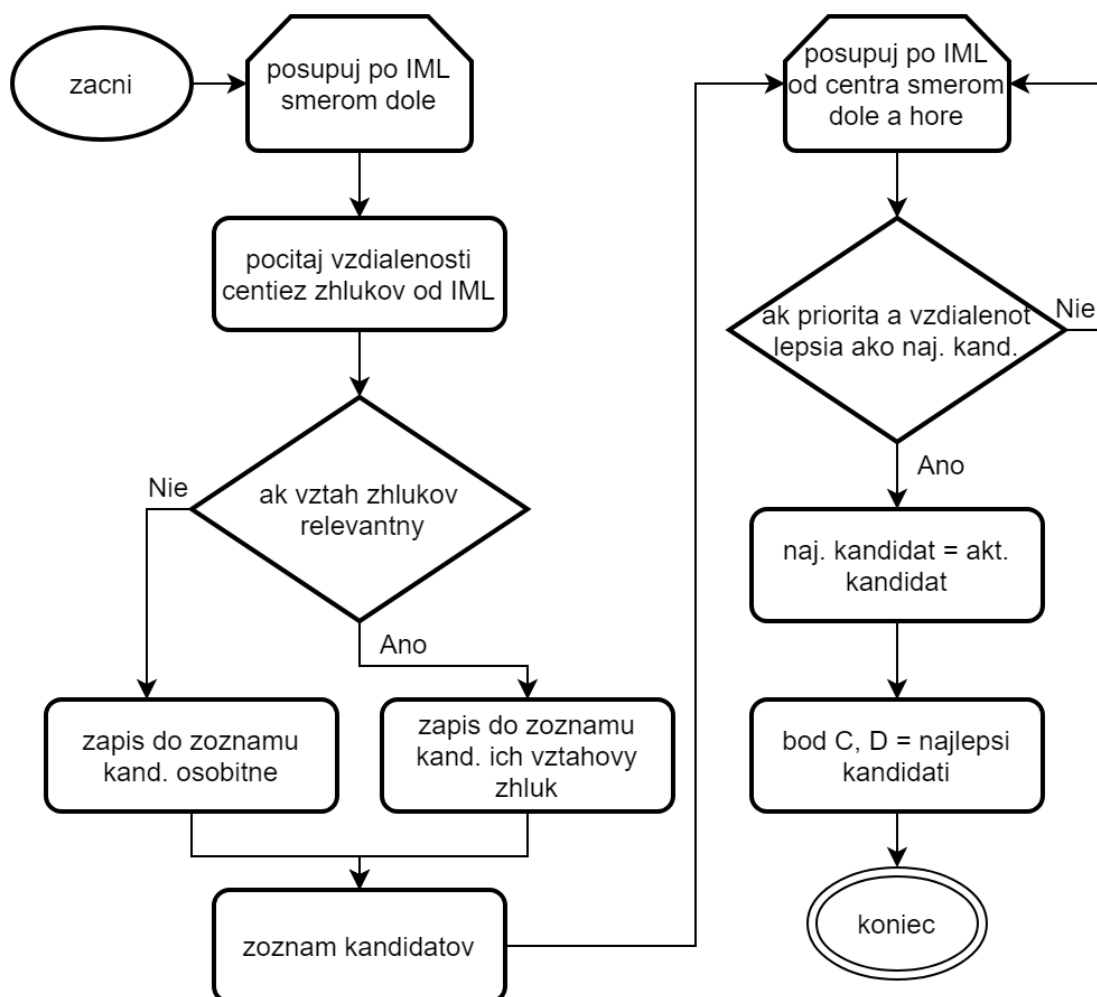


Obr. 3.19, Výpočet vzdialeností maxím a miním : ideálna úsečka v porovnaní s výpočtom lokálnych maxím, na základe ktorých je získaná IML

3.7 DETEKCIA RIADIACÍCH BODOV C, D

Spôsob akým sa vyhodnocujú riadiace body C a D, je založený na rozhodovacom strome. Pixely sú vyberané postupne tak, že algoritmus postupuje zhora nadol po ideálnej úsečke (IML), prechádza zoznam kandidátov na body C a D. V prípade, že sa v škálovanom okne nachádza kandidát pre riadiaci bod, je mu pridelená priorita. Ak je v škálovanom okne, vyčíslenom podľa miery vychýlenia IML viac kandidátov, je vytvorený nový kandidát, ktorý predstavuje stred dvoch zhlukov (pôvodných kandidátov). To predchádza dôslednej kontrole vzdialenosti, symetrie, výškovému rozdielu na základe vychýlenia IML a veľkosti zhlukov (kandidátov). Následne sa zoznam kandidátov upraví podľa predchádzajúcej filtrácie. Takto upravený zoznam ďalej prechádza kontrole, z ktorých sa vyberú najlepší kandidáti pre riadiace body C a D. Princíp výberu spočíva v prechádzaní CSF časti od stredu nahor, kde sa vyberie riadiaci bod C a od stredu nadol, následne výberu riadiaceho bodu D. Najvhodnejší kandidát je porovnávaný s prioritou a vzdialenosťou od IML. V prípade nenájdenia niektorého riadiaceho bodu je posun kvantifikovaný s jedným riadiacim bodom. V prípade nenájdenia žiadneho z riadiacich bodov C a D je kvantifikácia vyhodnotená ako neúspešná. To môže zapríčiniť nevhodná kvalita obrazu alebo nejasného zobrazenia zadnej mozgovej komory (3DV) a čelnej

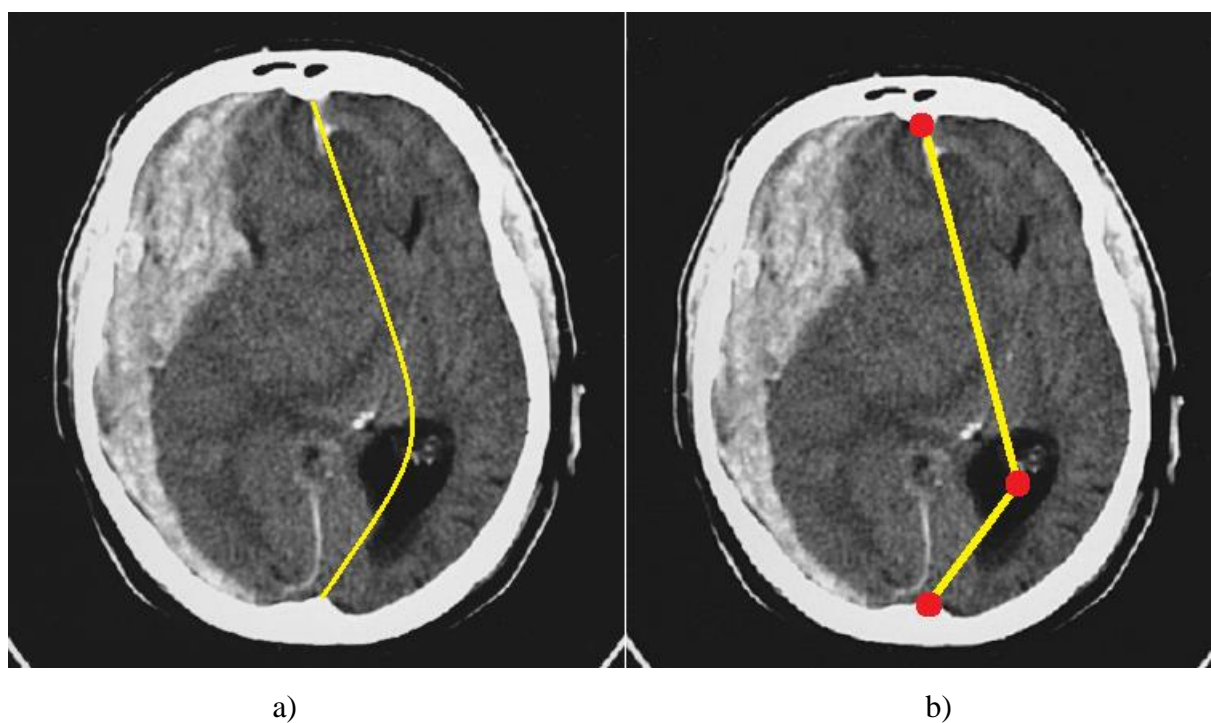
mozgovej komory (FH). Výslední kandidáti sú označení ako riadiace body, zapísané do objektu výsledných riadiacich bodov A, B, C, D.



Obr. 3.20, Detekcia bodov C,D v diagrame

4.8 KVANTIFIKÁCIA POSUNU

Na základe užívateľom zvolených vstupov, ako šírka lebky a spôsob prehl'adávania, sú v konzole vypísané výsledné vyhodnotenia veľkosti posunu. Patrí sem percentuálny posun v maximálnej vzdialenosti deformovanej krivky (DML) od ideálnej úsečky (IML). Ďalej veľkosť posunu v milimetroch, vyčíslený pomocou vzdialenosti DML od IML v pixeloch, šírka lebky (z vonkajších strán) v pixeloch a šírka lebky v mm (zadaný užívateľom, pôvodne stanovená šírka je svetový priemer 150mm). Aplikácia taktiež kalkuluje s vychýlením riadiacich bodov C a D do strán, v prípade nájdenia oboch riadiacich bodov. V prípade ich pozícií z rozličných strán IML je výsledný pomer hemisfér nemožné vyrátať. Obráz s IML a DML má usmerniť užívateľa k vlastnému empirickému posúdeniu o správnosti vyčíslenia, čo dodá na pravdivosti vyhodnotených výsledkov.



Obr. 3.21, Porovnanie výsledkov: (a) lekársky výsledok (b) výsledok aplikácie

V tejto kapitole sme si popísali princíp spracovania obrazu, rozdelenia na segmenty a detekciu kľúčových vlastností obrazu. Určili sme si riadiace body a naformovali ideálnu úsečku s deformovanou krivkou. Vyčíslili sme maximálnu vzdialenosť posunu a pomery plôch hemisfér. V ďalšej kapitole je tieto výsledky potrebné porovnať a určiť úspešnosť aplikácie.

4. EXPERIMENTOVANIE A VYHODNOTENIE VÝSLEDKOV

Aplikácia bola následne podrobená testovaniu úspešnosti vyčíslenia posunu. Testy odhalili slabiny aplikácie, ktoré bolo potrebné ošetriť tak, aby bola úspešnosť aplikácie vyššia. Boli nimi: konštantné odhadovanie hornej hranice intenzity a výber takých kandidátov v procese zhlukovania, ktorých veľkosť nebola obmedzená. Ďalej bola identifikovaná množina obrazov, ktoré aplikácia nie je schopná kvantifikovať z dôvodu nedostačujúcich vlastností obrazov, ako nízke rozlíšenie obrazu alebo výber takého obrazu CT, ktorý neobsahuje čelnú komoru (FH) a zadnú (3DV).

Do procesu experimentovania bol zahrnutých 80 vzoriek obrazov. Každý obraz bol vyčíslený dvoma spôsobmi, ktoré si vie užívateľ manuálne vybrať v grafickom plátne. Prvý, nazvaný QuickFinder (ktorý využíva metódu hľadania rovnakých susedných pixelov v metóde výberu lebečnej časti), druhý DeepFinder (využívajúci metódu zhlukovania pre výber lebečnej časti), ktorý v určitých prípadoch tenkej lebečnej časti (4.8%) je spoľahlivejší než QuickFinder. Celková úspešnosť aplikácie je 80.48%.

Obrazy, na ktorých bol úspešne vyčíslený posun, boli poskytnuté lekárovi, ktorý mal vyhodnotiť tieto obrazy nezávisle od výsledkov aplikácie. Konkrétne šlo o 11 obrazov. Rozdiely ktoré v každej vzorke vznikajú, sú spôsobené tým, že algoritmus vyčíslil stred zhuku presnejšie, na základe početnosti pixelov v zhuku, pričom lekár určuje stred zhuku intuitívne. To má za následok odchýlky, kde vznikajú milimetrové rozdiely.

Následujúca tabuľka porovnáva výsledný posun aplikácie a lekára. Keďže hodnotenie lekárom má za následok posun v milimetroch, to je dôvod k tomu, že každý výsledok je prirodzené číslo, v porovnaní s reálnym číslom vo výstupe aplikácie.

Tab. č. 3, Porovnanie výsledkov aplikácie a lekára

	výsledky [mm]		šírka lebky [mm]	rozdiel [mm]
	aplikácia	lekár		
v1	21.15	19	176	2.15
v2	18.08	13	150	5.08
v3	24.45	15	184	9.45
v4	30.27	16	160	14.27
v5	21.99	10	118	11.99
v6	16.76	15	198	1.76
v7	19.18	18	155	1.18
v8	7.94	10	129	2.06
v9	30.84	24	183	6.84
v10	15.05	7	155	8.05
v11	0.29	0	140	0.29
			priemer:	5.74

Výsledkom aplikácie je vyhodnotiť maximálnu vzdialenosť medzi ideálnou úsečkou (IML) a deformovanou krivkou (DML) a plochu medzi nimi. Z experimentov usudzujeme, že priemerná odchýlka výpočtu medzi ideálnou a deformovanou krivkou je 5.74mm (tab. 3).

Liao metóda – je založená na hľadaní symetrie medzi mozgovými štruktúrami mozgu. Zlyhanie nastáva v prípade identifikovania posunu aj vo vzorkách zdravého jedinca, kedy sú mozgové štruktúry v správnej polohe, ale nie v ideálnej symetrickej kompozícii [Liao2006]

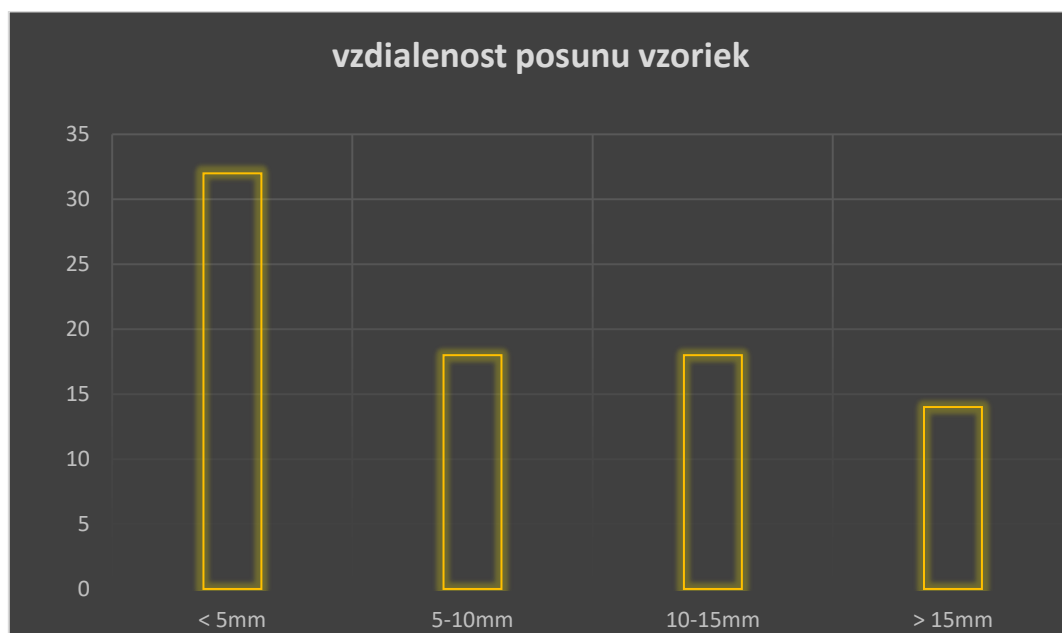
Chen metóda – je založená na porovnávaní zhody medzi tvarmi komôr. To nezahrňuje hľadanie falxov a teda veľkosť posunu je meraná od horizontálneho stredu lebky [Chen2010]

Tab. č. 4, Porovnanie úspešnosti vyčíslenia [Ruizhe2014]

Metódy	Maximálna vzdialenosť	
	priemer	smerodajná odch.
Liao metóda	14.07	16.67
Chen metóda	7.093	10.35
Ruizhe metóda	4.738	5.141
moja metóda	5.74	4.52

Z tabuľky 4. usudzujeme, že táto aplikácia je mierne horšia ako Ruizheho metóda. Podobnosť výsledkov sa dá odôvodniť tým, že táto aplikácia je navrhnutá na princípe anatomických značkových modelov rovnako, ako Ruizheho metóda.

Nasledujúci obrázok zobrazuje početnosť posunov z testovaných vzoriek. Je jasne viditeľné, že väčšina pacientov nemá žiaden posun, alebo posun do 5mm. Početnosť postupne klesá v závislosti od veľkosti posunu (pozri obr. 4.1)



Obr. 4.1, Vzďialenosť posunu z obrazov

ZÁVER

Posun v stredovej línii je osvedčeným indikátorom založeným na zobrazení, ktoré sú merateľné z CT obrazov. Štandardizácia merania uľahčuje porovnávanie medzi rôznymi hodnotiteľmi a umožňuje následnú automatizáciu. V práci je prezentovaný jednoduchý a efektívny spôsob detekcie posunu mozgových štruktúr, ktorá v podstate sleduje tmavé časti mozgu, vo vhodne vybranom reze CT. Tieto tmavé časti mozgu vo vhodne vybranom reze sú v podstate komory, ktoré slúžia ako indikátor pre stanovenie MLS. Práca objasňuje anatomický model značiek (riadiacich bodov A,B,C,D) a algoritmus využívajúci spôsob výberu kandidátov na automatické vyčíslenie posunu stredovej osi. Experimentálne výsledky poukazujú nato, že navrhovaný algoritmus efektívne vyhľadáva a kvantifikuje posun stredovej osi. V porovnaní s existujúcimi algoritmami, navrhovaný algoritmus znižuje chybu kvantifikácie. V ďalšom výskume sa plánuje rozšíriť súčasne algoritmy na základe ich sledovanej priamky, detekcie hematómu, rozšírenia o riadiace body E a F a navrhovaným kvantifikačným algoritmom. Taktiež sa plánuje ďalej študovať vzťahy medzi traumatom a posunom v stredovej osi. Tieto výsledky je vhodné ďalej integrovať do ďalších metadát, ako sú pohlavie, vek a ďalšie doplnujúce údaje o pacientoch. Zo získaných údajov o pacientoch, vytvorenia systému pre ich zhromažďovanie a na základe integrácie je možné vykonávať štatistické dolovanie dát. Pri výskume klinickej prognózy sa očakáva, že navrhovaný algoritmus a systém uľahčia klinický výskum diagnostiky a prognózy CT mozgu.

ZOZNAM POUŽITEJ LITERATÚRY

- [Byass2010] Byass P, de Courten M, Graham WJ et al. Reflections on the global burden of disease. *Epub PLoS. Med.* 10, 1477, 2010
- [Boone2007] Boone JM. Radiological interpretation 2020: toward quantitative image assessment. *Medical Physics* 2007;34(11):4173–9.
- [Rincon2019] Rincon-Guio C. The role of computed tomography as a prognostic tool in traumatic brain Trauma, 2019
- [Chen2010] Chen W, Najarian K, Ward K. Actual midline estimation from brain CT scan using multiple regions shape matching. *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2010:2552–5
- [Duda1972] Duda RO, Hart PE. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM* 1972;15(1):11–5. 1972
- [Gong2007] Gong T, Liu R, Tan C, Farzad N, Lee C, Pang B, et al. Classification of CT brain images of head trauma, *vol 4774 of Lecture Notes in Computer Science* 2007:401–8.
- [Harris2009] Harris OA, Muh CR, Surles MC, Pan Y, Rozycki G, Macleod J, Easley K. Discrete cerebral hypothermia in the management of traumatic brain injury: a randomized controlled trial, *110(6):1256-64*. Jun 2009
- [Wang2017] Wang HC, Ho SH, Furen X, Chou JH. A Simple Fast and Fully Automated Approach for Midline Shift Measurement on Brain Computed Tomography, 2017
- [Kazdal2015] Kazdal H, Kanat A, Findik H, Sen A, Ozdemir B, Batcik OE, Yavasi O, Inecikli MF. Transorbital Ultrasonographic Measurement of Optic Nerve Sheath Diameter for Intracranial Midline Shift in Patients with Head Trauma, *Epub 2015 Oct 17. PubMed PMID: 26485420*.
- [Liao2006] Liao CC, Chiang IJ, Xiao F, Wong JM., Tracing the deformed midline on brain CT. *Biomedical Engineering: Applications, Basis, and Communications* 2006;18:305–11.
- [Liu2006] Liu WG, Qiu WS, Zhang Y, Wang WM, Lu F, Yang, Effects of selective brain cooling in patients with severe traumatic brain injury: a preliminary study, *J Int Med Res.* 2006;34(1):58-64. 207
- [Qiu2006] Qiu W, Shen H, Zhang Y, Wang W, Liu W, Jiang Q, Luo M, Manou M, Non invasive selective brain cooling by head and neck cooling is protective in severe traumatic brain injury, *J Clin Neurosci.* 2006;13(10):995-1000.

- [Quattrocchi1991] Quattrocchi KB, Prasad P, Willitsm NH, Wagner FC. Quantification of midlineshift as a predictor of poor outcome following head injury. *Surgical Neurology*1991;35:183–8.
- [Ramachandran1980] Ramachandran K. A coding method for vector representation of engineeringdrawings. *Proceedings of the IEEE* 1980;68(7):813–7.
- [Ropper1986] Ropper AH. Lateral displacement of the brain and level of consciousness in patients with an acute hemispherical mass. *N Engl J Med.* 1986 Apr. *PubMed PMID:* 3960059.
- [Ruizhe2014] Ruizhe L, Shimiao L, Chew LT, Tze-Yun L, Boon CP, Tchoyoson L, Cheng K LeecaSchool. Automatic detection and quantification of brain midline shift usinganatomical marker model. *MOE Grant MOE2014-T2-2-146 (R252-000-480-112)*
- [Segun2021] Segun, Toyin, Dawodu. Traumatic Brain Injury (TBI). Definition, Epidemiology, Pathophysiology. 2021
- [Silver2005] Silver JM, Mcallister TW, Yodufsky SC. Textbook of traumatic brain injury. *Washington, DC: American Psychiatric Association;* 2005
- [Sullivan2008] Sullivan DC. Imaging as a quantitative science. *Radiology* 2008;248(2):328–32
- [Deserno2011] Deserno TM. Fundamentals of Biomedical Image Processing; 2011
- [Tromberg2019] Tromberg B, Leapman R. NIBIB-funded projects using computed tomography; 2019
- [Uribe2013] Uribe A., Gomez F., Munoz N., Bernal G. Diagnosis and treatment of adult patients with severe Traumatic Brain Injury; 2013
- [Zheng2002] Zheng Y, Liu C, Ding X, Pan S. A form frame-line detection algorithm based ondirectional single-connected chain, *Journal of Software* 2002;13:790–6.
- [web1] <https://insights.stackoverflow.com/survey/2020#most-popular-technologies>
- [web2] <https://reubenrochesingh.medium.com/comparison-of-10-programming-languages-f43b0ac337a4>
- [web3] <https://qvault.io/javascript/top-web-development-programming-languages-a-comparison>

ZOZNAM PRÍLOH

Obsah CD

- Priložené CD obsahuje zdrojové súbory projektu
- Preložený spustiteľný súbor MLS-detector.exe