



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY



Bakalářská práce

Filtry ve zpracování medicínských obrazových dat

Petra Ocelíková





**FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI**

**KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY**

Bakalářská práce

Filtry ve zpracování medicínských obrazových dat

Petra Ocelíková

Vedoucí práce

Ing. Martin Kryl

© Petra Ocelíková, 2023.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

Citace v seznamu literatury:

OCELÍKOVÁ, Petra. *Filtry ve zpracování medicínských obrazových dat*. Plzeň, 2023. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Ing. Martin Kryl.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Petra OCELÍKOVÁ**
Osobní číslo: **A20B0550P**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační systémy**
Téma práce: **Filtry ve zpracování medicínských obrazových dat**
Zadávací katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s principem digitálních obrazových filtrů.
2. Vyhleďte a souhrnně popište běžné způsoby uplatnění filtrů při zpracování CT snímků mozku.
3. Navrhněte nástroj pro dávkové zpracování sady obrazů dle uživatelem zadané konfigurace.
4. Implementujte nástroj v jazyce Python za pomoci knihovny SimpleITK.
5. Ověřte funkčnost na vybrané sadě medicínských snímků a diskutujte kvalitu řešení.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce

Vedoucí bakalářské práce: **Ing. Martin Kryl**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **3. října 2022**
Termín odevzdání bakalářské práce: **4. května 2023**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 25. října 2022

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 1. května 2023

.....

Petra Ocelíková

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Abstrakt

Cílem této práce je seznámit se s digitálními obrazovými filtry a jejich uplatněním při zpracování CT snímků. Nejprve je stručně představen obecný postup při zpracování medicínských obrazových dat. Poté jsou popsány jednotlivé metody jejich pořízení. Dalším tématem jsou pak samotné obrazové filtry, které jsou využívány při předzpracování a segmentaci medicínských obrazových dat. V praktické části je navržen a implementován vlastní nástroj pro dávkové zpracování sady medicínských snímků využívající knihovnu SimpleITK.

Abstract

The aim of this thesis is to become familiar with digital image filters and their application in the processing of CT scans. Firstly, operations of medical image processing are briefly introduced. Then, individual methods of medical image acquisition are described. The next topic is the actual image filters used in preprocessing and segmentation of medical images. In the practical part, a tool for batch processing of medical image sets is designed and implemented using the SimpleITK library.

Klíčová slova

zpracování obrazu • medicínská data • obrazové filtry • segmentace • SimpleITK

Poděkování

V první řadě bych chtěla poděkovat vedoucímu mé bakalářské práce, Ing. Martinu Krylovi, za jeho ochotu, za poskytnutí užitečných rad, nápadů a materiálů k mé práci a za opravdu důkladné čtení průběžných verzí práce a jejich kontrolu.

Zároveň bych také ráda poděkovala své rodině, především rodičům, za poskytnutí ubytování, stravy a finanční, materiální i psychické podpory v průběhu celého mého dosavadního studia.

Obsah

1	Úvod	3
2	Zpracování medicínských obrazových dat	5
3	Metody pořízení medicínských obrazových dat	7
3.1	Výpočetní tomografie	7
3.2	Rentgen	10
3.3	Magnetická rezonance	11
3.4	Sonografie	12
4	Uplatnění filtrů při předzpracování CT snímků	15
4.1	Úprava jasu	16
4.2	Úprava kontrastu	17
4.3	Úprava histogramu	17
4.4	Pseudobarva	19
4.5	Geometrické transformace	19
4.5.1	Transformace souřadnic	20
4.5.2	Interpolace	21
4.6	Filtrace	22
4.6.1	Konvoluce	22
4.6.2	Vyhlazovací filtry	23
4.6.3	Hranové filtry	25
5	Uplatnění filtrů při segmentaci CT snímků	27
5.1	Windowing	27
5.2	Segmentace prahováním	28
5.3	Segmentace založená na regionech	29
5.3.1	Metoda dělení a spojování oblastí	30
5.3.2	Metoda narůstání oblastí	31
5.4	Segmentace založená na hranách	31
5.4.1	Metoda sledování okrajů	31

6	Návrh nástroje pro zpracování medicínských snímků	33
6.1	Analýza a návrh	33
6.1.1	Existující nástroje pro zpracování obrazu	33
6.1.2	Návrh řešení	35
6.2	Implementace	36
6.2.1	Implementované funkce	38
6.3	Ukázky	41
6.4	Zhodnocení výsledků	45
7	Závěr	47
A	Uživatelská příručka	49
A.1	Požadavky a spuštění programu	49
A.2	Seznam funkcí a jejich parametrů	52
A.3	Podporované formáty dat	53
B	Struktura ZIP souboru	55
	Bibliografie	57
	Seznam obrázků	61
	Seznam tabulek	63
	Seznam výpisů	65

Zobrazování medicínských dat se stalo velice důležitým nástrojem moderní medicíny, který umožňuje lékařům a dalším zdravotnickým pracovníkům diagnostikovat a léčit širokou škálu nemocí. Použití zobrazovacích technik, jako je například rentgen, výpočetní tomografie, sonografie či magnetická rezonance, výrazně zlepšilo schopnost vizualizovat a porozumět základním strukturám a funkcím v lidském těle. Avšak velké množství dat generovaných těmito zobrazovacími metodami může být někdy obtížně analyzovatelné, takže bylo nutné vyvinout algoritmy a techniky pro automatické zpracování těchto dat a usnadnění jejich interpretace.

Jednou z takových technik je použití obrazových filtrů. Filtry jsou obecně matematické algoritmy, které lze aplikovat na digitální obrazová data za účelem vylepšení jejich vlastností nebo odstranění nežádoucích artefaktů. V oblasti zpracování medicínských obrazových dat slouží filtry k extrahování užitečných informací z medicínských snímků a ke zlepšení jejich kvality pro další analýzu. Lze je použít také ke zvýšení viditelnosti důležitých struktur či abnormalit ve snímcích, což usnadňuje jejich následnou interpretaci a diagnostiku.

Tato práce si klade za cíl prozkoumat proces zpracování medicínských obrazových dat a zejména roli, kterou v něm mají obrazové filtry. Tato práce poskytuje ucelený přehled o využití obrazových filtrů při předzpracování a segmentaci medicínských obrazových dat. Nejprve jsou představeny jednotlivé metody zobrazovacích technik a následně jsou rozebrány a prozkoumány obrazové filtry, jejich princip, limitace a uplatnění v různých fázích zpracování medicínských snímků. V rámci praktické části této práce je nejprve provedena stručná analýza existujících nástrojů pro zpracování obrazových dat a poté navržen a implementován nástroj pro dávkové zpracování sady medicínských dat dle uživatelem zadané konfigurace. Nástroj je implementován v jazyce Python s využitím knihovny pro zpracování a analýzu obrazu SimpleITK. Na závěr je ověřena funkčnost a kvalita navrženého nástroje na sadě reálných medicínských dat.

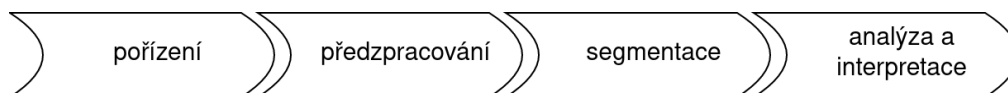
Zpracování medicínských obrazových dat

2

Zpracováním medicínských obrazových dat se míní posloupnost operací od samotného pořízení snímku, přes předzpracování a segmentaci, až po analýzu a interpretaci výsledků (Obrázek 2.1).

První operací při zpracování je pořízení obrazu, které spočívá v nasnímání obrazu a jeho následné digitalizaci. Ke snímání obrazu slouží více různých zobrazovacích metod, které jsou podrobněji rozebrány v kapitole 3. Obraz se pak zpracuje do digitální formy. Tento digitální obraz však může být z různých důvodů zkreslený, například nevhodným způsobem snímání či vlivem použité technologie. Aby se pak nad výsledným obrazem snáze prováděly analýzy, obraz se předzpracuje, a to především odstraněním šumu, korekcí jasu, kontrastu, apod. Metody předzpracování jsou detailně popsány v kapitole 4. Další operací v procesu zpracování je segmentace. Tou se rozumí vyčlenění oblastí zájmu v obraze. K tomu slouží metody jako např. prahování, detekce hran, apod., podrobněji popsány v kapitole 5.

Posledními operacemi jsou analýza a interpretace výsledků. Výstupem analýzy, resp. popisem objektů nalezených v předchozím kroku (tedy segmentaci), je například tabulka objektů s hodnotami tvarových charakteristik, plochy, obvodu, apod. Interpretace výsledků či porozumění obrazu je pak postup, který popsané objekty klasifikuje do skupin podle společných příznaků. K tomu se často používají metody umělé inteligence (např. neuronové sítě). [Bry19]



Obrázek 2.1: Posloupnost operací zpracování medicínských obrazových dat

2. Zpracování medicínských obrazových dat

Jak již bylo zmíněno výše, následující kapitoly budou podrobněji pojednávat o prvních třech operacích při zpracování medicínských obrazových dat (pořízení, předzpracování a segmentaci obrazu) a o obrazových filtrech, které se při těchto operacích obvykle uplatňují.

Metody pořízení medicínských obrazových dat

3

Pořízení medicínských obrazových dat, spočívající v jejich nasnímání a následné digitalizaci, je prvním důležitým krokem v procesu jejich zpracování. Metod, jak medicínská obrazová data pořídit, existuje více a jsou založeny na různých principech. Mezi nejběžnější a nejčastěji používané metody patří rentgen, výpočetní tomografie, magnetická rezonance a sonografie [uzi20].

Tato práce se dále zabývá filtry ve zpracování snímků konkrétně z výpočetní tomografie, proto je v této kapitole nejdetailněji rozebrána právě tato metoda.

3.1 Výpočetní tomografie

Výpočetní tomografie (angl. Computed Tomography, CT) je neinvazivní metoda využívaná v medicíně k diagnostice různých druhů onemocnění. Jedná se o radiologickou vyšetřovací metodu, která slouží k přesnému a věrnému zobrazení struktur v těle. Pomocí rentgenového záření umožňuje výpočetní tomografie detailní zobrazení struktur tkáně, vnitřních orgánů a kostí a jejich posouzení. Využívá se zejména při podezření na chronická onemocnění či akutní stavy, jako jsou cévní mozkové příhody, nitrolební krvácení, choroby tepen či zlomeniny kostí. [Ho22a]

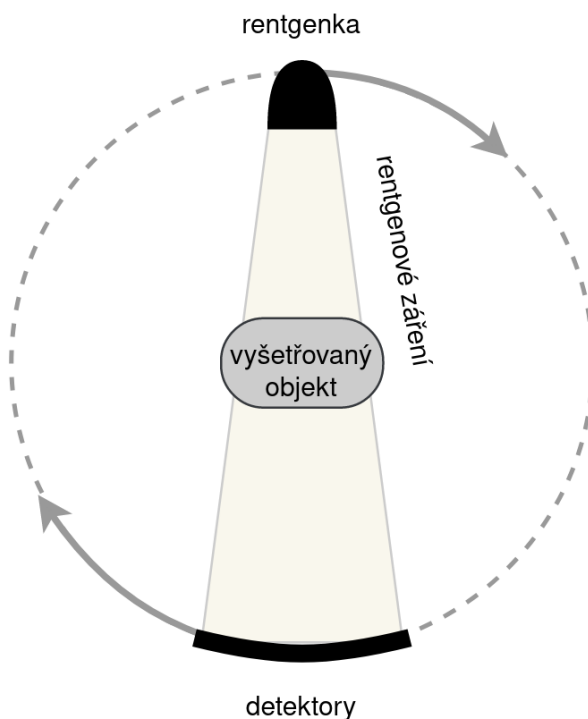
Pro snadnější odlišení vyšetřované oblasti od ostatních struktur v těle se může použít jodová kontrastní látka, která se nitrožilně aplikuje do těla před provedením CT vyšetření. Vyšetření s kontrastní látkou má však kontraindikace - může u pacienta například způsobit těžké alergické reakce. Také se vyšetření s jodovou kontrastní látkou nesmí provádět v těhotenství, nebo pokud trpí pacient poruchou funkce ledvin či štítné žlázy. Nativní vyšetření (tzn. bez kontrastní látky) se může provádět prakticky kdykoliv, relativní kontraindikací je v tomto případě pouze těhotenství. [Ho22a]

Princip výpočetní tomografie

Základním kamenem pro vývoj výpočetní tomografie byl objev rentgenového záření uskutečněný W. C. Röntgenem v roce 1895. Rentgenové záření je částečně absorbováno v objektu, kterým prochází, kdy množství absorbovaného záření závisí na složení onoho objektu. Intenzita propuštěného rentgenového záření se řídí rovnicí

$$I = I_0 e^{-ax}, \quad (1)$$

kde I je intenzita propuštěného záření, I_0 je intenzita záření před průchodem objektem, x je tloušťka objektu a a je lineární koeficient zeslabení záření pro konkrétní materiál. [UME96]



Obrázek 3.1: Princip výpočetní tomografie

Výpočetní tomografie schopnosti absorpce rentgenového záření využívá. Funguje na principu rotace rentgenky a detektorů kolem vyšetřovaného objektu (těla). Rentgenka a detektory jsou umístěny na kruhovém objektu zvaném gantry. Termínem rentgenka se označuje speciální dioda sloužící k produkci rentgenového záření. To prochází tělem, kde se jeho část absorbuje. Jak již bylo zmíněno výše, absorbované množství záření se liší podle hustoty a složení struktur v těle, tedy např. kosti absorbují více záření nežli měkké tkáně. Detektory umístěné na gantry na opačné straně od rentgenky neabsorbovanou část záření zachytí, transformují na signál

a ten odešlou do počítače. Po každé rotaci a nasnímání jednoho řezu se pohyblivé lůžko, na kterém leží vyšetřovaný objekt, mírně posune a proces se opakuje. Tímto způsobem lze získat trojrozměrný obraz složený z více snímků (řezů). Jednotlivé řezy jsou rozmístěny rovnoměrně a vzdálenost mezi nimi se nazývá tloušťka řezu. Čím menší tloušťka řezu je, tím více detailů je zachyceno. Rotace gantry zároveň umožňuje zachytit vyšetřovaný objekt z různých úhlů a při odlišné síle rentgenového záření, čímž výrazně zvyšuje přesnost zobrazení. [Lep19]

Rekonstrukce obrazu

Počítač dostane od detektorů na tomografu signály a jeho úkolem je tyto signály zpracovat a zrekonstruovat snímek jednoho řezu, který bude tvořen maticí voxelů (voxel = částice objemu, hodnota v mřížce 3D prostoru). Výsledek výpočtu spočívá ve vyřešení velké soustavy rovnic závislé na rozlišení snímků (počtu voxelů) a přiřazení skutečného koeficientu zeslabení každému z voxelů. Toho se dosáhne pomocí inverzní Radonovy transformace. Ta je detailně matematicky popsána ve zdrojích [Lep19; VS08].

Přímé použití vzorce inverzní Radonovy transformace však není z hlediska náročnosti numerického výpočtu vhodné. Proto se využívá alternativních postupů výpočtu, například metod založených na Fourierově řezovém teorému, nebo algebraických metod řešení úlohy $Rx = b$, kde R je aproximace diskrétní Radonovy transformace, x je zrekonstruovaný obraz a b jsou data z tomografu. [VS08] Komplexní popis inverzní Radonovy transformace a těchto metod však přesahuje rámec této práce.

Výsledné zrekonstruované snímky řezů jsou tedy uloženy jako matice voxelů. Pro každý voxel v matici je známa hodnota absorbovaného rentgenového záření označovaná jako denzita. Ta se udává v Haunsfieldových jednotkách (HU). [Lep19] Následující tabulka ukazuje přibližné hodnoty denzit vybraných struktur lidského těla.

Struktura	Hodnota denzity (HU)
vzduch	-1000
voda	0
měkké tkáně	25 až 70
krev	40
sražená krev	65 až 85
kosti	90 a více

Tabulka 3.1: Přehled denzit vybraných struktur [Lep19]

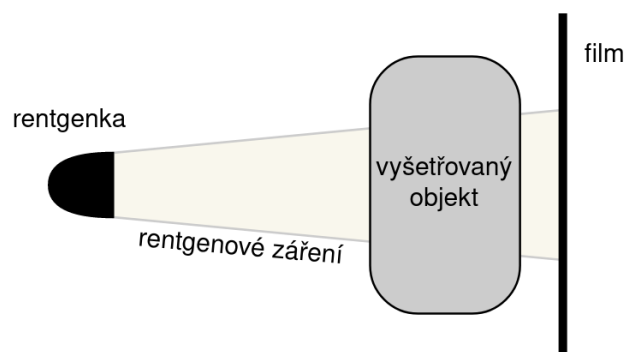
Denzita se obvykle vykresluje ve stupních šedi, nicméně na klasických monitorech je nemožné vykreslit celou stupnici, jelikož jsou schopné zobrazit pouze hloubku šedé barvy v rozmezí 0 až 255, což ani zdaleka nedosahuje možného rozsahu hodnot denzit. Řešením je například použití speciálních diagnostických černobílých monitorů, které dokážou zobrazit výrazně větší škálu šedých odstínů. Nicméně tolik odstínů šedé nelze lidským okem rozlišit. Proto se při analýze obrazu využívá zobrazení jen zvoleného „okna“ (intervalu) z celého rozsahu. V tomto případě se voxelům s hodnotou denzity nižší nebo stejnou jako spodní hranice zvoleného intervalu přiřadí černá barva a voxelům s hodnotou denzity vyšší než horní hranice intervalu se přiřadí barva bílá. Voxelům uvnitř intervalu je přiřazen konkrétní stupeň šedé barvy na základě jejich relativní pozice v něm. [Lep19]

3.2 Rentgen

Rentgen (RTG) je základní radiologická neinvazivní diagnostická metoda. Využívá se k zobrazení struktur v těle a následné diagnostice případných onemocnění. Pokud se při vyšetření rentgenem zjistí podezření na nějaké onemocnění a je tudíž potřeba přesnější a věrnější zobrazení zasažené oblasti, je obvykle pacient ještě poslán na vyšetření výpočetní tomografií nebo magnetickou rezonancí. Rentgen totiž nedokáže zachytit tolik detailů.

Princip rentgenu

Funguje na stejném principu jako výpočetní tomografie, ale s tím rozdílem, že rentgenka a film, který zachycuje rentgenové záření, se kolem vyšetřovaného objektu netočí (Obrázek 3.2). Výsledkem je tedy jen dvourozměrný snímek. Digitalizace obrazu probíhá taktéž obdobně jako u výpočetní tomografie, ale pouze pro jeden „řez“.



Obrázek 3.2: Princip rentgenu

3.3 Magnetická rezonance

Magnetická rezonance (MR/MRI) je neinvazivní diagnostická zobrazovací metoda, která je používána ve zdravotnictví k zobrazování jednotlivých tkání lidského těla. Na rozdíl od výpočetní tomografie či rentgenu nevyužívá rentgenového záření, ale magnetických vlastností atomů vodíku ovlivňovaných magnetickým polem přístroje. Výhodou magnetické rezonance je vysoký tkáňový kontrast, tudíž více rozlišuje jednotlivé tkáně, včetně různých struktur mozku. Tato metoda má ale také kontraindikace. Absolutní kontraindikací, kdy pacient nesmí být vyšetřen magnetickou rezonancí, jsou cizí či implantované kovové materiály, jako například kardiostimulátor, neurostimulátor či kochleární implantát. Relativními kontraindikacemi jsou kovové implantáty a protézy, těhotenství, kojení či klaustrofobie. [Ho22b]

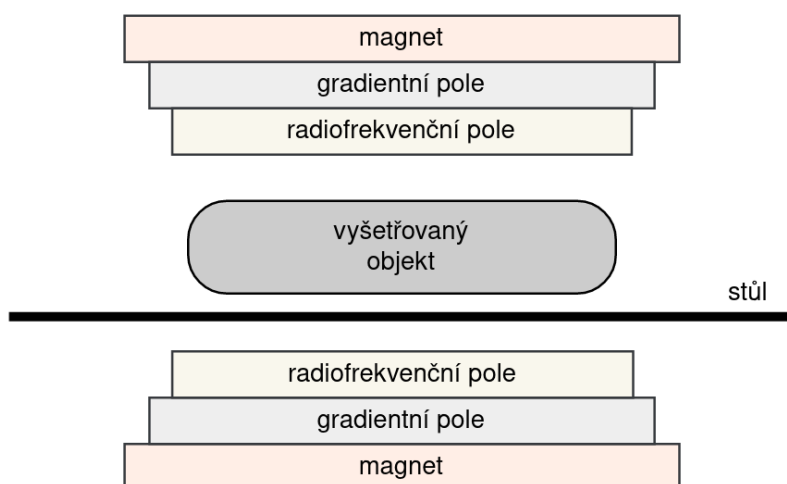
Princip magnetické rezonance

Princip magnetické rezonance spočívá v tom, že se protony v jádrech atomů v tkáních, které jsou jinak orientovány nahodile, v přítomnosti vnějšího magnetického pole uspořádají paralelně či antiparalelně s vektorem magnetického pole MR magnetu a vykonávají rotační pohyb kolem své osy. Vlastní rotační osa protonu vykonává ještě pohyb po plášti kužele, čemuž se říká precese. Frekvence tohoto precesního pohybu, tzv. Larmorova frekvence, závisí na magnetických vlastnostech daného atomového jádra a na intenzitě vnějšího magnetického pole. [Ben12; Toe17]

Různé tkáně mají různou tkáňovou magnetizaci. Pro změření velikosti této tkáňové magnetizace je do tkáně vyslán elektromagnetický impuls. Po skončení působení tohoto impulsu se protony začnou vracet do původního stavu. Zároveň klesá intenzita měřitelného signálu z tkáně, který dosahuje maxima právě těsně po skončení elektromagnetického impulsu. Aplikací gradientních polí [Toe17] se pak získá konkrétní pozice $[x, y, z]$ (voxel), ze které přichází měřený signál. Pozice každého voxelu je charakterizována jeho fází a frekvencí. [Ben12]

Rekonstrukce obrazu

Při rekonstrukci obrazu se z digitalizovaných MR signálů vypočítá pomocí algoritmu Fourierovy transformace jednotlivým voxelům hodnota šedého odstínu (0 až 255). Výsledkem je 2D nebo 3D obraz podobný jako při vyšetření výpočetní tomografií. [Ben12; Toe17]



Obrázek 3.3: Schéma magnetické rezonance

Princip magnetické rezonance není v této práci rozebrán příliš do detailů. Více informací o fyzikálním principu MR, její konstrukce a digitalizaci obrazu lze nalézt ve zdroji [Ben12].

3.4 Sonografie

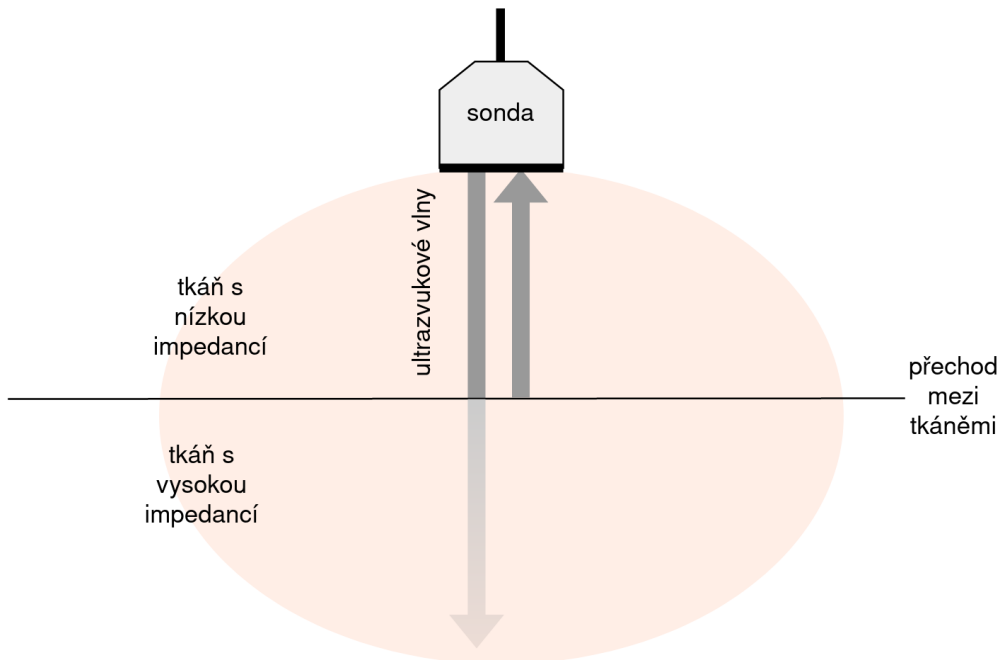
Sonografie (nebo také ultrazvuk) je neinvazivní diagnostická metoda, jejíž princip spočívá v zachytávání odrazového signálu tkáně po vysílání ultrazvukových vln. Ty jsou do těla vysílány pomocí speciální ultrazvukové sondy. Pro usnadnění přenosu těchto vln se na sondu aplikuje speciální gel. Ultrazvukové vlny se uvnitř těla odráží od tkání a orgánů, sondou jsou zpět přijímány a zpracovávají na dvourozměrný šedotónový obraz. [Ho22c; Buč21]

Sonografie je poměrně levná, efektivní a bezpečná metoda. Její velkou výhodou je, že se provádí v reálném čase a umožňuje tak zobrazit pohyby zobrazovaných orgánů a struktur. Je vhodnou metodou pro zobrazení orgánů v dutině břišní a měkkých tkání. Naopak pro zobrazení porézních tkání (kostí a plic), je zcela nevhodná, jelikož tyto tkáně vyslané ultrazvukové vlny téměř úplně ztlumí a vytvoří tak akustický stín. Kontraindikace sonografického vyšetření prakticky neexistují, nejsou známa ani žádná rizika. [Ho22c]

Princip sonografie

Zdrojem ultrazvukového vlnění jsou na sondě umístěné destičky z vhodného krystalického materiálu (např. křemene). Tyto destičky se vlivem periodického nabíjení

elektrickým proudem deformují (smršťují a rozpínají), čímž vzniká mechanické vlnění zvané piezoelektrický jev. To generuje ultrazvukové vlny, které pak procházejí tělem a odrážejí se od přechodů mezi jednotlivými tkáněmi s různou akustickou impedancí (prostupností ultrazvuku). [Buč21]



Obrázek 3.4: Princip sonografie

Klasická sonografie umožňuje takto získat obraz statických tkání. S využitím Dopplerova jevu lze však získat informaci o rychlostech pohybu tkání, a lze s ním tedy provádět i vyšetření cévního systému. Dopplerův jev popisuje změnu frekvence a vlnové délky vyslaného signálu vůči přijatému (odraženému od pohybujících se objektů). Ze změny frekvence vlnění se určí rychlost a směr pohybu objektu. Pokud se objekt přibližuje, frekvence přijatého vlnění je vyšší a pokud se objekt vzdaluje, je frekvence přijatého vlnění naopak nižší. [Buč21]

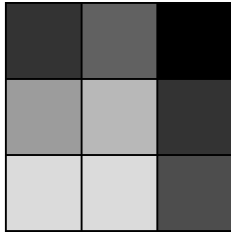
Uplatnění filtrů při předzpracování CT snímků

4

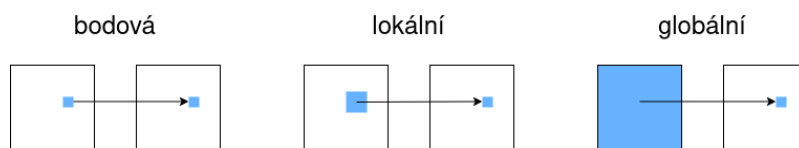
Předzpracování je velice důležitá a nezbytná operace při zpracování medicínských obrazových dat. A to z toho důvodu, že obraz pořízený tomografem (nebo obecně jakoukoliv zobrazovací metodou) se skoro nikdy nepodaří získat bez jakéhokoliv rušení, šumu a za optimálních světelných podmínek. Kromě technických problémů je častou imperfekcí i to, že pacient nemůže být nehybný po celou dobu vyšetření (dýchání, činnost srdce, ...). Předzpracování slouží k odstranění těchto vad z obrazu a umožňuje tak snazší a efektivnější analýzu výsledného obrazu.

Mezi základní metody předzpracování patří například úpravy jasu, kontrastu, histogramu a geometrické transformace. Většina úprav spočívá v jednoduchých aritmetických operacích nad snímkem. Dvourozměrný snímek je vlastně matice, jejímiž prvky jsou hodnoty pixelů (obvykle v rozmezí 0 až 255) (Obrázek 4.1). Nad touto maticí lze pak provádět operace a tak snímek upravovat. Jasové a kontrastové operace mohou být bodové (hodnota výsledného pixelu se vypočítá z hodnoty jednoho pixelu), lokální (hodnota výsledného pixelu se vypočítá z lokálního okolí pixelu) a globální (hodnota výsledného pixelu je vypočtena z celého snímku) (Obrázek 4.2). [Bry19; Hoz03]

45	100	0
144	190	45
220	220	85



Obrázek 4.1: Matice pixelů



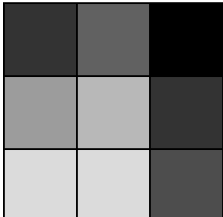
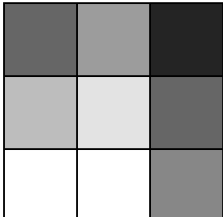
Obrázek 4.2: Druhy operací

Při předzpracování obrazových dat se hojně uplatňují filtry. Filtrace funguje na principu modifikace hodnot pixelů s ohledem na nejbližší okolí. K úpravě se nejčastěji používá čtvercová, tzv. filtrační matice. Často využívanými filtry jsou filtry konvoluční, vyhlazovací či hranové. [OOM02]

V této kapitole jsou popsány základní principy vybraných operací a filtrů, které se při předzpracování běžně používají.

4.1 Úprava jasu

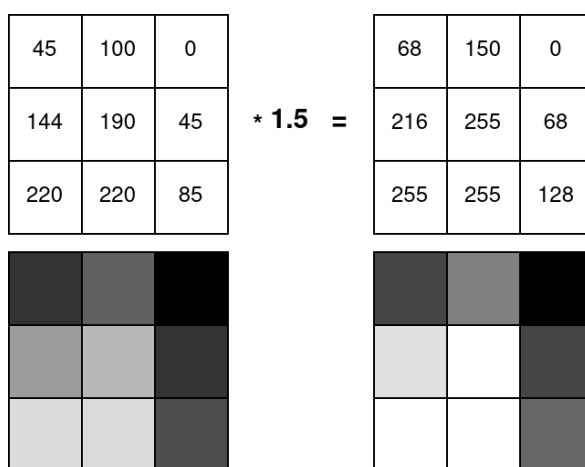
Korekce jasu je jednou z nejzákladnějších metod předzpracování snímku. Využívají se při ní aritmetické operace sčítání a odčítání. Pokud chceme u příliš tmavého snímku zvýšit jas, přičteme ke každému pixelu v matici konstantu. Pokud chceme naopak přezářený snímek ztmavit, konstantu od všech pixelů odečteme. Při těchto operacích může samozřejmě dojít k přetečení/podtečení hodnot z rozsahu 0 až 255, což se nejčastěji řeší omezením rozsahu na 0 při podtečení (např. $45 - 50 = -5 \rightarrow 0$) a omezením rozsahu na 255 při přetečení (např. $220 + 50 = 270 \rightarrow 255$). [Hoz03]

45	100	0	+ 50 =	95	150	50
144	190	45		194	240	95
220	220	85		255	255	135
						

Obrázek 4.3: Příklad zvýšení jasu

4.2 Úprava kontrastu

Zvýšení či snížení kontrastu snímku se upravuje pomocí aritmetických operací násobení a dělení. Při násobení pixelů v matici konstantou se kontrast zvyšuje a analogicky se při dělení kontrast snižuje. Opět může při těchto operacích dojít k přetečení/podtečení hodnot z rozsahu, což se řeší totožně jako u úpravy jasu, tedy omezením na 0 při podtečení a na 255 při přetečení. [Hoz03]

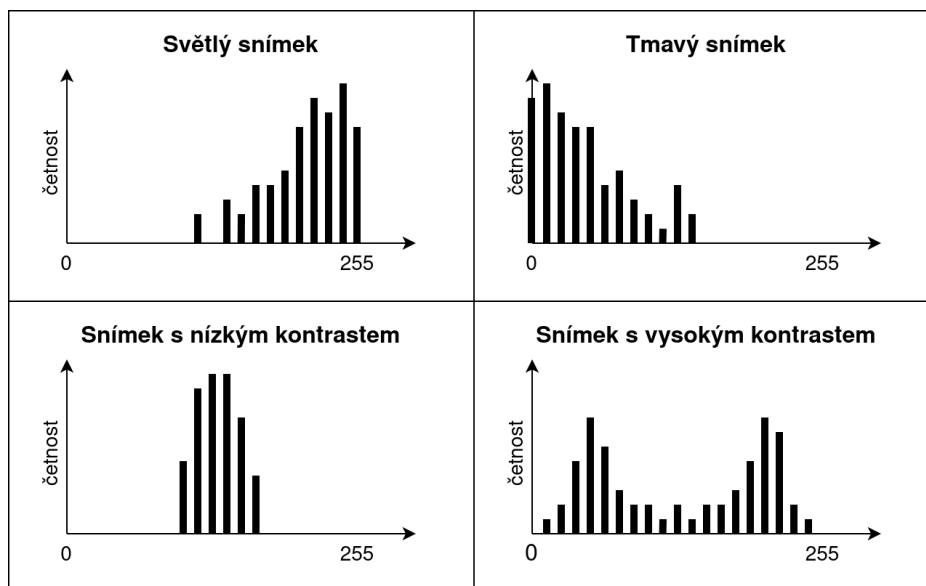


Obrázek 4.4: Příklad zvýšení kontrastu

4.3 Úprava histogramu

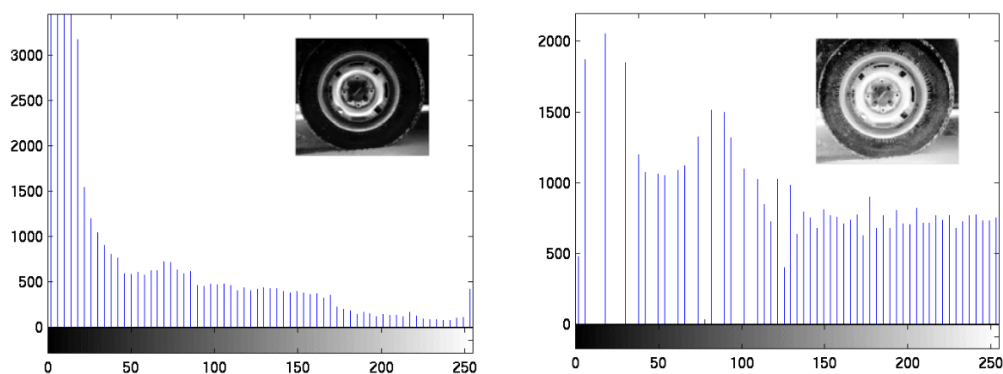
Histogram je graficky vyjádřená četnost hodnoty v nějakém souboru hodnot. V případě obrazových dat se tím rozumí četnost výskytu hodnot (barev) v daném snímku (matici pixelů). Z histogramu se dá vyčíst, zda je snímek příliš jasný/tmavý nebo zda má vysoký/nízký kontrast. U příliš jasného snímku převažují body s vysokou hodnotou jasu, tedy histogram je posunut vpravo. Naopak u tmavého snímku s převažujícími pixely s nízkou hodnotou jasu je histogram posunutý vlevo. U snímku s nízkým kontrastem je rozptýl hodnot jasu pixelů malý a histogram je soustředěný na úzkém prostoru grafu. U vysoko kontrastního snímku je histogram rozložen po celém spektru (s několika vrcholy) (Obrázek 4.5). [Hoz03]

4. Uplatnění filtrů při předzpracování CT snímků



Obrázek 4.5: Některé typy histogramů

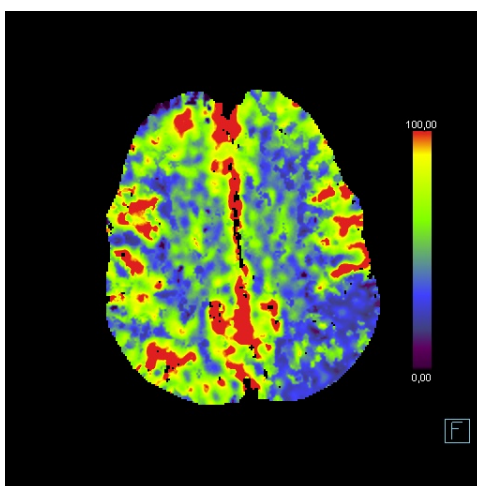
Úprava histogramu je častou metodou úpravy obrazových dat, a nejen těch medicínských. Změnou tvaru histogramu se mění dynamicky jak úroveň jasu, tak i kontrast snímku. Jednou z nejčastějších technik úpravy histogramu je tzv. **vyrovnání histogramu**. Jedná se o bodovou jasovou transformaci, která pro jasové úrovně blízko maxim zvyšuje kontrast a naopak blízko minim původního histogramu kontrast snižuje. [Hoz03]



Obrázek 4.6: Ukázka vyrovnání histogramu [Hoz03]

4.4 Pseudobarva

Pseudobarva je technika používaná k zobrazení šedotónových snímků pomocí barevných map. Často se využívá právě při zpracování medicínských obrazových dat, protože může pomoci zvýraznit důležité prvky ve snímku a usnadnit jeho interpretaci. Například na CT snímku jsou použity různé odstíny šedé k reprezentaci různých denzit tkání, ale rozlišit tyto odstíny může být obtížné. Pomocí pseudobarvy mohou být různé denzity tkání reprezentovány různými barvami, což usnadňuje jejich rozlišení. Pseudobarvu lze také použít ke zvýraznění abnormalit ve snímku. Namapováním těchto prvků na konkrétní barvu je lze pak snadněji identifikovat a sledovat v průběhu času. [Hla10]



Obrázek 4.7: Pseudobarevný snímek mozku

4.5 Geometrické transformace

Geometrické transformace se používají ve zpracování medicínských snímků k zarovnání nebo registraci (zarovnání + případná deformace) snímků nasnímaných v různém čase, nebo pořízených různými zobrazovacími metodami. Tyto transformace lze použít ke korekci rozdílů v orientaci, měřítku nebo poloze mezi snímky. Konkrétně pokud například lékař porovnává aktuální CT vyšetření pacienta s dřívějším vyšetřením, aby sledoval progresi onemocnění, je důležité, aby byly oba snímky zarovnané, aby bylo možné přesně změřit a porovnat změny ve tkáni. Podobně, pokud například chirurg plánuje zákrok pomocí snímků z více metod (jako CT či MR), je důležité, aby byly snímky registrovány, aby měl chirurg úplný a přesný pohled na anatomii pacienta. [Hov09]

Geometrické transformace lze také obecně použít ke zlepšení vizualizace medi-
cínských obrazových dat, kdy je například snímek pootočený nebo z nějakého dů-
vodu zdeformovaný. Použitím geometrické transformace pro korekci těchto zkres-
lení může být snímek lépe interpretovatelný. [Hov09]

Tyto transformace spočívají ve změně souřadnic bodů v původním snímku. Ge-
ometrickou transformaci lze zapsat jako rovnici

$$(x', y') = T(x, y), \quad (2)$$

kde (x, y) jsou souřadnice původního bodu, (x', y') jsou souřadnice výsledného
bodu a T je funkce představující samotnou geometrickou transformaci. [Hov09]

Geometrická transformace má dva kroky. Prvním je samotná transformace sou-
řadnic a druhým je interpolace, která vypočítává hodnotu (barvu) pro každý trans-
formovaný bod. [Hla10]

4.5.1 Transformace souřadnic

Mezi základní transformace patří posun, otočení a změna měřítka, tedy zvětše-
ní/zmenšení snímku.

Posun

Posun obrazu o hodnotu a ve směru x a o hodnotu b ve směru y se dá matematicky
vyjádřit rovnicemi

$$\begin{aligned} x' &= x + a \\ y' &= y + b. \end{aligned} \quad (3)$$

Matice posunutí má pak tvar

$$M(a, b) = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

[Hov09]

Otočení

Otočení obrazu o úhel α se dá matematicky zapsat rovnicemi

$$\begin{aligned} x' &= x \cos(\alpha) - y \sin(\alpha) \\ y' &= x \sin(\alpha) + y \cos(\alpha). \end{aligned} \quad (5)$$

Matice otočení (v tomto případě proti směru hodin) má pak tvar

$$M(a, b) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

[Hov09]

Zvětšení/zmenšení

Zmenšení/zmenšení obrazu o hodnotu u ve směru x a o hodnotu v ve směru y se dá matematicky popsat rovnicemi

$$\begin{aligned} x' &= ux \\ y' &= vy. \end{aligned} \quad (7)$$

Matice posunutí má pak tvar

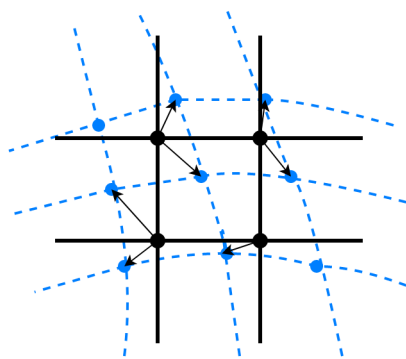
$$M(a, b) = \begin{bmatrix} u & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

[Hov09]

4.5.2 Interpolace

Po transformaci souřadnic je nutné z hodnot bodů v původním snímku určit hodnotu odpovídající pozici výsledného bodu v transformovaném snímku. Vzhledem k tomu, že po transformaci nemusí být souřadnice výsledného bodu (x', y') celočíselné, musí se jeho hodnota určit interpolací. [Hla10]

Existuje více metod k tomu sloužících. Nejjednodušší metoda je interpolace nejbližším sousedem. Hodnota výsledného bodu je určena hodnotou bodu v původním snímku, který je nejbliž k vypočteným souřadnicím (Obrázek 4.8). Na obrázku jsou modře vyznačené transformované body a černě s šipkami jsou vyznačeni nejbližší sousedi. [Hov09]



Obrázek 4.8: Interpolace nejbližším sousedem

4.6 Filtrace

Filtrace je označení pro transformaci snímku, která převádí hodnoty pixelů na jiné, za účelem zvýraznění či potlačení některých jeho vlastností. K výpočtu nové hodnoty pixelu slouží matice (maska), která se po snímku posouvá a vypočítává se nová hodnota pixelu uprostřed masky. Tento postup se nazývá konvoluce. [Hov09]

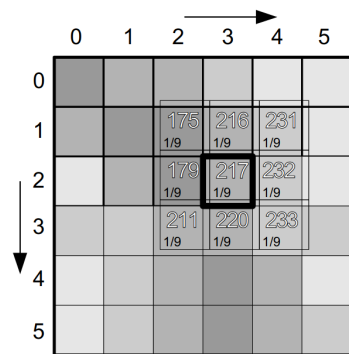
Dalšími filtry na které bude tato kapitola zaměřena jsou filtry vyhlazovací a hranové. Vyhlazovací filtry slouží k odstranění rušivých jevů ve snímku, jako je například šum. Hranové filtry slouží k detekování, resp. nalezení, hran ve snímku.

4.6.1 Konvoluce

Jak již bylo zmíněno výše, principem konvoluce je posouvání konvoluční masky (matice) po pixelech ve snímku. Při každém posunutí masky se vynásobí koeficient masky s hodnotami pixelů a tyto vypočtené výsledky se sečtou. Výsledná hodnota se uloží na pozici prostředku matice. Z tohoto důvodu se jako konvoluční masky používají matice s lichým počtem řádků i sloupců (Obrázek 4.9). [Bru06]

Při aplikaci masky dochází na okraji snímku k jeho překročení. To lze řešit třemi způsoby:

- na okraj snímku se přidají pixely a jejich hodnota se nastaví na 0, čímž se zajistí jejich ignorování při aplikaci masky
- maska se posouvá pouze v prostoru snímku, což ale zapříčiňuje chyby na okrajích, jelikož se hodnoty krajních pixelů nepřečítávají
- na okraji snímku se aplikuje zmenšená maska [Hov09]



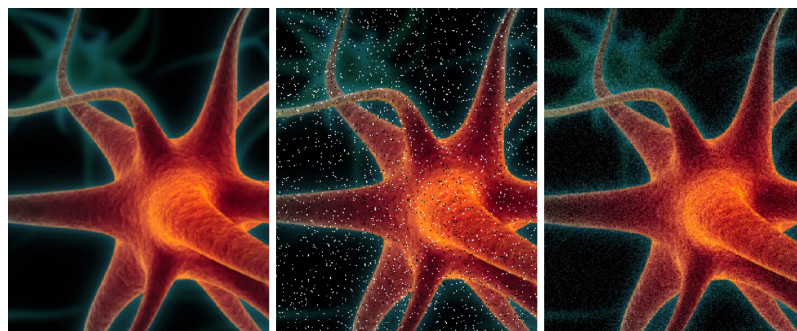
$$H_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\frac{1}{9} \cdot 175 + \frac{1}{9} \cdot 216 + \frac{1}{9} \cdot 231 + \frac{1}{9} \cdot 179 + \frac{1}{9} \cdot 217 + \frac{1}{9} \cdot 232 + \frac{1}{9} \cdot 211 + \frac{1}{9} \cdot 220 + \frac{1}{9} \cdot 233 = 213$$

Obrázek 4.9: Ukázka konvoluce [Bru06]

4.6.2 Vyhlažovací filtry

Vyhlažovací filtry slouží k odstranění nežádoucích jevů ve snímku, jako je například šum, který se na snímku projevuje jako zrnění. Nicméně vždy je dobré zvážit charakteristiky konkrétních dat, se kterými se pracuje, protože pokud jsou data již hladká nebo postrádají výrazné změny hodnot pixelů, nebude mít filtr příliš velký efekt. Dokonce v případě, že je filtr nastaven nevhodně (např. jako příliš agresivní), může odstranit důležité detaily či vlastnosti snímku. Vyhlažovací filtry jsou i přesto velice užitečnou a důležitou technikou při odstraňování šumu ze snímku. Nicméně existuje více druhů šumů a pro každý druh se hodí použít jiný vyhlažovací filtr. [Hov09]



originální snímek

šum typu sůl a pepř

Gaussův šum

Obrázek 4.10: Druhy šumu [Hov09]

Vyhlažovací filtry pracují na principu konvoluční masky. Metod vyhlažování je více a dělí se na lineární a nelineární metody. Níže jsou uvedeny a popsány jen některé vybrané metody, jejichž princip jde jednoduše a krátce v rámci této práce popsat. Rozsáhlejší výčet metod a jejich princip lze nalézt ve zdroji [Hov09].

Lineární metody

Novou hodnotu pixelu vypočítávají lineární metody jako lineární kombinaci hodnot z jeho okolí. Nevýhodou lineárních metod je to, že při potlačování šumu se značně vyhladí i hrany (náhlé změny hodnot) ve snímku, čímž dochází k určité ztrátě informace. [Kel03]

Vyhlazování průměrováním. Jedná se o nejjednodušší lineární metodu, která se nicméně příliš nehodí pro odstraňování šumu, jelikož opravdu výrazně vyhladí všechny hrany ve snímku. Používá se proto pouze za účelem vyhlazení snímku. Hodnotu pixelu vypočte tato metoda jako aritmetický průměr z hodnot v jeho okolí. K výpočtu lze využít konvoluční masku, která pro průměrování pro okolí rozměru 3x3 má tvar

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (9)$$

[Hov09; Kel03]

Gaussovo vyhlazování. Tato lineární metoda je ideální na odstranění tzv. Gaussova šumu (Obrázek 4.10). Využívá konvoluční masky, jejíž hodnoty blíže ke středu masky mají větší váhu a odpovídají tak hodnotám na Gaussově křivce v závislosti na střední hodnotě $n = 0$ a na zadaném rozptylu σ^2 , který určuje strmost křivky. V praxi se obvykle nepoužívají masky s $\sigma > 3$, pokud nestojíme o opravdu velké vyhlazení snímku. Hodnoty v masce se normují tak, aby jejich součet byl roven jedné. Příklad masky Gaussova vyhlazení s rozptylem $\sigma^2 = 1$ má tvar

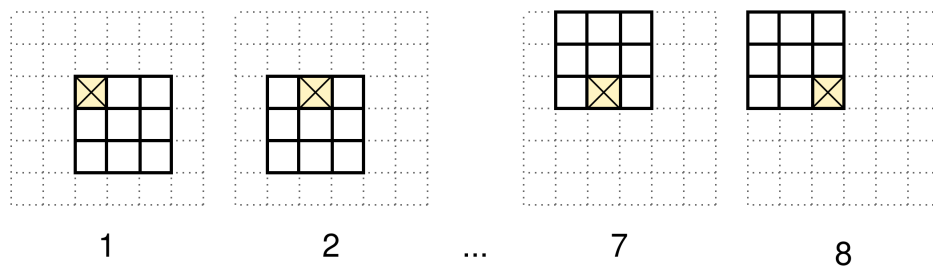
$$m = \frac{1}{1000} [4, 54, 242, 400, 242, 54, 4]. \quad (10)$$

Takováto jednorozměrná maska se aplikuje na řádky snímku a poté na sloupce. [Hov09]

Nelineární metody

Oproti lineárním metodám se nelineární metody vyhlazování snaží odstranit šum, ale zároveň ponechat a nevyhladit ve snímku hrany.

Metoda rotační masky. Tato metoda funguje na principu rotace konvoluční masky kolem daného pixelu. Vychází z lineární metody vyhlazování průměrováním. Pro každou polohu masky se spočte rozptyl hodnot celého okolí (pod maskou). Z masky, pro kterou vyjde rozptyl nejmenší, se vypočítá aritmetickým průměrem hodnota pro výsledný pixel. Nejen že tato metoda nevyhlazuje hrany, ale má dokonce i mírně ostřící účinek. [Hov09]



Obrázek 4.11: Metoda rotační masky

Metoda s omezením jasu (hodnoty pixelu). Využívá lineární metody vyhlazování průměrováním tak, že spočte aritmetický průměr okolí pixelu a pokud je rozdíl mezi vypočítaným průměrem a původní hodnotou pixelu menší než nějaký zvolený práh, nastaví jako novou hodnotu pixelu vypočítaný průměr, jinak pixelu nechá jeho původní hodnotu. Tím tato metoda zabraňuje nechtěnému vyhlazování hran. [Hov09]

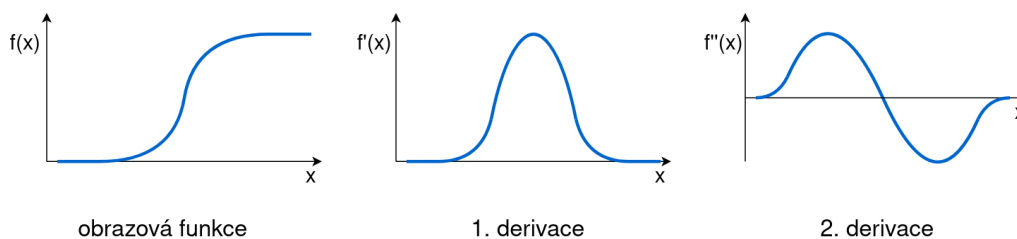
Metoda mediánu. Princip této nelineární metody spočívá v posunu masky po pixelech, kdy se na každé pozici vybere prostřední hodnota (medián) z hodnot pixelů v okolí (pod maskou). Tento filtr se hodí především na šum typu sůl a pepř (Obrázek 4.10). [Hov09]

4.6.3 Hranové filtry

Hranové filtry se běžně používají při zpracování medicínských snímků ke zvýšení jejich kontrastu a zvýraznění hranic mezi různými tkáněmi nebo strukturami, což vede k pro člověka lepšímu obrazovému vjemu. Zároveň jsou tyto filtry užitečné také pro zlepšení vizualizace malých struktur nebo abnormalit ve snímku a usnadňují identifikaci a měření specifických prvků v něm. [Hov09; Kel03]

Detekce hran spočívá v nalezení míst ve snímku, kde dochází k náhlé změně hodnot pixelů. S tím souvisí i ostření obrazu, kterého se dosáhne zvýšením strmosti hran. [Hov09; Kel03]

K detekci hran se využívají parciální derivace. Ty se aplikují na obrazovou funkci, což je funkce dvou proměnných, kde hodnota v každém bodě odpovídá intenzitě jasu pixelu na příslušné pozici. Obrazová funkce tedy obsahuje informace o rozložení intenzity jasu v obrazu. Pomocí derivace se nalezne gradient obrazové funkce. Z něho se dá poznat směr a velikost gradientu. Z velikosti gradientu se pak dá určit, zda se jedná o hranu či nikoliv. [Kel03]



Obrázek 4.12: Znárodnění hrany v obrazové funkci a její derivace

Hranové operátory lze rozdělit na dvě skupiny. Operátory v první skupině využívají konvoluční masky a hrany hledají na základě maxim v první derivaci obrazové funkce (maxima nabývá první derivace obrazové funkce právě v místě hrany). Operátory v druhé skupině hledají hrany v místech, kde druhá derivace obrazové funkce nabývá hodnoty nula. Mezi hlavní představitele hranových detektorů fungujících na tomto principu se řadí Cannyho hranový detektor, který je obecně považován za neoptimálnější. [Kel03]

Cannyho hranový detektor

Tento hranový detektor byl navržen tak, aby splňoval následující tři základní kritéria:

- nalezení všech reálných hran
- co nejpřesnější poloha nalezené hrany vůči reálné
- nedetekování jedné reálné hrany vícekrát

Algoritmus Cannyho hranového detektoru lze jednoduše popsat ve čtyřech krocích. Prvním krokem je vyhlazení obrazové funkce. Druhým krokem je určení velikosti intenzity a strmosti hran v každém bodě snímku, s využitím první derivace. Ve třetím kroku se označí všechny možné hrany v místech s maximální velikostí intenzity. Čtvrtým, posledním, krokem je prahování hran na základě velikosti jejich intenzity. Jsou dány dva prahy – horní a dolní. Algoritmus pro prahování sleduje posloupnost bodů potenciálně tvořících hranu. Pokud má zkoumaný bod intenzitu vyšší než je horní práh, je hned označen jako bod hrany. Pokud má intenzitu nižší než dolní práh, je označen jako pozadí. V případě, že má bod intenzitu mezi horním a dolním prahem, je jako bod hrany označen, jen pokud se v jeho okolních pixelech nachází bod, který už bodem hrany je nebo má intenzitu vyšší než horní práh. [Hov09; Kel03]

Uplatnění filtrů při segmentaci CT snímků

5

Segmentace je další operací při zpracování medicínských obrazových dat. Spočívá v rozdělení snímku na více oblastí na základě určitých společných vlastností jeho pixelů/voxelů. Obvykle se jedná o oddělení jednoho či více objektů zájmu ve snímku od pozadí, tedy o určení toho, které pixely/voxely náleží danému objektu. Provedením segmentace můžeme určit velikost (objem/obsah) daného objektu či jeho umístění ve snímku.

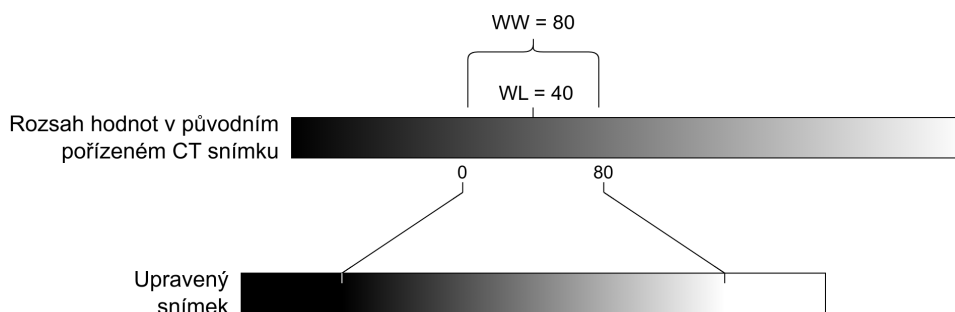
V oblasti medicíny se jedná o velice důležitou fázi zpracování snímku, která se využívá zejména ke sledování změn velikosti, tvaru či pohybu různých nálezů v těle (např. krevních sraženin, nádorů). [Toe17]

5.1 Windowing

Windowing je metoda, která umožňuje zvýraznit konkrétní struktury v CT snímku. Při použití této metody se vybere určitá oblast snímku, kterou chceme zvýraznit, a poté se aplikuje tzv. okno, které upraví jas a kontrast v této oblasti. Okno definují dva parametry, a to šířka a úroveň okna. Šířka okna (Window Width, WW) udává rozsah hodnot, které snímek obsahuje. Čím širší okno, tím větší rozsah hodnot se zobrazí. Tím však může dojít ke snížení rozdílů hodnot mezi měkkými tkáněmi, které se tak utlumí. Široké okno se tedy hodí pro snímky s výrazně rozdílnými denzitami. Naopak pro snímky s měkkými tkáněmi je vhodné použít úzké okno. Dalším parametrem je úroveň okna (Window Level, WL). Ta udává střed rozsahu zobrazených hodnot snímku. Když se úroveň okna sníží, snímek bude jasnější a naopak. Hodnoty CT snímku jsou udávány v Hounsfieldových jednotkách (HU), které udávají denzitu, tedy hodnotu absorbovaného rentgenového záření. [MBK23]

5. Uplatnění filtrů při segmentaci CT snímků

Z WW (šířka okna) a WL (úroveň okna) lze vypočítat horní a dolní úroveň šedé, tj. které hodnoty se už budou zobrazovat bíle a které černě. Horní úroveň se vypočte jako $WL + (WW/2)$ a dolní úroveň jako $WL - (WW/2)$. Tedy například pro WW:80 a WL:40 budou hodnoty > 80 zobrazeny bíle a hodnoty < 0 budou černé (Obrázek 5.1). [MBK23]



Obrázek 5.1: Princip metody windowing

Následující tabulka udává typické hodnoty WW a WL pro jednotlivé objekty v lidském těle.

Objekt	WW (HU)	WL (HU)
mozek	80	40
plíce	1500	-600
játra	150	30
měkké tkáně	200 až 400	20 až 60
kosti	1800	400

Tabulka 5.1: Přehled typických hodnot WW a WL [MBK23]

5.2 Segmentace prahováním

Prahování (angl. thresholding) je nejjednodušší metoda segmentace medicínských obrazových dat. Je však vhodná pouze pro snímky, kde oblast zájmu vyčnívá výrazně vyšší/nížší hodnotou jasu než pozadí, což většina CT snímků splňuje. Segmentace prahováním ale nemusí být účinná pro všechny medicínské snímky, zejména pro ty, které mají nízký kontrast, jsou zašuměné, nebo se objekty v nich překrývají.

Tato metoda přiřadí pixelům objektu zájmu hodnotu 1 a pixelům pozadí hodnotu 0. Tedy segmentaci s snímku f s prahem t pro pixel/voxel \mathbf{v} lze zapsat jako

$$s(\mathbf{v}) = \begin{cases} 1, & \text{pokud } f(\mathbf{v}) > t \\ 0, & \text{jinak} \end{cases} \quad (11)$$

[Toe17]

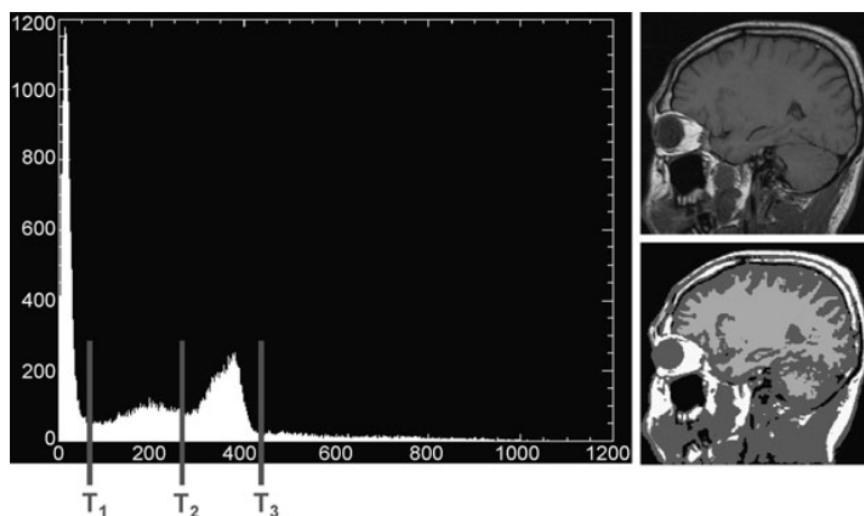
85	120	196
71	152	82
40	66	69

→

1	1	1
0	1	0
0	0	0

Obrázek 5.2: Příklad segmentace prahováním s prahem $t = 85$

Prahů pro segmentaci může být nastaveno více, snímek je pak rozdělen na více objektů. Údolí v histogramu snímku indikují potenciálně vhodné hodnoty prahů (Obrázek 5.3). [Toe17]



Obrázek 5.3: Ukázka segmentace prahováním s více různými prahy [Toe17]

Otsuova metoda automatického určení prahu

Práh pro segmentaci prahováním lze určit manuálně a nebo automaticky. Jednou z metod pro automatické určení prahu je právě Otsuova metoda. Ta iteračně hledá optimální práh pro histogram snímku. Výsledkem je pak práh t mezi dvěma třídami (oblast zájmu a pozadí) takový, že hodnota rozptylu jasu v rámci třídy (q_w) je nízká a naopak hodnota rozptylu mezi třídami (q_b) vysoká. Tedy poměr $\lambda(t) = q_b(t)/q_w(t)$ je co největší. [Špa12; Toe17]

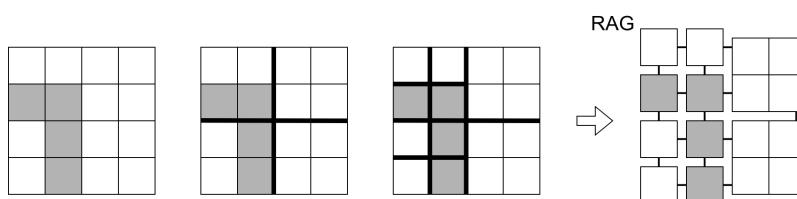
5.3 Segmentace založená na regionech

Základním prvkem segmentace založené na regionech je tzv. kritérium homogenity. To je např. hodnota rozptylu jasu nebo barvy oblasti ve snímku, pro kterou je oblast ještě považována za homogenní. Tyto metody jsou obecně účinné pro segmentaci

snímku s velkým množstvím regionů a pro snímky s velkým množstvím šumu, kde segmentace prahováním nebo segmentace založená na hranách selhává. [Toe17]

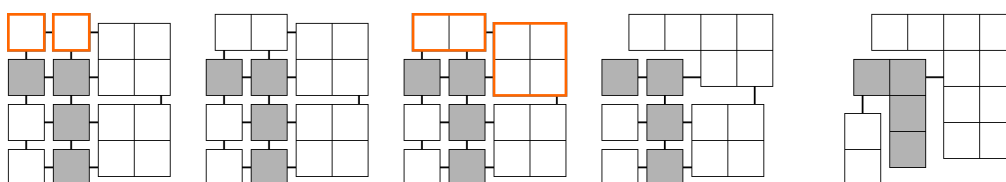
5.3.1 Metoda dělení a spojování oblastí

Tato metoda (angl. split and merge) nejprve snímek rozdělí na nejmenší homogenní oblasti. Zpočátku je celý snímek považován za jednu oblast. V každé iteraci se oblast, když nesplňuje kritérium homogenity, rozdělí na čtvrtiny. Tento krok se opakuje do doby, kdy všechny oblasti kritérium homogenity splňují. Jednotlivé oblasti se poté namapují do grafu sousedících oblastí (angl. Region-Adjacency-Graph, RAG). [Toe17]



Obrázek 5.4: Princip dělení oblastí

Po rozdělení snímku na homogenní oblasti a jejich namapování do RAG se spojí případné rozdělené oblasti. V každé iteraci algoritmu se pro oblast vypočte hodnota homogenity s každou sousedící oblastí. Dvě nejpodobnější oblasti se pak spojí v jednu. Tento postup se opakuje dokud existují oblasti, pro které jejich hodnota homogenity splňuje kritérium homogenity. [Toe17]

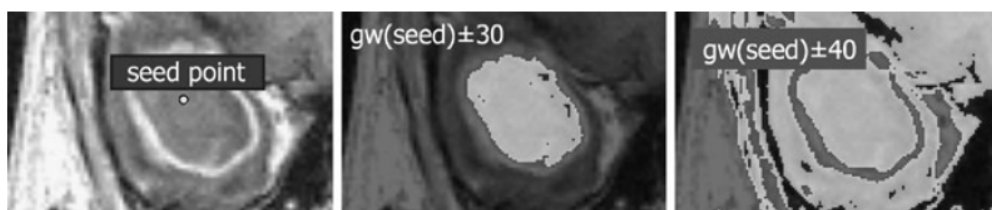


Obrázek 5.5: Princip spojování oblastí

Stejného výsledku by se dosáhlo i za použití pouze spojování oblastí. Snímek by byl na začátku algoritmu rozdělen na jednotlivé pixely, které by se pak podle výše popsaného algoritmu spojovaly. To by však nebylo příliš optimální co se týče výpočetní složitosti, která by tak vycházela na $O(N^2)$, kdy N je počet počátečních oblastí (kterých může být zvláště v případě 3D dat skutečně velké množství). Kombinace rozdělení a spojování oblastí výrazně sníží výpočetní složitost, jelikož výsledkem rozdělení na homogenní oblasti je výrazně méně oblastí než kolik bylo původních pixelů snímku. [Toe17]

5.3.2 Metoda narůstání oblastí

Proces segmentace narůstáním oblastí (angl. region growing) začíná určením tzv. seedu, což je počáteční pixel, od kterého se začínají hledat okolní pixely s podobnými vlastnostmi. Poté se postupně přidávají další pixely k dané oblasti na základě kritéria homogenity. Proces končí ve chvíli, kdy již nelze přidat další pixely, které by kritérium homogenity splňovaly. [Toe17]



Obrázek 5.6: Segmentace narůstání oblastí s různými kritérii homogenity [Toe17]

5.4 Segmentace založená na hranách

Segmentaci založenou na hranách lze použít, je-li snímek kontrastní a objekt zájmu obklopen hranami tvořícími uzavřenou křivku. I tento druh segmentace vyžaduje jistou homogenitu oblasti, ale mnohem méně než u výše zmíněných segmentací. Důležité je především to, aby případná změna hodnot pixelů nebyla vyhodnocena jako falešná hrana, aby hrany nebyly nespojité, několikanásobné, příliš silné či naopak tenké. Hrany, detekované hranovými detektory (popsanými v sekci 4.6.3), se proto obvykle ještě modifikují, právě zeslabením příliš silných hran, spojením nespojitých hran či odstraněním falešných hran. [Dor09]

5.4.1 Metoda sledování okrajů

Tato intuitivní metoda se dá využít pro snímky s oblastmi jednoznačně určenými hranami, které ale mohou být nespojité či několikanásobné. Samozřejmě je k použití této metody potřeba znát další informace jako polohu a směr hrany, získané při detekci hran. Taktéž se musí definovat kritérium podobnosti sousedních pixelů. [Dor09]

Metoda funguje tak, že vezme počáteční pixel hrany a mezi jeho nejbližšími sousedy hledá jemu nejpodobnější pixel. Takto pokračuje do doby, kdy buď v jeho okolí již není žádný pixel vyhovující požadovanému kritériu podobnosti, nebo dokud se křivka neuzavře, a nebo nenarazí na okraj snímku. Poté se určí nový počáteční pixel a algoritmus se opakuje. [Dor09]

Návrh nástroje pro zpracování medicínských snímků

6

V rámci praktické části této bakalářské práce je vytvořen nástroj pro dávkové zpracování sady medicínských obrazových dat dle uživatelem zadané konfigurace. Je implementován v jazyce Python s využitím knihovny pro zpracování a analýzu obrazu SimpleITK.

6.1 Analýza a návrh

6.1.1 Existující nástroje pro zpracování obrazu

Pro předzpracování, segmentaci a analýzu medicínských obrazových dat existuje více nástrojů. Jsou to buďto software ve formě aplikace s vlastním GUI (grafické uživatelské rozhraní), nebo jednotlivé funkce obsažené v knihovně či balíku programovacího jazyka.

Aplikace s GUI umožňuje uživateli upravovat obrazová data jednoduše pomocí grafických ovládacích prvků. Nevýhodou však může být přílišné množství pro konkrétního uživatele nepotřebných a zbytečných funkcí, kvůli čemuž mohou být pro uživatele nepřehledné. Nebo v nich naopak chybí funkce, které by uživatel zrovna potřeboval. Níže jsou stručně popsány často používané programy ImageJ a 3D Slicer.

Specializované knihovny či balíky pro programovací jazyky umožňují uživatelům vytvořit si vlastní sadu operací nad jejich obrazovými daty. Výhodou je, že takto vytvořený program obsahuje jen ty funkce, které uživatel skutečně potřebuje a chce používat. Nicméně v tomto případě je potřeba, aby uživatel měl nějaké programátorské znalosti a uměl programovací jazyk, pro který je konkrétní knihovna napsána. Níže jsou krátce představeny dvě knihovny pro zpracování obrazu napsané mimo jiné i pro Python – SciPy a SimpleITK.

ImageJ

ImageJ je program, který umožňuje zobrazovat, upravovat, zpracovávat, analyzovat, ukládat a tisknout 2D i 3D biomedicínské snímky. Podporuje mnoho obrazových formátů včetně BMP, JPEG, GIF, DICOM a TIFF. Poskytuje standardní funkce pro zpracování obrazu, jako je například úprava kontrastu a jasu, doostření, vyhlazování, filtrování a detekce hran. Také umožňuje mimo jiné provádět geometrické transformace a k dispozici je i funkce prostorové kalibrace, která poskytuje měření rozměrů objektů v reálném světě. [Im23b]

ImageJ je vyvíjen od roku 1997. V roce 2010 proběhla jeho úprava a redesign, tato jeho distribuce je známa pod názvem ImageJ2. Další distribucí původního ImageJ je Fiji, který umožňuje i konverzi potřebných datových struktur mezi ImageJ a ImageJ2. Licence ImageJ je veřejná (public domain). [Im23a]

Distribuce jsou ke stažení pro Windows, Linux i macOS. ImageJ je napsán v jazyce Java a umožňuje rozšíření prostřednictvím pluginů. [Im23b]

3D Slicer

3D Slicer je software pro vizualizaci, zpracování, segmentaci, registraci, anotaci a analýzu lékařských, biomedicínských (ale i jiných) 2D, 3D či 4D snímků. Podporuje i zobrazování dat z výpočetní tomografie, magnetické rezonance a sonografie. Mimo standardních funkcí pro zpracování obrazu umožňuje také zobrazování v reálném čase například z chirurgických navigačních systémů, robotických zařízení a senzorů. Vizualizaci dat umožňuje nejen v desktopové aplikaci, ale i ve virtuální realitě. [Sli23]

Software 3D Slicer začal být vyvíjen v roce 1998 a v roce 2007 byla vydána kompletně přepracovaná verze 3. je distribuován pod BSD open source licencí, která neobsahuje žádná omezení na použití softwaru. [Sli23]

Jedná se o bezplatný open-source software dostupný pro operační systémy Windows, Linux a macOS. Uživatelé mohou rozšířit funkčnost 3D Sliceru doinstalováním dalších modulů, spuštěním vlastních skriptů ve vestavěné Python konzoli, nebo implementací vlastních modulů v Pythonu nebo C++. [Sli23]

SciPy

SciPy rozšiřuje o další nástroje a struktury knihovnu NumPy, která obsahuje vícerozměrné pole a maticové datové struktury a mnoho matematických algoritmů a funkcí. Balíček `scipy.ndimage` nabízí řadu funkcí pro zpracování a analýzu obrazu, konkrétně například lineární a nelineární filtrování, interpolaci, binární morfologii a měření hodnot objektů. [Vir+20]

Python začal být vyvíjen v 80. letech minulého století. SciPy se pak objevuje na konci 90. let. Po roce 2000 se začíná dostávat do povědomí vědecké společnosti a později se jí také dostává řádné dokumentace. [Vir+20]

SciPy kombinuje flexibilitu Pythonu s rychlostí kompilovaného kódu díky tomu, že je implementována v nízkoúrovňových jazycích (Fortran, ANSI C, C++). [Vir+20]

SimpleITK

SimpleITK je zjednodušené programovací rozhraní pro algoritmy a datové struktury z Insight Toolkit (ITK). Sada nástrojů SimpleITK podporuje více než 15 různých obrazových formátů, poskytuje přes 280 filtrů pro zpracování a analýzu obrazu a umožňuje provádět transformace a registrace 2D až 5D snímků. V kontextu biomedicíny se SimpleITK používá především k analýze struktur zobrazených pomocí výpočetní tomografie, magnetické rezonance a mikroskopie. [Sim23]

Zásadním konceptem v SimpleITK (a obecně v ITK), který ji tak výrazně odlišuje od mnoha jiných knihoven pro analýzu obrazu, je to, že se snímkem není zacházeno jako s polem, ale že je definován sadou bodů na mřížce zabírající fyzickou oblast v prostoru. Díky tomu existuje představa o umístění obrazu ve fyzickém prostoru. [Si20b]

SimpleITK byla vytvořena za účelem zjednodušení používání Insight Toolkit (ITK), aby byl přístupnější širšímu publiku. První verze byla veřejnosti představena v roce 2017. Projekt SimpleITK je součástí neziskového vzdělávacího konsorcia, které se věnuje propagaci a údržbě volně dostupného softwaru pro analýzu medicínských obrazových dat. SimpleITK je distribuována pod licencí Apache License 2.0. [Sim23]

Podporuje rozhraní pro několik programovacích jazyků včetně C++, Python, R, Java, C# či Ruby. To dává uživatelům možnost vyvinout si svůj vlastní postup zpracování a analýzy obrazu v programovacím jazyce, který znají nejlépe. [Sim23]

6.1.2 Návrh řešení

Prvním krokem při návrhu nástroje pro dávkové zpracování sady medicínských obrazových dat bylo specifikovat, k čemu a komu má sloužit, co všechno má umět a jak by se měl ovládat.

Cílem je vytvořit jednoduchý nástroj, který bude sloužit ke zpracování, zejména tedy předzpracování CT snímků, a bude obsahovat funkce a filtry, které se obvykle při předzpracování takovýchto snímků používají.

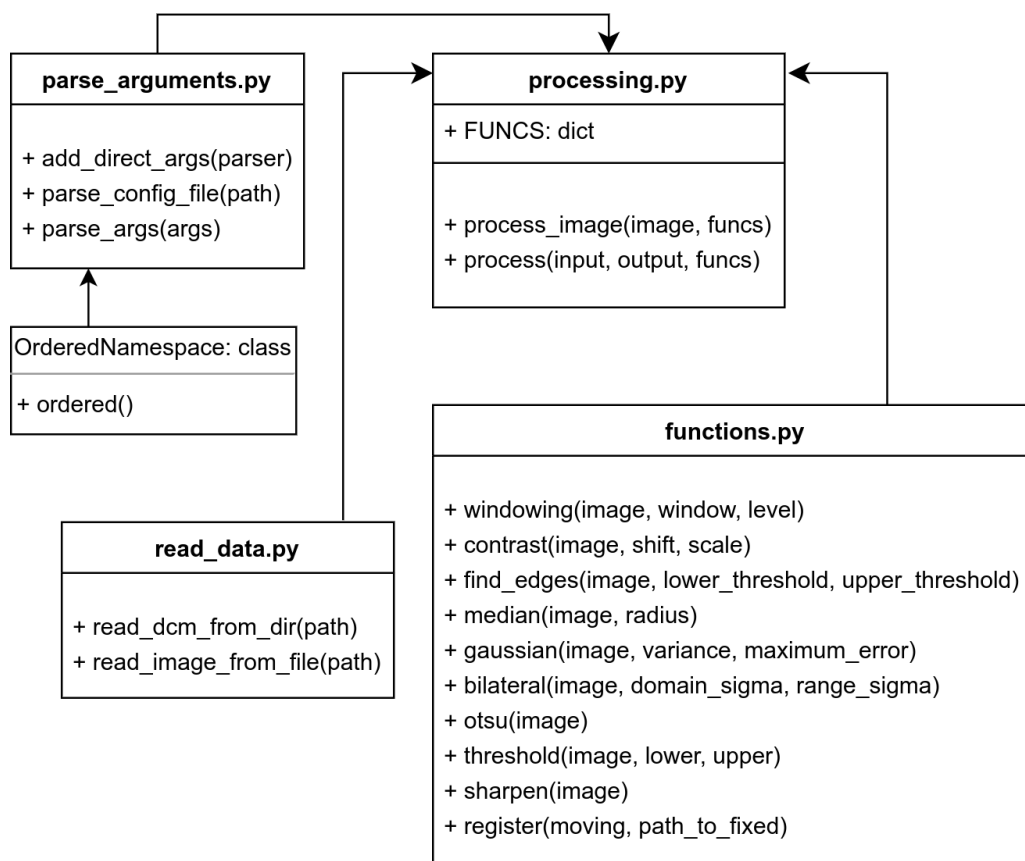
Účelem navrženého nástroje je snadné a rychlé zpracování sady medicínských obrazových dat s možností automatizace některých operací. Zároveň se u uživatele předpokládá jistá znalost filtrů a toho, co a jak se snímky chce provést. Navržený nástroj má tedy umožnit jejich rychlé, jednoduché a efektivní použití. Také má sloužit pro dávkové zpracování dat, tedy pro zpracování většího množství dat najednou se stejnými parametry filtrů. A nebo například pro vyzkoušení různých variant nastavení filtrů u stejné sady snímků. Z těchto důvodů bylo upuštěno od implementace programu s GUI, jelikož pro tento účel je vhodnější implementovat nástroj jako skript volaný s parametry. Zpracovávané snímky a funkce s jejich parametry, které mají být na snímky aplikovány, jsou skriptu předávány při spuštění buďto přímo v příkazové řádce, nebo v konfiguračním souboru. To umožní proces zpracování obrazových dat snadno zautomatizovat. V případě, že by uživateli nestačily implementované funkce, může skript dále rozšířit o jakékoliv další funkce, které potřebuje.

Navržený skript bude implementován v jazyce Python, který je pro programování skriptů ideální. Navíc má širokou podporu pro zpracování obrazu a mnoho grafických knihoven a jeho popularita v poslední době výrazně stoupá [TIO23]. Tento jazyk navíc podporuje i knihovna SimpleITK, která byla vyvinuta speciálně pro zpracování medicínských obrazových dat a obsahuje veliké množství funkcí a filtrů (jak již bylo zmíněno v předchozí sekci 6.1.1). Tato knihovna tak umožní rychlou a snadnou implementaci požadovaných funkcí a případné doplnění a rozšíření navrhovaného skriptu.

6.2 Implementace

Jak již bylo zmíněno výše, nástroj je implementován v programovacím jazyce Python, ve formě skriptu spouštěného z příkazové řádky, a využívá filtry z knihovny SimpleITK.

Implementace skriptu je rozdělena do 4 samostatných modulů: `processing.py`, `parse_arguments.py`, `read_data.py` a `functions.py` (Obrázek 6.1). Hlavním modulem, který se spouští, je `processing.py`. V něm se volají funkce z ostatních modulů a pomocí nich se obrazová data zpracovávají. Modul `parse_arguments.py` obsahuje funkce, které načtou, zpracují a uloží argumenty z příkazové řádky zadané při spuštění programu. V modulu `read_data.py` jsou funkce pro načtení vstupních dat. A nakonec modul `functions.py` obsahuje jednotlivé funkce pro zpracování vstupních dat.



Obrázek 6.1: Diagram modulů implementovaného nástroje

Nástroj lze používat dvěma různými způsoby. První možností je zadat všechny argumenty (vstupní soubor(y), výstupní adresář a jednotlivé filtry s jejich parametry) při spuštění do příkazové řádky. V takovém případě se zadané filtry aplikují na vstupní data postupně za sebou, jak byly zadány v příkazové řádce – na pořadí zadávaných filtrů tedy záleží. Druhým způsobem je zadat jako argumenty vstupní soubor(y) a konfigurační soubor. V konfiguračním souboru může být více řádek s posloupnostmi různých filtrů, podobně jako je tomu u jejich zadávání do příkazové řádky. To umožňuje zpracovat stejná vstupní data více rozdílnými posloupnostmi filtrů a vyzkoušet si tak například, jak se pro jednotlivé posloupnosti filtrů liší výsledek, či určit, která posloupnost bude pro naše účely nejvhodnější. Do konfiguračního souboru si také můžeme uložit pouze jednu posloupnost filtrů, kterou pak budeme aplikovat na různá vstupní data, například za účelem jejich hromadného zpracování. Příklady spuštění skriptu a ukázkou konfiguračního souboru lze nalézt v uživatelské příručce v sekci A.1.

To, aby se jednotlivé filtry aplikovaly v pořadí, v jakém jsou zadané v příkazové řádce (nebo konfiguračním souboru) je implementováno s pomocí modulu `argparse`. Tento modul obsahuje funkce, které dokážou jednoduše rozparsovat jednotlivé zadané argumenty, zkontrolovat zda jsou validní a případně i vypsat implicitní nápovědu. Při parsování se s využitím upravené původní třídy `Namespace` uloží argumenty v zadaném pořadí do struktury slovník, kde klíčem jsou názvy funkcí a hodnotou jejich parametry. Poté se pro všechna vstupní data aplikují postupně filtry uložené v tomto slovníku. V případě konfiguračního souboru se toto parsování děje postupně pro každou jeho řádku.

Formáty vstupních dat, které tento skript podporuje, jsou vypsány v uživatelské příručce v sekci A.3. Zjednodušeně lze říci, že podporovány jsou formáty DICOM, PNG, JPG a BMP. Formát výstupního souboru je vždy stejný jako formát vstupního souboru. Jedinou drobnou výjimku tvoří DICOM, pokud se jedná o adresář obsahující sérii 2D snímků (řezů). V tomto případě se adresář načte jako 3D snímek (jeden soubor obsahující sérii řezů). V tomto formátu se dále zpracovává a je v něm i uložen. Tedy na začátku je adresář s několika soubory (řezy) a na výstupu zpracovaný jeden 3D DICOM soubor.

6.2.1 Implementované funkce

Skript má sloužit zejména k předzpracování medicínských obrazových dat, měl by tedy uživateli poskytovat základní funkčnost, kterou pro tento úkon potřebuje. Implementováno je níže popsanych 10 základních funkcí pro zpracování medicínských snímků.

Úprava kontrastu a jasu

V této funkci se využívá filtr ze `SimpleITK ShiftScaleImageFilter()`, kterému se předá hodnota, o kterou se bude upravovat kontrast, a hodnota, o kterou se posune škála hodnot pixelů – tedy úprava jasu. Pokud je hodnota pro posun škály rovna nule, nic se nestane. Pokud je zadaná hodnota záporná, snímek bude tmavší a pokud kladná, snímek bude jasnější, viz sekce 4.1. Úprava kontrastu funguje na principu násobení matice pixelů číslem (podrobně sekce 4.2). Pokud je tedy zadaná hodnota > 1 , kontrast se zvyšuje, pokud je hodnota < 1 , kontrast se snižuje. Nemá také smysl zadávat zbytečně vysoké či záporné hodnoty, výsledný snímek pak bude celý bílý či černý. Toto lze do určité míry kompenzovat právě zadáním vhodné hodnoty pro posun škály hodnot pixelů. Problém s možnými zápornými hodnotami pixelů v DICOM snímcích řeší `SimpleITK` tak, že automaticky převádí všechny hodnoty pixelů na nezáporné hodnoty.

Windowing

Windowing je poměrně důležitou metodou při zpracování medicínských obrazových dat, proto mezi naimplementovanými funkcemi v navrženém nástroji nesmí chybět. Je implementována pomocí SimpleITK filtru `IntensityWindowingImageFilter()` a uživatel jí předává dva parametry – hodnotu šířky okna a jeho úrovně. Metoda windowing i význam jejich parametrů a jejich typických hodnot jsou detailněji popsány v sekci 5.1.

Prahování se zadanými prahy

Klasická funkce pro prahování, které uživatel předá dolní a horní práh. Hodnoty pixelů mezi těmito prahy zůstanou původní a ty mimo toto rozmezí se nastaví na černou barvu. Zadávané dolní a horní prahy jsou kladná celá čísla. Filtr z knihovny SimpleITK, který je v tomto případě použit, se jmenuje `ThresholdImageFilter()`.

Prahování Otsuovou metodou

Stejně jako předchozí funkce, i tato segmentuje snímek prahováním. Uživatel ovšem žádnou hodnotu prahu nezadá. Tu totiž vypočítává sám filtr jako optimální hodnotu pro histogram snímku, jak je popsáno v sekci 5.2. Tato funkce je implementována pomocí filtru `OtsuThresholdImageFilter()`.

Vyhlazování metodou mediánu

Metoda mediánu je jednou z nelineárních vyhlazovacích filtrů, která slouží především k odstranění šumu typu sůl a pepř ze snímku, jak je popsáno v sekci 4.6.2. V tomto skriptu je implementována s pomocí filtru `MedianImageFilter()` z knihovny SimpleITK. Předává se jí jeden parametr – rádius, který určuje, jak moc vyhlazený výsledný snímek bude. Tedy čím vyšší hodnota rádiusu, tím více bude snímek vyhlazený.

Gaussovo vyhlazování

Jedná se o lineární metodu vyhlazování, která se hodí zejména na odstranění Gaussova šumu ze snímku, viz sekce 4.6.2. V navrženém skriptu funkce implementuje ze SimpleITK filtr `DiscreteGaussianImageFilter()` a předává mu dva parametry – odchylku a maximální chybu. Oba tyto parametry jsou celá kladná čísla. Čím vyšší je hodnota odchylky, tím více je snímek vyhlazen a naopak. Hodnota maximální chyby se musí pohybovat v intervalu $< 0, 1 >$. Čím blíže je tato hodnota nastavena k nule, tím vyhlazenější výsledný snímek bude.

Bilaterální vyhlazování

Zde se využívá filtr `BilateralImageFilter`. Výsledkem bilaterálního filtrování je snímek vyhlazený v homogenních oblastech, ale s zachovanými hranami. Umožňuje tak odstranit i poměrně výrazný šum bez poškození hran ve snímku. Filtru se předávají dva parametry, a to sigma oblasti a rozsahu. Jsou to kladná čísla. Čím vyšších hodnot nabývají, tím vyhlazenější snímek bude. V rámci užitečnosti tohoto filtru by však neměly být jejich hodnoty příliš vysoké, pak aplikace tohoto filtru na medicínské snímky postrádá smysl.

Doostření

Tato funkce doostří snímek s využitím SimpleITK filtru `LaplacianSharpeningImageFilter()`. Tento filtr k doostření snímku používá tzv. Laplacián. To je operátor, který umožňuje získat informaci o změnách intenzity v obraze, a díky tomu zvýraznit hrany a detaily v obraze.

Detekce hran

Pomocí filtru `CannyEdgeDetectionImageFilter()` detekuje tato funkce v DICOM snímcích hrany. Nejprve snímek vyhledá Gaussovo vyhlazováním, poté s využitím derivací určí a označí všechny potenciální hrany a nakonec aplikuje prahování, kterému uživatel určí dva prahy – horní a dolní. V sekci 4.6.3 je postup detekce hran i význam těchto dvou prahů rozebrán podrobněji.

Registrace

Registrace dvou snímků znamená nalezení geometrické transformace, která je umožní zarovnat tak, aby se snímky co nejvíce překrývaly. Je potřeba najít odpovídající body v obou snímcích a následně iteračně vhodně transformovat na sebe. Ve zpracování medicínských obrazových dat se registrace často používá pro porovnání dvou snímků stejného objektu, které byly pořízeny z různých úhlů nebo v různých časech. SimpleITK poskytuje pro registraci snímků konfigurovatelný framework (aplikační rámec) implementovaný ve třídě `ImageRegistrationMethod`. V tomto případě se při registraci snímek otáčí a posouvá, nedeformuje se. Výsledný snímek pak vypadá tak, že v místech, kde jsou objekty totožné (překrývají se) je černá barva a v místech kde se liší, je barva podle původního objektu (defaultně světle červená a zelená). [Si20a]

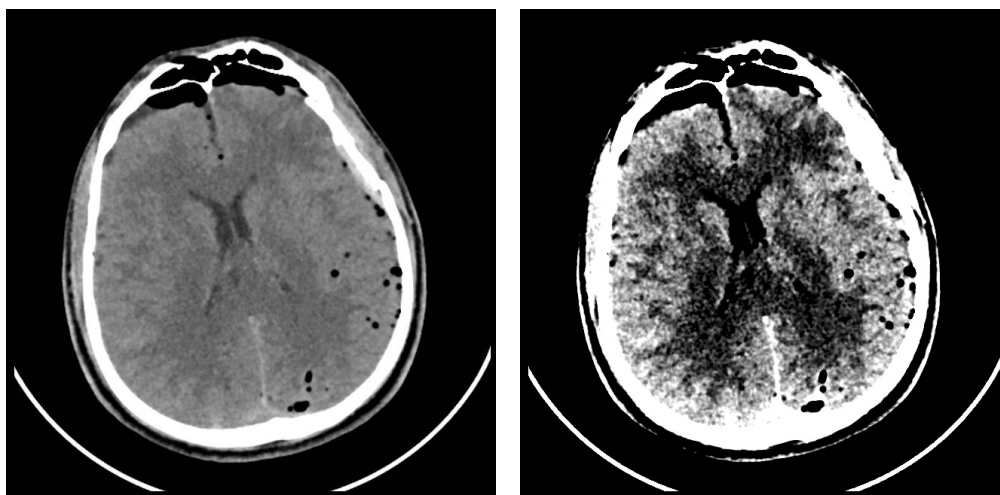
6.3 Ukázky

Ukázka 1: Windowing

Na adresář obsahující JPG soubory s CT snímky mozku je aplikována funkce windowing s šířkou okna (window width) = 50 a úrovní okna (window level) = 120. Na výsledném snímku je zřetelně vidět, že rozdíly jasu ve hmotě mozku jsou v novém snímku mnohem výraznější než v tom původním. (Obrázek 6.2) Výsledek se uloží do adresáře ukazka1.

Výpis 6.1: Příklad spuštění ukázky 1

```
1 $ python processing.py jpgdir -o ukazka1 --windowing 50 120
```



(a) Původní snímek [kag20]

(b) Nový snímek

Obrázek 6.2: Ukázka 1: Aplikace windowingu

Ukázka 2: Vyhlazování

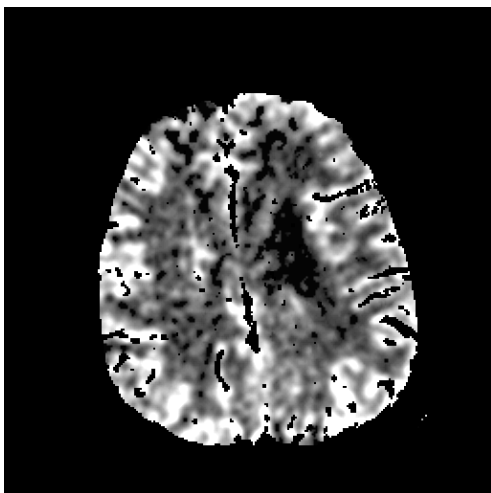
Ze snímku BMP zobrazujícího CT sken mozku se odstraní šum a „zubaté“ okraje s využitím vyhlazovacího filtru, který používá nelineární metodu vyhlazování mediánem. Rádus je nastaven na hodnotu 1. Hodnota rádusu musí být dostatečně vysoká, aby filtr odstranil nežádoucí šum, ale zároveň i přiměřeně nízká, aby nevyhladil snímek příliš. V novém snímku je patrné, že okraje jsou hladké a celkově snímek působí jemněji (Obrázek 6.3).

V dalším kroku je na snímek ještě aplikována úprava kontrastu. Posun škály hodnot pixelů je nastaven na -20 a hodnota pro úpravu kontrastu je rovna 1.2.

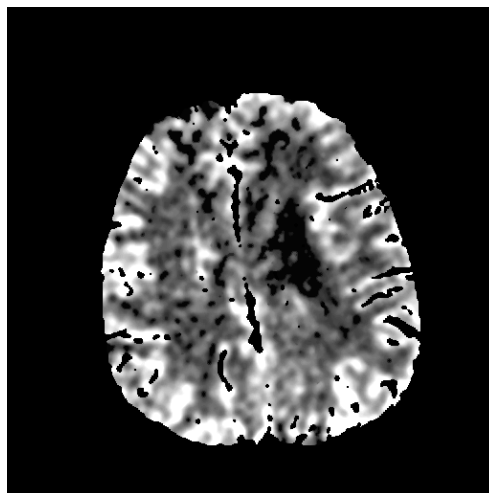
Výsledný snímek, uložený do adresáře ukazka2, je viditelně kontrastnější, rozdíly mezi hodnotami pixelů jsou více výrazné (Obrázek 6.4).

Výpis 6.2: Příklad spuštění ukázky 2

```
1 $ python processing.py image.bmp -o ukazka2 -median 1  
   -contrast -20 1.2
```

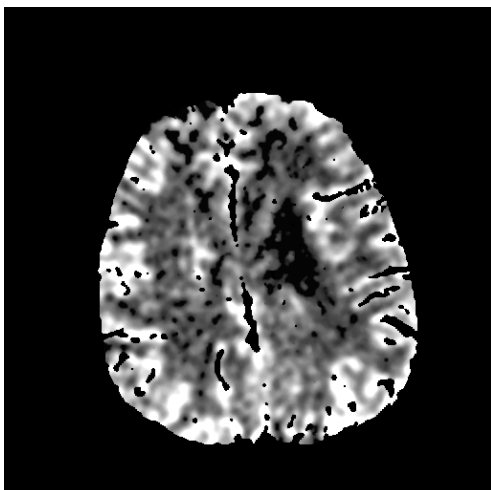


(a) Původní snímek [Mau12]

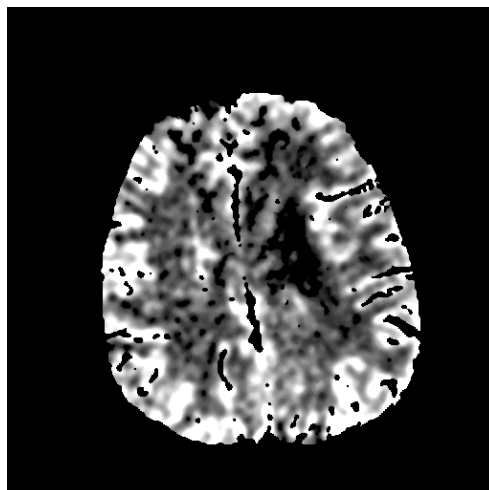


(b) Nový snímek

Obrázek 6.3: Ukázka 2: Vyhlazování metodou mediánu



(a) Původní snímek



(b) Nový snímek

Obrázek 6.4: Ukázka 2: Úprava kontrastu

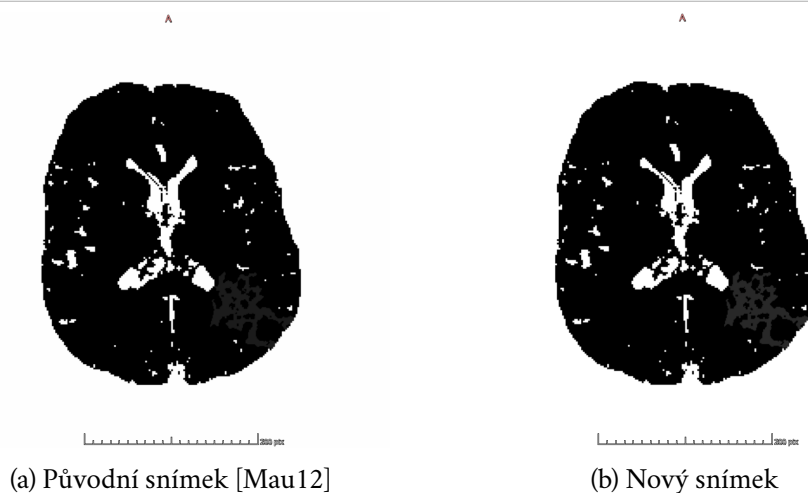
Ukázka 3: Detekce hran

V DICOM souboru s CT řezů mozku chceme detekovat hrany. Nejprve se upraví ve snímku kontrast. Hodnota pro posun škály hodnot je -20 a hodnota pro samotnou úpravu kontrastu je 1.5 . Kontrast se zde upravuje proto, aby následné prahování při detekci hran bylo přesnější. (Obrázek 6.5)

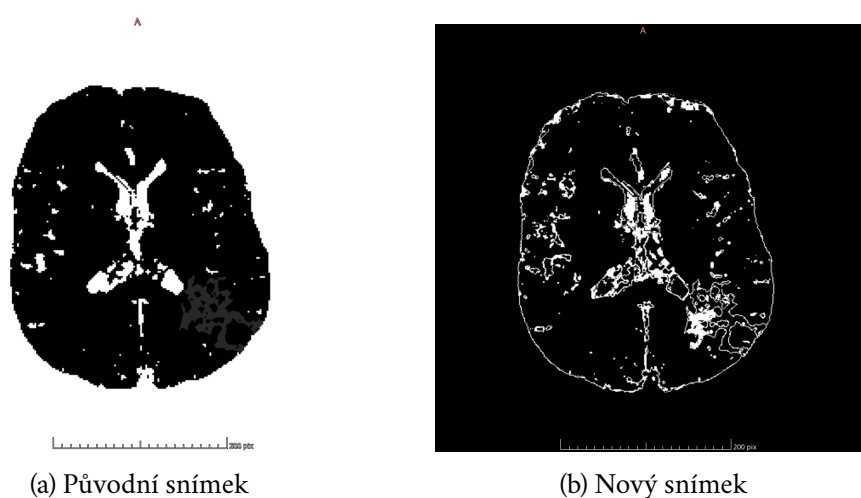
Poté se aplikuje funkce pro detekci hran, kdy dolní a horní prahy jsou nastaveny na hodnoty 40 a 60 . Ve výsledném snímku lze vidět nalezené hrany (Obrázek 6.6). Výsledek je uložen do adresáře ukazka3.

Výpis 6.3: Příklad spuštění ukázky 3

```
1 $ python processing.py image.dcm -o ukazka3 -contrast -20
   1.5 -edges 40 60
```



Obrázek 6.5: Ukázka 3: Úprava kontrastu



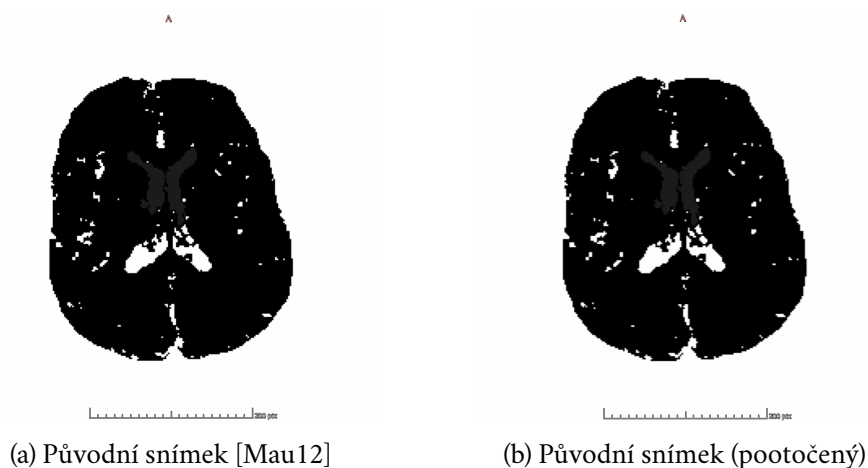
Obrázek 6.6: Ukázka 3: Detekce hran

Ukázka 4: Registrace

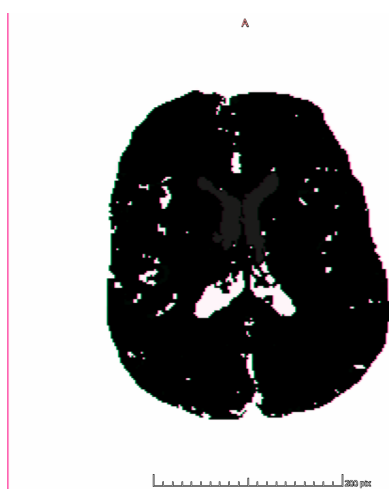
Na dva DICOM snímky mozku budeme aplikovat registraci. Ta se obvykle používá na snímky zachycující stejný objekt v různých úhlech, nebo v různých časech. V této ukázce jsou použity dva snímky lišící se pouze v tom, že jeden z nich je mírně pootočený (Obrázek 6.7). Na výsledku registrace můžeme po okrajích vidět barevné rozmazání, které ukazuje drobnou nepřesnost po krokování (Obrázek 6.8). Zároveň je vidět po jeho levém okraji růžová linie. Ta je známkou toho, že na sebe registrované snímky úplně „nesedly“, jelikož byl jeden z nich původně pootočený. Výsledný snímek je uložen v adresáři ukazka4.

Výpis 6.4: Příklad spuštění ukázky 4

```
1 $ python processing.py rotated.dcm -o ukazka4 -register image.dcm
```



Obrázek 6.7: Ukázka 4: Původní snímky



Obrázek 6.8: Ukázka 4: Registrace

6.4 Zhodnocení výsledků

V rámci praktické části této bakalářské práce byl vytvořen jednoduchý nástroj pro dávkové zpracování sady snímků dle uživatelem zadané konfigurace. Nástroj je implementován ve formě skriptu a spouští se z příkazové řádky. Uživatelská konfigurace (tedy filtry, které budou na snímky aplikovány) se zadává buďto přímo do příkazové řádky nebo do konfiguračního souboru. Nástroj tak umožňuje zpracovat jednoduše a rychle větší sady medicínských obrazových dat a také automatizaci procesu zpracování.

Nástroj je implementován v programovacím jazyce Python a ke zpracování snímků využívá filtry z knihovny SimpleITK. S použitím této knihovny byla implementace funkcí pro zpracování snímků poměrně rychlá, snadná a intuitivní. Člověk se s filtry ze SimpleITK velice rychle seznámí, pochopí jak je správně použít a jaké parametry nastavit. Ovšem u nastavování parametrů filtrů se objevila drobná nevýhoda. V dokumentaci pro SimpleITK jsou u některých filtrů nedostatečně popsány nastavované parametry. Někdy je uveden odkaz na tentýž filtr v ITK, kde je více informací, a nebo jsou názvy parametrů samovysvětlující. Jinak ale člověk musí buďto filtr dobře znát a vědět jaké parametry a jak nastavit, nebo zkoušet metodou „pokus–omyl“ jak se nastavují a zkoumat jak vlastně různé hodnoty parametru snímek modifikují.

Implementováno je 10 základních funkcí, které se obvykle při zpracování medicínských obrazových dat používají. Nástroj obsahuje především funkce sloužící k předzpracování CT snímků mozku, ale také několik funkcí, které slouží k jejich segmentaci. Umožňuje také případné snadné doplnění implementovaných funkcí o další funkce, které uživatel pro zpracování konkrétních obrazových dat potřebuje.

Nástroj pracuje s více formáty dat. Podporuje především DICOM, který se pro ukládání a manipulaci s medicínskými daty používá nejčastěji. Dále pak podporuje formáty JPG, PNG a BMP.

Funkčnost implementovaného nástroje je ověřena na datech ve formátu DICOM, BMP a JPG obsahující CT skeny mozku. Nástroj tyto data zpracoval rychle, bez problémů a s očekávanými výsledky.

Tato bakalářská práce se zabývala využitím filtrů ve zpracování medicínských obrazových dat. V krátkosti byl popsán celý postup zpracování medicínských snímků od jejich pořízení až po analýzu a interpretaci výsledků. Byly představeny nejčastěji používané metody zobrazovacích technik a detailně rozebrány vybrané filtry, jejich princip, limitace a uplatnění při předzpracování či segmentaci medicínských obrazových dat.

V praktické části byla provedena stručná analýza existujících aplikací a knihoven pro zpracování obrazových dat. Byl navržen nástroj pro dávkové zpracování sady medicínských snímků dle uživatelem zadané konfigurace. Nástroj je implementován v jazyce Python a využívá filtry z knihovny pro zpracování a analýzu obrazu SimpleITK. Je ve formě skriptu, který se spouští s parametry z příkazové řádky. Implementuje několik základních funkcí, které se běžně při zpracování medicínských obrazových dat používají. Díky zvolené architektuře je možné nástroj dále rozšiřovat o implementace dalších požadovaných způsobů zpracování či volání filtrů z knihovny SimpleITK.

V rámci bakalářské práce byly popsány digitální obrazové filtry a běžné způsoby jejich uplatnění při zpracování medicínských snímků. Byl navržen a implementován nástroj pro dávkové zpracování sady obrazů dle uživatelem zadané konfigurace a na závěr ověřena jeho funkčnost na vybrané sadě medicínských snímků. Tím jsou splněny všechny body zadání bakalářské práce.

Uživatelská příručka

A

A.1 Požadavky a spuštění programu

Požadavky

- Python 3.9
- Závislosti: SimpleITK

Instalace závislostí (SimpleITK). Způsob instalace závislostí (v tomto případě pouze SimpleITK) závisí na použité distribuci Pythonu, například:

Výpis A.1: Instalace SimpleITK pomocí modulu pip (verze >9.0)

```
1 $ pip install -r requirements.txt
```

Výpis A.2: Instalace SimpleITK pomocí správce balíků a prostředí conda/miniconda

```
1 $ conda install -c https://conda.anaconda.org/simpleitk  
SimpleITK
```

Spuštění programu

Výpis A.3: Příkaz pro spuštění programu

```
1 $ python processing.py <vstupní_data> [-o <výstupní_adresář>]
```

Skript se spouští jednoduše v příkazové řádce příkazem z Výpis A.3. Takto spuštěný skript načte vstupní data a uloží je do výstupního adresáře, pokud je tento parametr zadán, jinak do defaultního adresáře `output`. Pro aplikování filtrů se zadávají další parametry viz sekce A.2. Na jejich pořadí záleží – budou se takto aplikovat na zpracovávaná data. Program v tomto případě načte vstupní data, zpracuje je a zpracovaná data pak uloží do zadaného nebo defaultního výstupního adresáře.

Výpis A.4: Příkaz pro spuštění programu

```
1 $ python processing.py <vstupní_data> [-f <konfig_soubor>]
```

Dalším způsobem jak spustit skript, je pomocí příkazu z Výpis A.4. V tomto případě se na vstupní data aplikují filtry vypsane v konfiguračním souboru. Výstupní adresář pro uložení zpracovaných dat je také zadán v konfiguračním souboru.

```
1 --output out_addr_1 --windowing 80 40
2 --output out_addr_2 -- contrast 0 1.3 --threshold 70 130
3
4 -o out_addr3 -c 40 1 -e 80 120
5
6
```

Obrázek A.1: Příklad konfiguračního souboru

Příklady spuštění programu

Výpis A.5: Příklad spuštění programu 1

```
1 $ python processing.py image.dcm --output result --contrast
  0 1.2 --otsu
```

V tomto příkladu program načte `image.dcm`, upraví kontrast v tomto snímku, nalezne v něm hrany a výsledek uloží do adresáře `result`.

Výpis A.6: Příklad spuštění programu 2

```
1 $ python processing.py rotated.dcm --registration image.dcm
```

V tomto příkladu program načte `rotated.dcm` a `image.dcm`, provede jejich registraci a výsledek uloží do defaultního adresáře `output`.

Výpis A.7: Příklad spuštění programu 3

```
1 $ python processing.py image.jpg image.png image.bmp jpgdir
  --configfile config.txt
```

V tomto příkladu program načte postupně všechny vstupní soubory a jednotlivé snímky z adresáře `jpgdir`, aplikuje na ně filtry z konfiguračního souboru `config.txt` a výsledky uloží do adresářů definovaných v `config.txt` (například Obrázek A.1).

Nápověda

Výpis A.8: Spuštění nápovědy

```

1 $ python processing.py --help
2
3 usage: processing.py [-h] [-f <config_file>] [-o
   <output_image>] [-c <shift> <scale>] [-w <window_width>
   <window_level>] [-t <lower_threshold> <upper_threshold>]
   [-u] [-m <max_error>] [-g <variance> <max_error>] [-b
   <variance> <max_error>] [-s] [-e <lower_threshold>
   <upper_threshold>] [-r <reference_image>]
4 <input_image>
5
6 positional arguments:
7 <input_image>      image to be processed
8
9 optional arguments:
10 -h, --help          show this help message and exit
11 -f <config_file>, --configfile <config_file>
12                    config file containing direct arguments
13 -o <output_image>, --output <output_image>
14                    path for processed image
15 -c <shift> <scale>, --contrast <shift> <scale>
16                    contrast the image
17 -w <window_width> <window_level>, --windowing
18 <window_width> <window_level>
19                    window the image
20 -t <lower_threshold> <upper_threshold>, --threshold
21 <lower_threshold> <upper_threshold>
22                    threshold the image
23 -u, --otsu          threshold the image
24 -m <max_error>, --median <max_error>
25                    smooth the image using median filter
26 -g <variance> <max_error>, --gaussian <variance> <max_error>
27                    smooth the image using gaussian filter
28 -b <variance> <max_error>, --bilateral <variance>
29 <max_error>
30                    smooth the image using bilateral filter
31 -s, --sharpen      sharpen the image
32 -e <lower_threshold> <upper_threshold>, --edges
33 <lower_threshold> <upper_threshold>
34                    find edges in the image
35 -r <reference_image>, --register <reference_image>
36                    register the image to the reference image
37
38 If config file (-f/--configfile) is specified, all other
39 optional arguments are ignored.

```

A.2 Seznam funkcí a jejich parametrů

Zde je tabulka implementovaných funkcí a jejich parametrů i s informacemi o formátu dat, na které lze dané funkce aplikovat. Při zadávání těchto funkcí (jak do příkazové řádky, tak do konfiguračního souboru) záleží na jejich pořadí – v zadaném pořadí se budou aplikovat na snímek.

flag	zkratka	parametry	funkce	podporované formáty
--contrast	-c	<posun> <kontrast>	úprava kontrastu (a jasu)	DICOM, PNG, JPG, BMP
--windowing	-w	<šířka okna> <úroveň okna>	windowing	DICOM, PNG, JPG, BMP
--threshold	-t	<dolní práh> <horní práh>	prahování	DICOM, PNG, JPG, BMP
--otsu	-u		prahování (Otsuova metoda)	DICOM, PNG, JPG, BMP
--median	-m	<rádus>	vyhlazování metodou mediánu	DICOM, PNG, JPG, BMP
--gaussian	-g	<odchylka> <max error>	Gaussovo vyhlazování	DICOM, PNG, JPG, BMP
--bilateral	-b	<domain sigma> <range sigma>	bilaterální vyhlazování	DICOM, PNG, JPG, BMP
--sharpen	-s		doostření	DICOM, PNG, JPG, BMP
--edges	-e	<dolní práh> <horní práh>	detekce hran pomocí Cannyho detektoru hran	DICOM
--register	-r	<referenční snímek>	registrace: zarovnání 2 snímků na sebe	3D DICOM (série řezů)

Tabulka A.1: Přehled funkcí, jejich flagů a parametrů

A.3 Podporované formáty dat

- 2D snímek (1 řez): DICOM, PNG, JPG, BMP
- 3D snímek (série řezů): DICOM
- série řezů v adresáři: DICOM, PNG, JPG, BMP

Struktura ZIP souboru



Obsah souboru `Readme.txt` z odevzdávaného ZIP souboru:

.Text_prace

`zdrojove_soubory` – zdrojové kódy textu v \TeX u a obrázky
`bakalarska_prace.pdf` – text bakalářské práce

.Aplikace_a_knihovny

`config_1.txt` – příklad konfiguračního souboru
`config_2.txt` – příklad konfiguračního souboru
`functions.py` – modul obsahující jednotlivé filtry
`parse_arguments.py` – modul obsahující funkce pro parsování argumentů z příkazové řádky
`processing.py` – hlavní, spustitelný modul, který zpracuje data
`read_data.py` – modul obsahující funkce pro načtení vstupních dat
`requirements.txt` – soubor pro instalaci závislostí (pro pip)
`ukazky_spusteni.txt` - obsahuje příklady příkazů pro spuštění programu (ze sekce 'Ukázky' v bakalářské práci)

.Vstupni_data

`jpgdir` – adresář obsahující soubory (snímky) ve formátu JPG
`image.bmp` – snímek ve formátu BMP
`image.dcm` – snímek ve formátu DICOM (3D)
`rotated.dcm` – mírně pootočený snímek ve formátu DICOM

.Vysledky

ukazka1 – adresář s výsledky po spuštění programu

viz Aplikace_a_knihovny > ukazky_spusteni.txt

ukazka2 – adresář s výsledky po spuštění programu

viz Aplikace_a_knihovny > ukazky_spusteni.txt

ukazka3 – adresář s výsledky po spuštění programu

viz Aplikace_a_knihovny > ukazky_spusteni.txt

ukazka4 – adresář s výsledky po spuštění programu

viz Aplikace_a_knihovny > ukazky_spusteni.txt

.Readme.txt

Bibliografie

- [UME96] (US), National Research Council; MEDICINE (US) COMMITTEE ON THE MATHEMATICS, Institute of; EMERGING DYNAMIC BIOMEDICAL IMAGING, Physics of. Mathematics and Physics of Emerging Biomedical Imaging. *Chapter 3: X-Ray Computed Tomography*. 1996. Dostupné také z: <https://www.ncbi.nlm.nih.gov/books/NBK232484/>.
- [Sim23] *About SimpleITK*. 2023. Dostupné také z: <https://simpleitk.org/about.html>.
- [Ben12] BENEŠ, Petr. Světoví výrobci magnetických rezonancí. 2012. Dostupné také z: <https://otik.uk.zcu.cz/bitstream/11025/2258/1/Bakalarska%20prace%20Petr%20Benes.pdf>.
- [kag20] *Brain CT Images with Intracranial Hemorrhage Masks*. 2020. Dostupné také z: <https://www.kaggle.com/datasets/vbookshelf/computed-tomography-ct-images>.
- [Bru06] BRUCHANOV, Martin. *Diskrétní 2D konvoluce*. ČVUT Praha, 2006. Dostupné také z: <http://bruxy.regnet.cz/fel/36ACS/konvoluce.pdf>.
- [Bry19] BRYJOVA, Iveta. Zpracování medicínských a obrazových dat. 2019. Dostupné také z: https://repozitar.cz/repo/37495/Bryjova_Z.M.A.O.D.pdf.
- [Buč21] BUČEK, M. Ultrasonografie. 2021. Dostupné také z: <https://is.muni.cz/el/med/jaro2021/BRPR0422p/Ultrasonografie.pdf>.
- [TIO23] BV, TIOBE Software. *TIOBE Index for April 2023*. 2023. Dostupné také z: <https://www.tiobe.com/tiobe-index/>.
- [Dor09] DORAZIL, Jan. Analýza a segmentace tomografických obrazů. 2009. Dostupné také z: <https://core.ac.uk/download/pdf/30297223.pdf>.
- [Hla10] HLAVÁČ, Václav. Jasové geometrické transformace. *České vysoké učení technické v Praze*. 2010. Dostupné také z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/18BrightGeomTxCz.pdf>.

- [Hov09] HOVORKA, Luboš. Geometrické transformace, lineární a nelineární filtry. 2009. Dostupné také z: <https://dk.upce.cz/handle/10195/34736>.
- [Hoz03] HOZMAN, J. Základní metody předzpracování obrazu. 2003. Dostupné také z: http://webzam.fbmi.cvut.cz/hozman/Zprac_obr_prisp_kurz_UEM_3_2003.pdf.
- [Im23a] *ImageJ Docs*. 2023. Dostupné také z: <https://imagej.net/software/imagej/>.
- [Im23b] *ImageJ: Image Processing and Analysis in Java*. 2023. Dostupné také z: <https://imagej.net/ij/index.html>.
- [Kel03] KELLER, Petr. Bezkontaktní měření rozměrů zpracováním digitálního obrazu. 2003. Dostupné také z: https://dspace.tul.cz/bitstream/handle/15240/38742/U_384_S.pdf.
- [Lep19] LEPEŠKA, Martin. Projekce snímků výpočetní tomografie. 2019. Dostupné také z: https://dk.upce.cz/bitstream/handle/10195/73212/LepeskaM.ProjekceSnimku_PV_2019.pdf.
- [Mau12] MAULE, Petr. Automatická detekce nekrotické tkáně. 2012. Dostupné také z: <https://otik.uk.zcu.cz/handle/11025/5424>.
- [MBK23] MURPHY, A.; BABA, Y.; KUBE, J. Windowing (CT). 2023. Dostupné z DOI: <https://doi.org/10.53347/rID-52108>.
- [Ho22a] *CT - výpočetní tomografie*. Nemocnice na Homolce. Dostupné také z: <https://www.homolka.cz/nase-oddeleni/11635-diagnosticky-program/11635-radiodiagnosticke-oddeleni-rdg/11780-nase-sluzby/11782-ct-vypocetni-pocitacova-tomografie/>.
- [Ho22b] *MR - magnetická rezonance*. Nemocnice na Homolce. Dostupné také z: <https://www.homolka.cz/nase-oddeleni/11635-diagnosticky-program/11635-radiodiagnosticke-oddeleni-rdg/11780-nase-sluzby/11782-magneticka-rezonance/>.
- [Ho22c] *UZ - ultrazvuk*. Nemocnice na Homolce. Dostupné také z: <https://www.homolka.cz/nase-oddeleni/11635-diagnosticky-program/11635-radiodiagnosticke-oddeleni-rdg/11780-nase-sluzby/11782-ultrazvuk-sonografie/>.
- [OOM02] O., Zmeškal; O., Sedlák; M., Nežádal. Metody obrazové analýzy dat. *Ústav fyzikální a spotřební chemie, Fakulta chemická, Vysoké učení technické, Brno*. 2002. Dostupné také z: http://imagesci.fch.vut.cz/download/stud04_cesbud02.pdf.

- [Si20a] *SimpleITK - Registration Overview*. 2020. Dostupné také z: <https://simpleitk.readthedocs.io/en/v2.2.1/registrationOverview.html>.
- [Si20b] *SimpleITK Sphinx Documentation*. 2020. Dostupné také z: <https://simpleitk.readthedocs.io/en/v2.2.1/>.
- [Sli23] SLICERCOMMUNITY. *3D Slicer Image Computing Platform*. 2023. Dostupné také z: <https://www.slicer.org/>.
- [uzi20] *Stručný přehled činnosti oboru radiologie a zobrazovací metody za období 2007–2020*. Ústav zdravotnických informací a statistiky ČR. Dostupné také z: <https://www.uzis.cz/res/f/008394/nzis-rep-2021-k19-a049-radiologie-a-zobrazovaci-metody-2020.pdf>.
- [Špa12] ŠPATENKA, Pavel. Předzpracování a 3D segmentace lékařských obrazových dat. 2012. Dostupné také z: <https://otik.uk.zcu.cz/handle/11025/2653>.
- [Toe17] TOENNIES, Klaus D. *Advances in Computer Vision and Pattern Recognition*. Second. Springer, 2017. ISBN 978-1-4471-7318-2.
- [Vir+20] VIRTANEN, Pauli et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020, roč. 17, s. 261–272. Dostupné z doi: 10.1038/s41592-019-0686-2.
- [VS08] VLČEK, Vítězslav Vít; SEGETH, Karel. Matematika dokonale ukrytá v počítačové tomografii. *Pokroky matematiky, fyziky a astronomie*. 2008, roč. 53, č. 3, s. 199–210. Dostupné také z: https://dml.cz/bitstream/handle/10338.dmlcz/141859/PokrokyMFA_53-2008-3_3.pdf.

Seznam obrázků

2.1	Posloupnost operací zpracování medicínských obrazových dat	5
3.1	Princip výpočetní tomografie	8
3.2	Princip rentgenu	10
3.3	Schéma magnetické rezonance	12
3.4	Princip sonografie	13
4.1	Maticе pixelů	15
4.2	Druhy operací	16
4.3	Příklad zvýšení jasu	16
4.4	Příklad zvýšení kontrastu	17
4.5	Některé typy histogramů	18
4.6	Ukázka vyrovnání histogramu [Hoz03]	18
4.7	Pseudobarevný snímek mozku	19
4.8	Interpolace nejbližším sousedem	22
4.9	Ukázka konvoluce [Bru06]	23
4.10	Druhy šumu [Hov09]	23
4.11	Metoda rotační masky	25
4.12	Znázornění hrany v obrazové funkci a její derivace	26
5.1	Princip metody windowing	28
5.2	Příklad segmentace prahováním s prahem $t = 85$	29
5.3	Ukázka segmentace prahováním s více různými prahy [Toe17]	29
5.4	Princip dělení oblastí	30

5.5	Princip spojování oblastí	30
5.6	Segmentace narůstání oblastí s různými kritérii homogenity [Toe17]	31
6.1	Diagram modulů implementovaného nástroje	37
6.2	Ukázka 1: Aplikace windowingu	41
6.3	Ukázka 2: Vyhlazování metodou mediánu	42
6.4	Ukázka 2: Úprava kontrastu	42
6.5	Ukázka 3: Úprava kontrastu	43
6.6	Ukázka 3: Detekce hran	43
6.7	Ukázka 4: Původní snímky	44
6.8	Ukázka 4: Registrace	44
A.1	Příklad konfiguračního souboru	50

Seznam tabulek

3.1	Přehled denzit vybraných struktur [Lep19]	9
5.1	Přehled typických hodnot WW a WL [MBK23]	28
A.1	Přehled funkcí, jejich flagů a parametrů	52

Seznam výpisů

6.1	Příklad spuštění ukázky 1	41
6.2	Příklad spuštění ukázky 2	42
6.3	Příklad spuštění ukázky 3	43
6.4	Příklad spuštění ukázky 4	44
A.1	Instalace SimpleITK pomocí modulu pip (verze >9.0)	49
A.2	Instalace SimpleITK pomocí správce balíků a prostředí conda/mi- nicondy	49
A.3	Příkaz pro spuštění programu	49
A.4	Příkaz pro spuštění programu	50
A.5	Příklad spuštění programu 1	50
A.6	Příklad spuštění programu 2	50
A.7	Příklad spuštění programu 3	50
A.8	Spuštění nápovědy	51

1101001 1100001
1010110001110010 1100001
1010110101 10101 10

11010011101101001 101 101
01100001 101 101
1110001011 101 101