Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Tomáš Tůma

Segmentace cév mozku v CT angiografii

Kabinet software a výuky informatiky

Vedoucí bakalářské práce: RNDr. Josef Pelikán
Studijní program: informatika

2006

Děkuji RNDr. Josefu Pelikánovi za vedení bakalářské práce i ročníkového projektu, MUDr. Bc. Martinu Horákovi z Fakultní nemocnice Na Bulovce v Praze za četné konzultace a rady, poskytnutí dat a pomoc s orientací v problematice, Ing. Martinu Čapkovi, PhD. z Oddělení biomatematiky Fyziologického ústavu Akademie věd České republiky za e-mailové konzultace.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 31.5.2006 Tomáš Tůma

Obsah

Název práce: Segmentace cév mozku v CT angiografii
Autor: Tomáš Tůma
Katedra (ústav): Kabinet software a výuky informatiky
Vedoucí bakalářské práce: RNDr. Josef Pelikán
e-mail vedoucího: Josef.Pelikan@mff.cuni.cz

Abstrakt: Jednou z klíčových úloh při zpracování obrazových dat získaných při angiografickém vyšetření pomocí počítačové tomografie je segmentace cévních struktur. Cílem práce je usnadnit diagnostiku cév názorným zobrazením - vizualizací pomocí odečítání pre-kontrastního a post-kontrastního snímku. Hlavním problémem k řešení je nalezení použitelných algoritmů pro optimální fitting snímků. Kromě toho je vhodné seznámit se i s dalšími metodami používanými při segmentaci cév neurokrania a v případě potřeby implementovat některý z dalších algoritmů popsaný v literatuře. Data i přesnější zadání dodá v rámci již ustanovené spolupráce MUDr. Martin Horák z radiodiagnostického pracoviště Fakultní nemocnice Na Bulovce. Aplikace vytvořená v rámci práce nemusí být nutně určena pro běžné laické uživatele - naopak je žádoucí, aby zasvěcený uživatel měl možnost experimentovat a nastavovat širokou škálu parametrů ovlivňujících prováděné algoritmy. Navržené algoritmy a datové struktury je nutné dobře komentovat.

Klíčová slova: CT angiografie, registrace obrazu, mutual information

Title: Segmentation of neurocranium vessels in CT angiography
Author: Tomáš Tůma
Department: Department of software and computer science education
Supervisor: RNDr. Josef Pelikán
Supervisor's e-mail address: Josef.Pelikan@mff.cuni.cz

Abstract: One of the key tasks involved in processing of CT angiography image data is the vascular system segmentation. The aim is to provide an objective vizualization of acquired data and thus make the diagnostics easier. This work aims at studying the methods used for segmentation of neurocranium vessels and implementing one of the existing algorithms in co-operation with radiodiagnostic department of Na Bulovce faculty hospital.

Keywords: CT angiography, image registration, mutual information

Kapitola 1

Úvod

Současná medicína diponuje několika neinvazivními zobrazovacími metodami (*modalitami*), které nacházejí uplatnění při angiografickém vyšetření. Jednou z těchto modalit je počítačová tomografie (CT, computed tomography), která se používá pro zobrazení cév při diagnostice cévních uzávěrů, stenóz, cévních malformací (aneurysma, žilně-tepenný zkrat) a úrazů. Jedním ze základních úkolů počítačového zpracování pořízených dat je segmentace cévních struktur, jejímž cílem je zlepšit vizualizaci dat, a tím usnadnit jejich diagnostiku.

1.1 Princip počítačové tomografie

Počítačová tomografie (CT) je zobrazovací metoda založená na využití rentgenového záření. Pomocí algoritmů výpočetní geometrie rekonstruuje z posloupnosti dvourozměrných rentgenových snímků pořízených kolem jedné rotační osy trojrozměrný obraz zkoumaného objektu, včetně jeho vnitřní anatomie. Kromě medicíny se používá například i v oblasti nedestruktivního testování materiálů. [16]

Tzv. spirální nebo helikální CT systémy zpracovávají namísto jednotlivých samostatných řezů data získaná průběžným posunováním objektu skrz snímací pole, a generují tak přímo trojrozměrná objemová data.

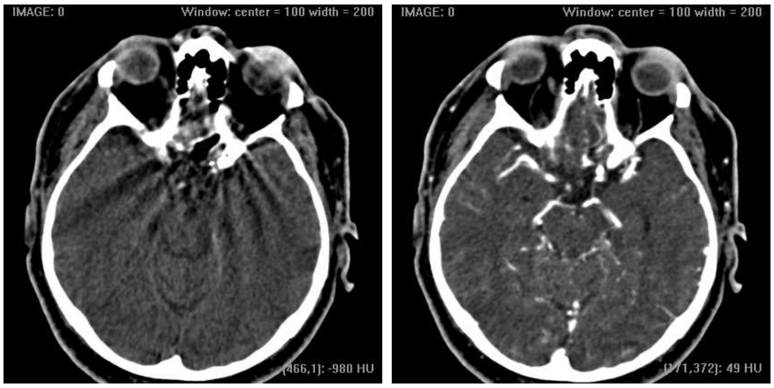
Proud dat získaný ze senzorů je během zpracování přepočítán na odhad *radiografické denzity*, která se vyjadřuje v jednotkách Hounsfieldů (HU) . [1]

Pro diagnostické účely se při CT vyšetření v některých případech používá intravenózní kontrastní médium na bázi jódu. Umožní se tak zobrazení tkání a struktur (např. cév), které by jinak bylo obtížné odlišit od okolních tkání vzhledem k podobným denzitám. Kontrastní médium může rovněž poskytnout funkční informace o tkáni.

[1] Godfrey Newbold Hounsfield vynalezl CT v roce 1972 a společně s Allanem McLeodem Cormackem získal v roce 1979 Nobelovu cenu za medicínu.

1.2 Princip angiografického vyšetření

Angiografické vyšetření pomocí CT je založeno na využití kontrastního média (obvykle látky na bázi jódu). Nejprve je pořízen *prekontrastní sken*, následně je pacientovi intravenózně podána kontrastní látka. Poté, co začne arteriální fáze distribuce kontrastního média, je sejmut *postkontrastní sken*. Diagnóza je stanovena vizuálním prozkoumáním pořízených snímků, resp. jejich počítačových rekonstrukcí.



Obrázek 1.1: Snímek prekontrastní (vlevo) a postkontrastní (vpravo) série. V pravém snímku jsou zřetelné cévy zvýrazněné pomocí kontrastního média. Oba snímky jsou zobrazeny v prostředí programu *ctseg*, v horní a dolní části jsou technické informace o zobrazení.

1.3 Motivace a cíl práce

Z charakteru počítačové tomografie vyplývá, že v pořízených snímcích jsou kromě cév zvýrazněných kontrastním médiem zobrazeny i ostatní tkáně. Zatímco málo denzní a příliš denzní tkáně lze úspěšně odstranit prahováním, nelze takto odlišit kosti od cév s kontrastním médiem. Důvodem je, že průnik intervalu denzit kostí a kontrastního média je neprazdný. Použitím jednoduché segmentace prahováním dochází k výraznému poškození zobrazení cévních struktur, respektive k nedostatečnému odstranění kostí. Při následné trojrozměrné rekonstrukci dat pak vzniká neakceptovatelné zkreslení.

V této práci chceme nalézt, prozkoumat a implementovat algoritmy, které umožní zlepšit analýzu obrazu v angiografii cév mozku. Důraz bude kladen na přehledné zobrazení vaskulárních struktur neurokrania. Pro vybranou skupinu algoritmů chceme vyvinout aplikaci, která umožní další experimentování s jejich parametry a napojení vzniklých modulů do procesu zpracování dat na radiologickém pracovišti. Tato aplikace však nesmí být používána v klinické praxi a jejím účelem je pouze podpora výzkumu a vývoje.

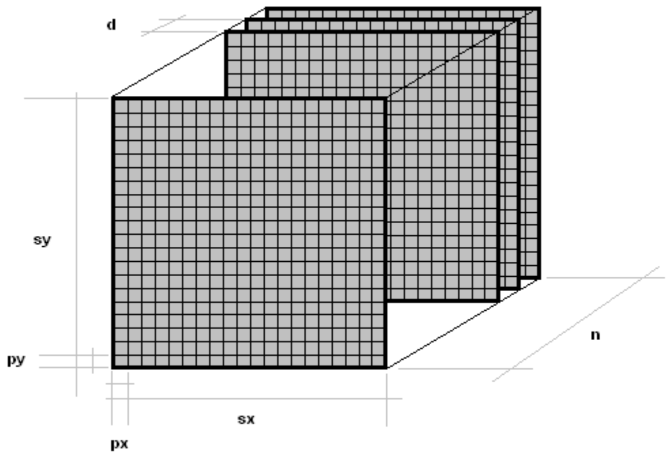
V kapitole 2 nejprve uvedeme charakter dat, která jsou vstupem algoritmů včetně možností jejich vizualizace. Následuje stručná klasifikace algoritmů pro segmentaci cév (kapitola 3). V kapitole 4 se podrobněji zabýváme algoritmy, které tvoří jádro subtrakčního přístupu k segmentaci. Tento přístup se ukazuje jako zajímavé a elegantní řešení pro aplikační doménu angiografie mozku. Kapitola 5 uvádí postup navržený pro odstranění artefaktů způsobených nepřesnou registrací. Následují kapitoly o implementaci subtrakčních algoritmů a užívání vzniklé aplikace.

Kapitola 2

Vstupní a výstupní data

2.1 Logická struktura dat

Prostorová organizace Struktura dat, se kterými pracujeme, je určena principem snímání a následné tomografické rekonstrukce dat, kterou provádí firmware CT systému. Jeho výstupem je posloupnost transverzálních *řezů* (anglicky *slice*), která se obvykle nazývá *série*. Jednotlivé řezy jsou reprezentovány dvourozměrnou maticí denzit a představují odhad denzit v rovině kolmé na směr snímání (osu pacienta).



Obrázek 2.1: Schéma organizace vstupních dat

Základní parametry, které určují geometrii a sémantiku této reprezentace, uvádíme v tabulce a náčrtku.

Parametr	Označení	Typická hodnota
rozměry řezů	sx, sy	512 x 512 pixelů
velikost pixelů jednoho řezu	px, py	0.5 mm x 0.5 mm
vzdálenosti mezi řezy	d	0.75 mm - 4 mm

V praxi omezujeme tyto parametry tak, že všechny řezy mají stejný rozměr a všechny vzdálenosti mezi řezy jsou stejné. Navíc budeme předpokládat, že rozměry řezů jsou v obou osách stejné (řezy jsou čtvercové).

Z dvourozměrných snímků můžeme sestavit trojrozměrná objemová data, která jsou výčtovou reprezentací snímaného trojrozměrného prostoru.

Souřadnicový systém Volba souřadnicového systému obvykle respektuje výše uvedenou geometrii dat. Osy x, y jsou rovnoběžné s vodorovnou, resp. svislou osou řezů, kladná poloosa z indexuje jednotlivé řezy. Počátek je umístěn do počátku souřadnicového systému prvního řezu série. Při vizualizaci je nutné dbát na vhodnou konverzi mezi orientací souřadného systému série a výstupního zařízení.

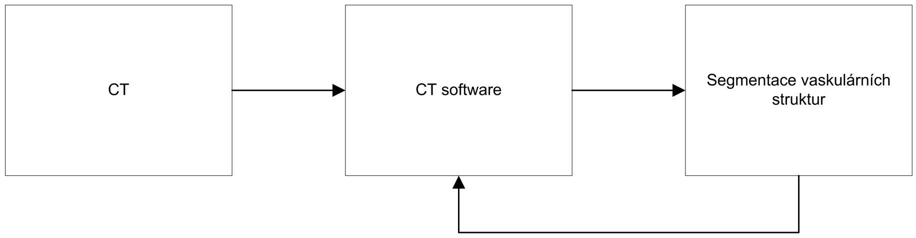
Souřadnicové systémy je vhodné dále zkoumat podle těchto kritérií [2]:

Metrika O *metrickém souřadnicovém systému* hovoříme tehdy, pokud souřadnice respektují skutečné vzdálenosti mezi naměřenými vzorky. Takový systém je nutné uvažovat v případě, kdy používáme zároveň data pořízená s různými snímacími parametry (např. vzdáleností mezi řezy), případně při vizualizaci, kdy je žádoucí zachovat skutečné proporce zobrazovaného objektu. Pro další výpočty se obvykle provádí konverze do univerzálního nemetrického souřadnicového systému s isotropickými voxely. Souřadnice v *nemetrickém systému* reprezentují přímo (celočíselné) pozice voxelů v trojrozměrné matici dat příslušné série.

Spojitost Data jsou diskrétní povahy a jsou reprezentována trojrozměrnou maticí. Přirozeně se tedy nabízí používání *diskrétního souřadnicového systému.* V případě potřeby přechodu ke spojitému souřadnicovému systému reprezentují souřadnice $[x, y, z]$ geometrický střed voxelu s indexem $[x, y, z]$. Jeho rozměr je v případě metrického systému dán konkrétními hodnotami $sx * sy * d$ (voxel může být anisotropický), v případě nemetrického systému se jedná o krychli o hraně délky 1.

2.2 Konkrétní souborový formát

Vzhledem k zapojení programu do procesu zpracování tomografických dat (ilustrováno na obrázku 2.2) vzniká požadavek na podporu univerzálního souborového formátu pro vstupní a výstupní data.



Obrázek 2.2: Předpokládané zapojení implementovaných segmentačních algoritmů do procesu zpracování dat. Jako primární software pro vizualizaci a práci s daty zůstává originální vybavení radiodiagnostického pracoviště, algoritmy pro segmentaci se včlení do řetězce ve formě dodatečného zpracování.

V současné době je de facto standardem v oblasti medicínského zobrazování DICOM (Digital Imaging and Communications in Medicine). Tento standard vznikl v roce 1993 jako následník nepříliš úspěšného standardu ACR-NEMA a v roce 1995 byl přijat jako formální standard v Evropě. Je řešením, které se snaží sjednotit různorodé medicínské zobrazovací systémy a definuje nejen souborové formáty, nýbrž i protokol pro komunikaci mezi DICOM zařízeními [10]. Některé části standardu DICOM jsou implementovány v open-source knihovnách a lze je vhodně použít pro komunikaci s ostatním software v této oblasti. Podrobnosti implementace jsou uvedeny v části 6.3.

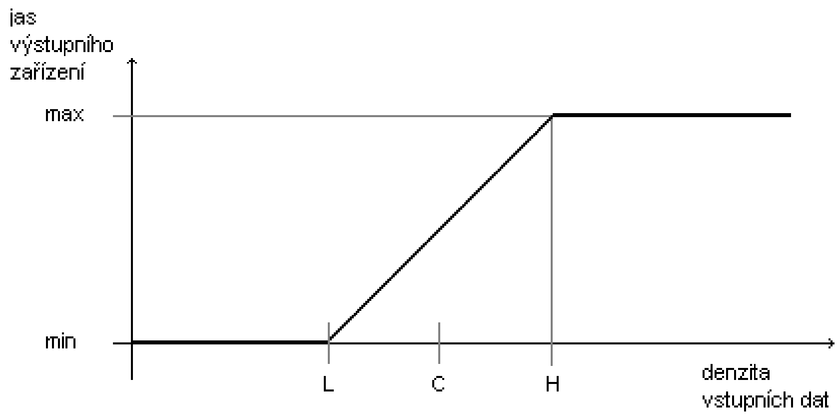
2.3 Vizualizace dat

Přehled metod pro vizualizaci objemových dat lze nalézt např. v [17]. Zde uvádíme tři základní metody, které reprezentují různé přístupy k vizualizaci medicínských dat:

- 2D zobrazení bez 3D rekonstrukce
- 3D metody nehledající povrch (multiplanar reformation(MPR), MIP)
- hledání izoploch, volume rendering (VR)

2.3.1 Procházení řezů

Procházení jednotlivých řezů série je základním způsobem, jak zobrazit pořízená data. Jeho výhodami jsou rychlost a přesnost. Stejně jako u ostatních zobrazovacích metod je však nutné vyřešit reprezentaci hodnot voxelů (denzit) na obrazovce počítače. Zatímco hodnoty denzit se v modalitě CT obvykle nacházejí v rozsahu -1024 až 3071 HU, na bežných zařízeních je k dispozici pouze 256 odstínů šedi. Tato situace vyžaduje použití prahování (resp. poloprahování, viz [4]). Rozložení denzit v zobrazovaných snímcích se liší nejen podle modality, ale také podle vyšetřované části těla a konkrétního pacienta. Obvykle se proto volí manuální nastavení prahů s tím, že k dispozici je sada předvoleb, které uživatel upraví podle konkrétní situace. Pro transformaci zvoleného rozsahu denzit do intervalu 0 až 255 (hodnoty jasu) postačuje použít lineární zobrazení.



Obrázek 2.3: Transformace denzit vstupních dat do rozsahu výstupního zařízení pomocí prahování. Hodnoty spadající do zvoleného intervalu (L,H) jsou transformovány pomocí lineárního zobrazení. Častý způsob zadávání parametrů prahování v medicínském softwaru je pomocí dvojice střed (C) a šířka (H-L)

Implementace Protože procházení řezů je obvykle základním nástrojem používaným pro počáteční orientaci v datech i důkladnou diagnostiku, je nutné, aby zobrazování řezů fungovalo rychle a plynule. V programu proto pro každý snímek, který patří do zobrazované série, udržujeme prahovaný obraz transformovaný do stupňů šedi. Vhodně zvolený formát tohoto bufferu umožňuje navíc velmi rychlé vykreslení pomocí funkce SetDIBBits ([9]) platformy Win32.

2.3.2 Metody nehledající povrch

V medicínském zobrazování je populárním způsobem trojrozměrné rekonstrukce metoda MIP (*maximum intensity projection*). Funguje na principu *vrhání paprsku.* Označíme-li J množinu voxelů na dráze paprsku a f obrazovou funkci, pak hodnotou pixelu, kterým paprsek prochází, je

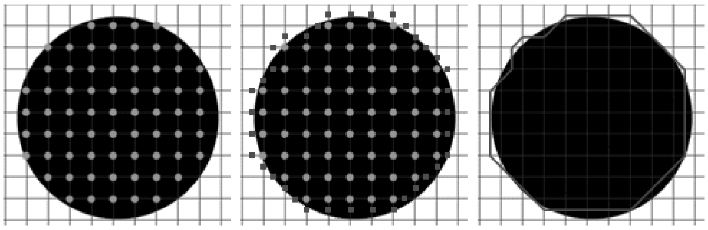
$$\max_{v \in J} f(v)$$

Často se používá v modifikaci *average intensity projection*, tj.

$$\frac{\sum_{v \in J} f(v)}{|J|}$$

2.3.3 Marching cubes

Algoritmus marching cubes [7] (také např. v [17]) se používá pro vizualizaci izoploch (povrchů s konstantní hodnotou) v objemových datech. Tyto povrchy aproximuje sítí trojúhelníků a umožňuje tak jejich zobrazení pomocí běžné grafické knihovny, např. OpenGL [11]. Algoritmus je příkladem použití divide-and-conquer paradigmatu, protože problematiku nalezení izoplochy štěpí na problém nalezení průsečíku izoplochy s jedním voxelem.



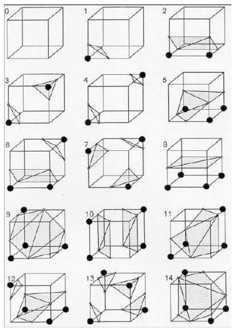
Obrázek 2.4: Demonstrace principu algoritmu na dvourozměrném případě (obrázky [14])

Základní princip lze velmi názorně demonstrovat na dvourozměrném případě [14]. Uvažujme jako vstup algoritmu obrázek 2.4 a prahovou hodnotu T takovou, že pixely s menší hodnotou než T tvoří diskrétní reprezentaci kruhu. Naším úkolem je nalézt aproximaci "povrchu" kruhu, v tomto dvourozměrném případě tedy aproximaci kružnice, která jej ohraničuje. Zvýrazníme-li pixely, jejichž hodnota je menší než T, vidíme, že kružnici můžeme vytvořit spojením těch zvýrazněných pixelů, které jsou na hranici zvýrazněné oblasti. Ostatní pixely nebudou mít na podobu hranice vliv. Pokud se neomezíme pouze na to, že vrcholy hranice leží v diskrétním souřadnicovém systému, dosáhneme přesnější aproximace hledáním průsečíku aproximovaného kruhu s jednotkovým čtvercem pixelové sítě (viz dále).

Obrázek procházíme po jednotlivých čtvercích, které jsou tvořeny vždy čtyřmi sousedními pixely. Každý z vrcholů tohoto čtverce ohodnotíme podle toho, jestli je jeho hodnota menší než T, resp. větší (pixel je uvnitř, resp. venku). Pro každý ze čtyř vrcholů máme dva stavy, celkem tedy 16 možných konfigurací. Využitím symetrie čtverce můžeme některé stavy eliminovat a pro každý ze zbylých stavů určit hrany, kterými prochází hledaný povrch. Pro stavy, v nichž mají všechny vrcholy stejné ohodnocení, platí, že jimi povrch neprochazí - zkoumaný čtverec můžeme jednoduše vynechat. Ve zbylých případech určíme průsečík povrchu s hranou. Na ilustračním obrazku je použito zjednodušení, kdy průsečík umístíme vždy do poloviny hrany. V praxi se používá lineární interpolace. Autoři [7] zmiňují experimenty s interpolací vyšších řádů, které však prokázaly nepatrné zlepšení. Spojením takto nalezených průsečíků získáváme výslednou aproximaci (izoplochu).

Přenesení problému do trojrozměrného prostoru je přímočaré. Jednotkovým problémem je hledání průsečíku izoplochy s jednotkovou krychlí (odtud název algoritmu). V případě krychle existuje 256 konfigurací, které lze však s využitím symetrie krychle snížit na pouhých 15 případů. Tyto případy jsou vyznačeny na obrázku. Každá krychlička vygeneruje 0 až 5 trojúhelníků.

Nevýhodou algoritmu Marching cubes je velká paměťová a časová náročnost. V některých případech dochází k nesprávnému odhadu tvaru izoplochy ([17]).



Obrázek 2.5: 15 základních konfigurací algoritmu marching cubes

Kapitola 3

Segmentace

Segmentací obrazu rozumíme „rozčlenění obrazu do částí, které mají úzkou souvislost s předměty či oblastmi reálného světa zachyceného na obraze“ [4]. Formálněji, segmentací obrazu R nazveme konečnou neprázdnou množinu oblastí $\{R_1, R_2, \ldots, R_n\}$, pro kterou platí:

$$\bigcup R_i = R$$

Segmentace cév je jednou ze základních úloh počítačového zpracování medicínských dat. V současné době existuje řada metod, které se touto úlohou zabývají. Vzhledem k různorodosti vstupních dat a požadavků na výstup však zatím neexistuje žádný univerzální přístup, který by poskytnul uspokojivé řešení ve všech aplikačních doménách. Důležitým hlediskem, které o nasazení metody často rozhoduje, je kromě korektnosti a robustnosti také míra automatizace segmentačního procesu a časová složitost algoritmů. V [5] je podán obecný přehled metod pro extrakci cév, v [1] je možné nalézt přehled vybraných skupin segmentačních metod.

V této kapitole uvedeme obecný přehled existujících algoritmů. Neuvádíme odkazy na konkrétní software pro medicínskou segmentaci, který zpravidla implementuje speciální kombinaci algoritmů, jejichž přehled následuje. Open-source implementaci řady z uváděných postupů lze nalézt v projektu ITK (Insight Segmentation Toolkit).

3.1 Metody pro extrakci cév

Metody pro extrakci cév můžeme podle [5] rozdělit do následujích kategorií (podrobnosti a odkazy na konkrétní autory lze nalézt v [5]):

Rozpoznávání vzorů (pattern recognition) Jde o techniky pro automatickou klasifikaci a rozpoznávání objektů v obraze, které se často řadí také do oblasti počítačového vidění. Patří sem segmentace narůstáním oblastí, geometrické metody založené na hledání osy cévy (centerline detection) a tvorby kostry cévní struktury (skeletonization), na diferenciální geometrii, aplikaci morfologických operátorů (eroze, dilatace) a další.

Přístupy založené na modelech Vaskulární struktury jsou segmentovány pomocí aplikace explicitního modelu zkoumané cévní struktury. Lze tak využít apriorní znalosti zkoumané cévní struktury, kterou má operátor, společně s technikami pro automatizovanou analýzu obrazu. Geometrický model struktury, která je předmětem zájmu, je umístěn do blízkosti její reprezentace v pořízených datech. Následně je tento model automaticky deformován tak, aby co nejlépe aproximoval skutečný tvar objektu. Deformace je řízena např. na bázi minimalizace energie.

Sledování cév Algoritmy pro sledování cév vycházejí z interaktivně zadaného vstupního bodu a postupně konstruují osu či hranici sledované cévy analýzou hodnot pixelů v rovině kolmé na směr cévy. Výhodou je, že se jedná o lokální metodu, která pracuje pouze s malým množstvím dat, která jsou předmětem zájmu. Nevýhodou je značná míra interaktivity.

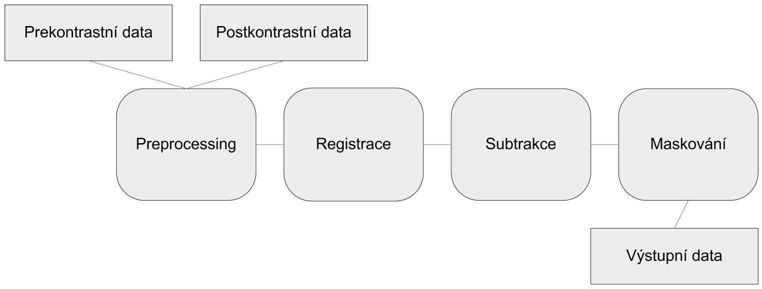
Umělá inteligence Zatím se používá převážně pro 2D segmentační úlohy. Apriorní znalost segmentovaných struktur je vyjadřena pomocí množiny pravidel popisujících segmentované objekty (u cév jde např. o charakterizaci osy, tvaru cévy, jejího okolí, ...), která řídí proces segmentace a aplikaci nízkoúrovňových algoritmů (např. morfologické operátory, prahování).

Neuronové sítě Výhodou použití neuronových sítí v analýze obrazu je jejich schopnost učení během tréninku, kdy mohou získat složitá pravidla pro segmentaci cévních struktur. Jejich nasazení obvykle zahrnuje učení na připravené množině dat a následné použití pro analýzu skutečných dat. Úspěšnost je do značné míry závislá na vhodně zvolených tréninkových datech.

3.2 Segmentace odčítáním

Myšlenka segmentace cév odčítáním se poprvé objevila v *subtrakční angiografii*. Před zavedením kontrastního média byl pořízen prekontrastní snímek, jehož (manuálním) odečtením od postkontrastních snímků bylo možné eliminovat z obrazu denzní tkáně a zviditelnit tak cévní struktury. Moderní analogie postupu se v současnosti používá v digitální subtrakční angiografii (DSA, Digital subtraction angiography) a souvisejících metodách (Rotational DSA, 3D DSA). Základním aspektem, který ovlivňuje funkčnost této skupiny metod, je nalezení korektní registrace prekontrastního a postkontrastního snímku.

V následujícím textu se budeme zabývat aplikací subtrakčního algoritmu pro segmentaci cév mozku v CT angiografii. Tato varianta CTA bývá někdy označována jako DS-CTA (Digital subtraction CTA). Algoritmus se skládá z následujících modulů (jejich propojení je vyznačeno na obrázku), z nichž některé jsou podrobněji popsány v dalších kapitolách textu.



Obrázek 3.1: Schéma subtrakčního algoritmu pro segmentaci cév

Preprocessing - prahování Před dalším zpracováním dat (registrací, subtrakcí) může být výhodné provést počáteční segmentaci pomocí jednoduchého prahování a odstranit tak méně denzní tkáně. Tato operace má význam tehdy, pokud chceme zachovat skutečné hodnoty denzit segmentovaných cév. Pokud neprovedeme prahování, při odečtení budou denzity cév zvýrazněných kontrastní látkou sníženy o denzitu na odpovídajícím místě v prekontrastním snímku. Dalším aspektem je, že např. transformaci povrchu pokožky nelze aproximovat rigidní transformací. Odečtením tak mohou ve výsledném obraze vzniknout nežádoucí artefakty.

Registrace Cílem registračního algoritmu je nalézt co nejlepší prostorový alignment prekontrastních a postkontrastních dat. Jedna ze sérií je obvykle označena jako referenční (reference) a druhá jako plovoucí (floating). Registrací rozumíme nalezení optimální transformace plovoucích dat do prostoru referenčních dat. Registraci je věnována kapitola 4.

Subtrakce Subtrakcí rozumíme odečtení prekontrastních dat od postkontrastích dat, které probíhá voxel po voxelu. Záporné hodnoty jsou nahrazeny minimální denzitou.

Maskování Vzhledem k technologii snímání není možné nalézt optimální regsitraci prekontrastních a postkontrastních dat. Vlivem diskrétní reprezentace dat a partial volume effectu jsou v odečteném obraze přítomny artefakty, které znepřehledňují vizualizaci a diagnostiku. Maskování je jeden z algoritmů, který řeší tento problém s využitím informací v obou data setech.

Kapitola 4

Registrace

Registrace je proces, který umožňuje integrovat informace obsažené ve dvou obrazech tím, že hledá jejich optimální prostorové ztotožnění.

4.1 Klasifikace algoritmů pro registraci

Klasifikace registračních metod používaných v medicínském zobrazování je podrobně uvedena např. v [8]. Autoři zavádí devět klasifikačních kritérií:

Dimenze Nejčastějším případem je v současnosti registrace dvou 3D objemů, objevuje se však i 2D/3D registrace (rentgenový 2D snímek pořízený během operace se registruje oproti předoperačním CT datům) a 2D/2D registrace (např. v subtrakční angiografii).

Podstata registrace

Nevlastní (extrinsic) Nevlastní registrační metody jsou založené na použití cizích objektů v prostoru obrazu. Obvykle jde o připevnění různých speciálních objektů (což může a nemusí mít invazivní charakter), které pak slouží k určení rigidní transformace.

Vlastní (intrinsic) Registrace vychází z obrazových dat pořízených pouze snímáním pacienta. Její podstatou může být identifikace významných (charakteristických) bodů v registrovaných obrazech (tzv. landmarků), případně se může jednat o registraci segmentovaných objektů (např. povrch pokožky při registraci snímků hlavy), jejichž vlastnosti jsou známé. Nejnovějším přístupem je použití algoritmů, které určují registraci pouze na základě hodnot voxelů (voxel property based methods). Může se jednat o metody *reduktivního* charakteru, kdy je registrace převedena například na registraci těžiště a os obrazu vypočtených statistickými metodami, nebo lze využít celých obrazových dat. Výhodou je velká flexibilita, protože algoritmy nevyžadují žádné konkrétní znalosti o obsahu obrazu. Nevýhodou je velká prostorová i časová složitost těchto algoritmů.

Typ transformace

Rigidní Pouze translace a rotace.

Afinní Rovnoběžné přímky se transformují na rovnoběžné přímky.

Perspektivní Přímky se transformují na přímky, ale nemusí být zachována jejich rovnoběžnost.

Elastická Přímky se transformují na křivky.

Interaktivita Z hlediska interakce lze registrační metody rozdělit na plně automatické, poloautomatické (semi-automatic), kde uživatel algoritmus inicializuje či upravuje průběh výpočtu, a interaktivní, kde software zprostředkovává především reprezentaci, resp. vizualizaci dat. Pro životaschopnost algoritmu jako kandidáta pro použití v klinické praxi je důležitá vysoká míra automatizace.

Optimalizační procedura Parametry transformace mohou být explicitně vypočítány nebo nalezeny jako optimum nějaké funkce definované na prostoru parametrů transformace. V druhém případě vzniká potřeba využít některou z existujících optimalizačních metod, jejíž charakter obvykle určuje funkce, kterou chceme optimalizovat. V oblasti registrace medicínských dat se nejčastěji objevuje Powellova optimalizace, Downhill Simplex metoda, genetické metody, ASA, a další.

Modality Registrace může být monomodalitní (registrovaná data pocházejí v obou případech ze stejné modality) nebo multimodalitní (kombinace doplňujících se informací z různých modalit, typicky CT a MR snímek neurokrania). V některých případech může být modalitou např. i anatomický model/atlas. Nejčastěji registrovanými modalitami jsou CT, MR, PET, SPECT, a další.

Subjekt O *intrasubject* registraci hovoříme v případě, že všechna data v registračním procesu pocházejí od jednoho pacienta. Vstupem *intersubject* registrace jsou data více pacientů nebo data pacienta a modelu, resp. atlasu.

Objekt Objektem je v medicínské registraci zkoumaná část lidského těla.

4.2 Registrace v CTA

Z hlediska klasifikačních kritérií zavedených v odstavci 4.1 můžeme úlohu registrace dat pro subtrakční CT angiografii popsat následovně:

Klasifikační kritérium	**Kategorie**
Dimenze	3D/3D
Podstata registrace	Vlastní (intrinsic)
Typ transformace	Rigidní
Interaktivita	Automatická, případně semiautomatická
Optimalizační procedura	Hledání optima pomocí Powellovy metody
Modality	Monomodalitní CT
Subjekt	Intrasubject
Objekt	Cévy neurokrania

4.2.1 Matematické vyjádření rigidní transformace

Homogenními souřadnicemi bodu v prostoru s kartézskými souřadnicemi $[X, Y, Z]$ nazveme uspořádanou čtveřici čísel $[x, y, z, w]$, jestliže platí

$$X = \frac{x}{w}, Y = \frac{y}{w}, Z = \frac{z}{w}, w \neq 0$$

Homogenní souřadnicový systém je výhodný pro reprezentaci rigidní transformace, protože umožňuje vyjádřit translaci i rotaci jednotným způsobem pomocí matice. [17]

Lineární transformaci bodu $P = [x, y, z, w]$ na bod $P' = [x', y', z', w']$ vyjádříme pomocí matice A o rozměrech $4 * 4$

$$P' = P * A$$

tedy

$$\begin{pmatrix} x' & y' & z' & w' \end{pmatrix} = \begin{pmatrix} x & y & z & w \end{pmatrix} * \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

Souřadnice bodů reprezentujeme při maticových transformacích řádkovým vektorem $1 * 4$. Složení lineárních transformací A, B, C v uvedeném pořadí (nejprve se provede A, pak B, a nakonec C) získáme jako součin $A * B * C$. Plyne to z asociativity maticového součinu:

$$(((P * A) * B) * C) = P * (A * B * C)$$

Rotace kolem osy x Rotaci (otočení) bodu kolem osy x o úhel α odpovídá matice

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotace kolem osy y Rotaci (otočení) bodu kolem osy y o úhel β odpovídá matice

$$R_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotace kolem osy z Rotaci (otočení) bodu kolem osy z o úhel γ odpovídá matice

$$R_z(\gamma) = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Inverzní transformace k rotacím získáme podle následujících vztahů:

$$R_x(\alpha)^{-1} = R_x(-\alpha)$$

$$R_y(\beta)^{-1} = R_y(-\beta)$$

$$R_z(\gamma)^{-1} = R_z(-\gamma)$$

Translace Translaci (posunutí) bodu určené vektorem (T_x, T_y, T_z) odpovídá matice

$$T(T_x, T_y, T_z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}$$

Inverzní transformaci získáme podle vztahu

$$T(T_x, T_y, T_z)^{-1} = T(-T_x, -T_y, -T_z)$$

4.2.2 Implementace rigidní transformace

Jednu z registrovaných sérií označme jako referenční (R) a druhou jako plovoucí (F). Rigidní transformaci z F do R označme jako A a vektor jejích parametrů $\phi = (\phi_x, \phi_y, \phi_z, \phi_\alpha, \phi_\beta, \phi_\gamma)$. Platí

$$A(\phi) = C_F * R_x(\phi_\alpha) * R_y(\phi_\beta) * R_z(\phi_\gamma) * T(\phi_x, \phi_y, \phi_z) * C_R^{-1}$$

Jak uvádíme v části 2.1, počátek souřadnicového systému referenčních i plovoucích dat je umístěn v jednom z rohů kvádru, který vymezuje jejich objem. Aby byla parametrizace pomocí úhlů ϕ_x, ϕ_y, ϕ_z intuitivní, před provedením samotné rigidní transformace aplikujeme translaci C_F, která přenese počátek souřadnicového systému plovoucích dat do jejich geometrického středu. Po dokončení transformace do prostoru referenčních dat musí být analogicky provedena inverzní translace, tentokrát v prostoru referenčních dat (C_R^{-1}).

V obecném případě mohou být voxely registrovaných dat *anisotropní* (nemají tvar krychle) a jejich rozměry v plovoucích a referenčních datech se mohou lišit. V tom případě musí být v transformační matici obsažena korekce anisotropie a rozměrů voxelů. Tuto problematiku je nutné řešit zejména v případě multimodalitní registrace. V DS-CTA můžeme obvykle předpokládat, že plovoucí i referenční data mají voxely stejných velikostí a proporcí, takže je možné korekci vynechat.

4.2.3 Interpolace

Nechť $X \in F$ je bod v plovoucích datech. Pak jeho transformace $X * A(\phi)$ nebude v obecném případě incidovat s celočíselným souřadnicovým systémem referenčních dat. Je tedy nutné implementovat *interpolaci*, abychom zjistili pro účely dalšího porovnávání hodnotu referenčních dat v bodě $X * A(\phi)$.

Označme $V_{000}, V_{001}, \ldots, V_{111}$ voxely sousedící s bodem $Y = X * A(\phi)$ tak, že V_{ijk} je bod na souřadnicích $[\lfloor Y_x \rfloor + i, \lfloor Y_y \rfloor + j, \lfloor Y_z \rfloor + k]$.

Metoda nejbližšího souseda

Označíme-li d vzdálenost dvou bodů v souřadnicovém systému, pak při interpolaci metodou nejbližšího souseda (tzv. interpolace nultého řádu) platí pro hodnotu referenčních dat R

$$R(X * A(\phi)) = R(\arg_{i,j,k=0,1} \min(d(V_{ijk}, X * A(\phi))))$$

Vyjádřeno srozumitelněji, hodnotou v interpolovaném bodě je hodnota v nejbližším z bodů $V_{000}, V_{001}, \ldots, V_{111}$.

Trilineární interpolace

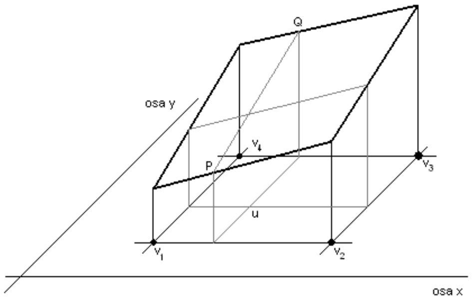
Trilineární interpolace je založena na postupném trojím využití lineární interpolace. *Lineární interpolací* hodnoty v bodě C se sousedními body A, B je

$$h(C) = h(A) + (h(B) - h(A)) * \frac{|C - A|}{|B - A|}$$

Speciálně, pokud vzdálenost $|B - A| = 1$ a nahradíme $|C - A| = C_f$, kde $C = \lfloor C \rfloor + C_f$, můžeme psát

$$h(C) = (1 - C_f) * h(A) + C_f * h(B)$$

Bilineární interpolace je aplikace lineární interpolace v dvourozměrném prostoru. Hodnoty interpolované funkce aproximuje pomocí po částech lomené plochy. Nechť interpolovaný bod je U a sousední body jsou V_1, V_2, V_3, V_4. Souřadnicové osy označme x, y. Nejprve lineární interpolací hodnot v bodech V_1, V_2 v ose x získáme hodnotu v bodě $P = [U.x, V_1.y]$ a interpolací hodnot ve V_3, V_4 v ose x získáme hodnotu v bodě $Q = [U.x, V_3.y]$. Nyní lineární interpolací hodnot v P, Q podle osy y získáme hodnotu v bodě U.



Obrázek 4.1: Ilustrace k popisu bilinerní interpolace

Trilineární interpolace sestává z aplikace bilineární interpolace (dvakrát, pro dvě protější stěny krychle) a lineární interpolace (ve třetí ose). Vrátíme-li se ke značení ze začátku odstavce o interpolaci, a označíme-li neceločíselné části souřadnic bodu Y jako χ_x, χ_y, χ_z, tedy

$$Y_x = \lfloor Y_x \rfloor + \chi_x$$
$$Y_y = \lfloor Y_y \rfloor + \chi_y$$

$$Y_z = \lfloor Y_z \rfloor + \chi_z$$

pak interpolovaná hodnota v bodě Y je

$$\begin{aligned}
&V_{000} * (1-\chi_x) * (1-\chi_y) * (1-\chi_z) + \\
&V_{100} * \chi_x * (1-\chi_y) * (1-\chi_z) + \\
&V_{010} * (1-\chi_x) * \chi_y * (1-\chi_z) + \\
&V_{001} * (1-\chi_x) * (1-\chi_y) * \chi_z + \\
&V_{101} * \chi_x * (1-\chi_y) * \chi_z + \\
&V_{011} * (1-\chi_x) * \chi_y * \chi_z + \\
&V_{110} * \chi_x * \chi_y * (1-\chi_z) + \\
&V_{111} * \chi_x * \chi_y * \chi_z
\end{aligned}$$

4.2.4 Kritérium

Kritérium registrace je zobrazení $C : \Phi \to \mathbb{R}$ z prostoru parametrů transformace Φ do reálných čísel . Nalezení nejlepší registrace obrazů je pak představováno úlohou optimalizace C.

Vzájemná informace (mutual information)

Vzájemná informace (MI) je v poslední době jedním z nejpoužívanějších kritérií v medicínské registraci. Jedná se o míru statistické závislosti mezi dvěma datasety. Je obzvláště vhodná pro multimodalitní registraci, často se používá pro registraci CT a MR dat. V [12] je k dispozici průzkum literatury o registraci pomocí MI v medicínské oblasti.

Myšlenku vzájemné informace budeme ilustrovat nejprve na Hartleyho a Shannonově entropii. Uvedeme také *normalizovanou vzájemnou informaci* (NMI).

Hartleyho a Shannonova entropie Entropií budeme v tomto textu rozumět míru informace obsažené v jisté zprávě (posloupnosti symbolů z dané abecedy), případně diskrétní náhodné veličině. Hartley definoval pro zprávu o n symbolech, přičemž každý symbol může nabývat s hodnot, entropii jako

$$H = n * \log s \; = \log s^n$$

Vycházel z myšlenky, že množství informace obsažené ve zprávě musí záviset lineárně na počtu znaků n, tedy $H = K * n$ pro nějakou konstantu K, a zároveň pro dvě zprávy s parametry s_1, n_1, s_2, n_2, kde platí $s_1^{n_1} = s_2^{n_2}$, musí být entropie stejná. Hartleyho způsob výpočtu entropie však nereflektuje skutečnost, že různé znaky (a různé zprávy) se vyskytují s odlišnou pravděpodobností. Tuto nevýhodu odstraňuje *Shannonova entropie*, která je pro jevy $e_1, e_2, ..., e_n$ s pravděpodobnostmi $p_1, p_2, ..., p_n$ definována jako

$$H = \sum_i p_i * log\frac{1}{p_i} = -\sum_i p_i * logp_i$$

Protože logaritmus je rostoucí na $(0, \infty)$ a výraz $\log \frac{1}{p_i}$ tedy s rostoucí pravděpodobností klesá, lze definiční vztah interpretovat tak, že množství informace plynoucí z jevu je nepřímo úměrné tomu, jak často se jev vyskytuje. To je přirozené: pokud existuje jev, který se vyskytuje jen velmi zřídka, jeho informační hodnota je mnohem větší, než u jevu, který se objevuje velmi často. Entropie může být také interpretována jako *míra nejistoty*: čím větší číselná hodnota je výsledkem, tím větší je nejistota ohledně obsahu zprávy. Největší hodnoty nabývá tehdy, pokud jsou všechny jevy stejně pravděpodobné, a tedy nejistota je největší.

Celkem tedy můžeme nalézt tři užitečné interpretace entropie: [12]

- Množství informace
- Míra nejistoty
- Rozptyl (disperze) rozdělení pravděpodobností (histogramu)

Shannonovu entropii je samozřejmě možné vypočítat i pro obrazová data. Obrazovou funkci chápeme jako diskrétní náhodnou veličinu, přičemž rozdělení pravděpodobnosti odhadneme normalizací histogramu.

Při registraci obrazu má význam výpočet *sdružené entropie*, což je pro náhodné veličiny A, B

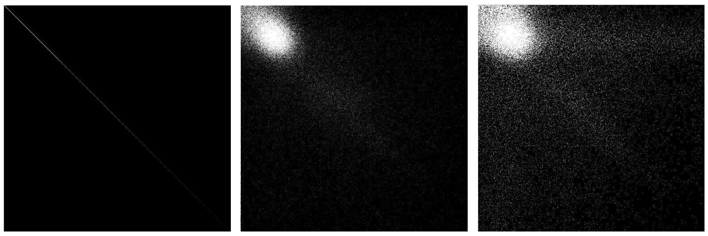
$$H(A, B) = \sum_{a,b} p(a, b) * log\frac{1}{p(a, b)}$$

kde $p(a, b)$ je sdružená pravděpodobnost. Analogicky k předchozímu případu lze určit sdružené pravděpodobnosti ze *sdruženého histogramu* náhodných veličin A, B.

Pomocí sdružené entropie lze dobře charakterizovat disperzi sdruženého histogramu. Čím větší je disperze histogramu, tím větší je i sdružená entropie. Nejlepší registraci obrazů odpovídá nejmenší hodnota sdružené entropie (nejmenší rozptyl v histogramu).

Vzájemná informace (mutual information) Nechť A, B jsou obrazová data a H je Shannonova entropie. Tři nejpoužívanější definice mutual information $I(A, B)$ podle [12] jsou:

$$I(A, B) = H(B) - H(B|A)$$
$$I(A, B) = H(A) + H(B) - H(A, B)$$
$$I(A, B) = \sum_{a,b} p(a, b) * log\frac{p(a, b)}{p(a)p(b)}$$



Obrázek 4.2: Souvislost disperze sdruženého histogramu a registrace obrazu. Vlevo vidíme sdružený histogram odpovídající obrazům s úplnou registrací, uprostřed a vpravo histogramy pro misregistrované obrazy (histogramy pořízeny při vývoji programu ctseg). Disperze sdruženého histogramu je dobrým kritériem registrace obrazů.

Smysl vzájemné informace je nejlépe patrný z první uvedené definice: vzájemná informace obrazů A, B je rovna množství informace obsažené v B sníženému o množství informace v B, pokud A již známe. Tedy čím více informací o obrazu B obsahuje obraz A, tím je množství vzájemně obsažené informace větší. Zmiňovaná "vzájemnost" plyne ze symetrie

$$I(A, B) = I(B, A)$$

Z druhé uvedené definice je zřejmé, že vzájemnou informaci maximalizujeme právě tehdy, když minimalizujeme sdruženou entropii. Maximalizace vzájemné informace tedy odpovídá nalezení nejlepší registrace obrazů A,B.

Normalizovaná vzájemná informace (NMI) Vzájemná informace MI ve výše uvedených definicích vykazuje nedostatky, pokud ji používáme jako registrační kritérium. Její hodnota totiž závisí na velikosti regionu, ve kterém se překrývají plovoucí a referenční data, čímž může docházet k preferenci transformací, které neodpovídají optimální registraci obrazů. Proto je v [15] definována *normalizovaná vzájemná informace* jako

$$Y(A, B) = \frac{H(A) + H(B)}{H(A, B)}$$

4.2.5 Optimalizace

Powellova metoda

Powellova metoda [13] pro optimalizaci funkcí více proměnných patří do kategorie tzv. *direction set* metod, tedy algoritmů, které pohyb v prostoru parametrů optimalizované funkce řídí s pomocí množiny směrových vektorů.

Předpokládejme, že máme k dispozici funkci, která provádí minimalizaci zadané funkce v jednom rozměru. Jejím vstupem je počáteční bod a směrový vektor, výstupem je bod, v němž funkce nalezla minimum. Ve spojení s Powellovým algoritmem se jako tato minimalizační funkce často používá Brentův algoritmus (Brent's line minimization, [13]).

Dále mějme seznam směrových vektorů a počáteční bod optimalizace. Pokud optimalizujeme funkci přes n-rozměrný prostor parametrů, pak těchto směrových vektorů musí být n. Základní myšlenkou direction set metody je nalézt pomocí jednorozměrné optimalizační metody optimum nejprve ve směru prvního směrového vektoru, vycházeje z počátečního bodu, a odtamtud hledat minimum ve směru dalšího směrového vektoru. Cyklem přes všech n směrů se provede jeden základní krok algoritmu.

Základním problémem algoritmu je nalezení správné množiny směrových vektorů. Je zřejmé, že jednoduché řešení např. pomocí množiny bázových vektorů prostoru parametrů je pro některé druhy funkcí nevhodné a může vést k velmi pomalé konvergenci. Příkladem, který uvádí [13], je funkce ve tvaru dlouhého úzkého údolí, jehož hlavní osa je v nějakém (netriviálním) úhlu natočená k systému bázových vektorů. Množina vektorů navíc musí mít speciální vlastnost, že již nalezené optimum není poškozeno hledáním optima v dalších směrech (tzv. *konjugované směry*).

Powell tento problém řeší aktualizací množiny směrových vektorů mezi základními kroky algoritmu. První směrový vektor zahodí a ke zbývajícím vektorům připojí (na konec) nový směrový vektor $P_n - P_0$, kde P_0 je počáteční bod, a P_n odhadnutý optimalizovaný bod. Tento směrový vektor můžeme interpretovat jako "průměrný" směr pohybu po vyzkoušení všech směrových vektorů.

Bohužel, tento přístup vede k vytváření množin lineárně závislých vektorů, což může zapříčinit nekorektní výsledky optimalizace (algoritmus se omezí jen na podprostor prostoru parametrů). Existuje několik způsobů, jak tyto potíže odstranit (blíže [13]). Jedním z nich je použít heuristiky: pro zahození nevybereme první směrový vektor z množiny, nýbrž ten, v jehož směru došlo k největšímu poklesu hodnoty optimalizované funkce. Je totiž pravděpodobné, že tento směr je nejvíce zastoupen v nově přidaném vektoru, a tudíž by mohl v další množině vektorů způsobit lineární závislost.

Powellův algoritmus v této implementaci vypadá následovně:

```
/* Inicializace směrových vektorů (n[i] jsou bázové vektory) */

for i := 1 to n do
 u[i] := n[i]

/* Základní krok */
while(funkce klesá) do
begin

 P[0] := počáteční bod

 /* Minimalizace podle směrů v množině */
 for i := 1 to n do
 begin

     minimalizuj lineárně z bodu P[i-1] ve směru u[i]
     výsledný bod ulož do P[i]

 end

 df := největší pokles hodnoty optimalizované funkce f při lineární optimalizaci

 /* Aktualizace množiny směrů */
 fE := f(2*P[n] - P[0])
 f0 := f(P[0])
 fN := f(P[n])

 if( fE >= f0 )
 {
    /* Zachovej množinu směrů : P[n] - P[0] není dobrý kandidát */

 }
 else
 if( 2*(f0 - 2*fN + fE) * sqr(f0 - fN - df) >= sqr(f0 - fE)*df)
 {

    /* Zachovej množinu směrů (podrobnosti viz literatura) */

 }
 else
 {

    /* Nahraď vektor s největším poklesem vektorem P[n] - P[0] */

 }

end
```

Kapitola 5

Maskování

Vzhledem k diskrétní povaze tomografických dat a skutečnosti, že algoritmy pro odhad rigidní transformace jsou pouze iterativní a nepřesné, dochází při odečtení registrovaných dat ke vzniku artefaktů ve výsledném obraze. Tyto artefakty mohou v určitých případech zhoršit výsledky vizualizace. Například při zobrazení metodou MIP (maximum intensity projection, viz 2.3.2) mohou být některé části obrazu díky drobné misregistraci zakryty artefakty s vysokou denzitou, které jsou pozůstatkem nepřesného odečtení obrazu v místech rozhraní kostní tkáně s méně denzním okolím. Jedním ze způsobů, jak tyto artefakty odstranit, je algoritmus navržený v [6].

5.1 Vstup a výstup algoritmu

Vstupem maskovacího algoritmu jsou:

- postkontrastní data (CTA)
- prekontrastní data (CT)
- odečtená data (S)

Předpokládejme, že jako referenční série byla zvolena CTA data a jako plovoucí CT data. Registrační algoritmus nalezl transformaci CT dat do prostoru CTA dat (budeme značit TCT), která by měla být nejlepším prostorovým ztotožněním těchto sérií. Výstupem algoritmu je maska M určená pro aplikaci na dataset S. Masku M dostaneme aplikací logického OR na *masku kostí* a *masku vzduchu*.

5.2 Konstrukce masky

Konstrukce masky kostí i masky vzduchu má dvě fáze:

- Vytvoření počáteční masky
- Expanze masky

Vytvoření počáteční masky je provedeno v prostoru TCT dat (dat pořízených před aplikací kontrastního média, ale transformovaných do prostoru CTA dat), kde jsou cévy zachyceny s nízkou denzitou. Pomocí prahování je v TCT možné efektivně segmentovat kostní tkáň i vzduch, čímž vytvoříme počáteční masku. Následná expanze masky je řízena srovnáním s CTA daty: u voxelů na hranici masky zkoumáme voxely na korespondující pozici v CTA datech a pokud je učiněno rozhodnutí, že voxel CTA dat má být odstraněn maskou, je maska rozšířena na jeho souřadnice. Tento proces se iterativně opakuje, dokud existují takové voxely. Konkrétní podmínky pro rozhodnutí se liší u masky kostí a masky vzduchu.

5.3 Maska kostí

Počáteční masku kostí vytvoříme jako

$$\begin{aligned} M_b(x,y,z) &= 1,\ \text{pokud } TCT(x,y,z) > B_{th} \\ &= 0,\ \text{jinak} \end{aligned}$$

kde B_{th} je uživatelsky stanovený práh pro segmentaci kostní tkáně. Expanze probíhá podle pravidla

$$\begin{aligned} M_b(x+i,y+j,z+k) = 1,\ \text{pokud } & M_b(x,y,z) = 1, \\ & M_b(x+i,y+j,z+k) = 0, \\ & CTA(x,y,z)- \\ & -CTA(x+i,y+j,z+k) > B_{gr} \end{aligned}$$

pro $i,j,k = -1,0,1$ vyjma $i = j = k = 0$.

kde B_{gr} je uživatelsky nastavený *práh gradientu*. Do masky tedy přidáme jen takový voxel U, který sousedí s existujícím voxelem masky V a pokles v denzitě mezi U a V je dostatečně vysoký. Takovým poklesem jsou charakterizovány voxely na rozhraní kostní tkáně a vzduchu (resp. jiných málo denzních tkání), správné nastavení gradientu však brání, aby se maska rozšířila z kostní tkáně do cévy (charakteristický je malý pokles v denzitě).

5.4 Maska vzduchu

Počáteční masku vzduchu vytvoříme jako

$$\begin{aligned} M_a(x,y,z) &= 1, \text{ pokud } TCT(x,y,z) < A_{th} \\ &= 0, \text{ jinak} \end{aligned}$$

kde A_{th} je uživatelsky stanovený práh pro segmentaci vzduchu. Expanze probíhá podle pravidla

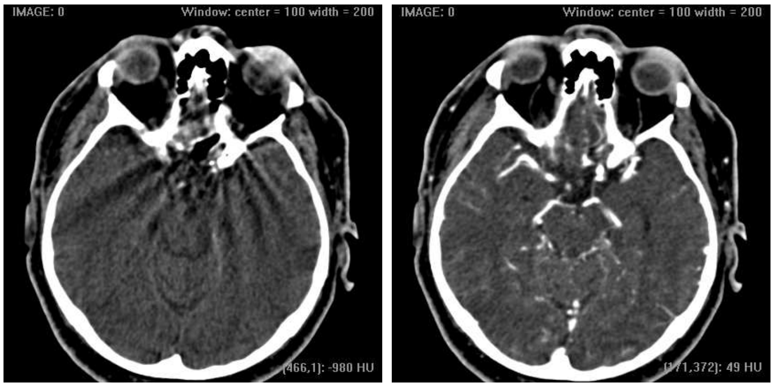
$$\begin{aligned} M_a(x+i,y+j,z+k) = 1, \text{ pokud } & M_a(x,y,z) = 1, \\ & M_a(x+i,y+j,z+k) = 0, \\ & CTA(x+i,y+j,z+k) < A_{th} \end{aligned}$$

pro $i,j,k = -1,0,1$ vyjma $i = j = k = 0$.

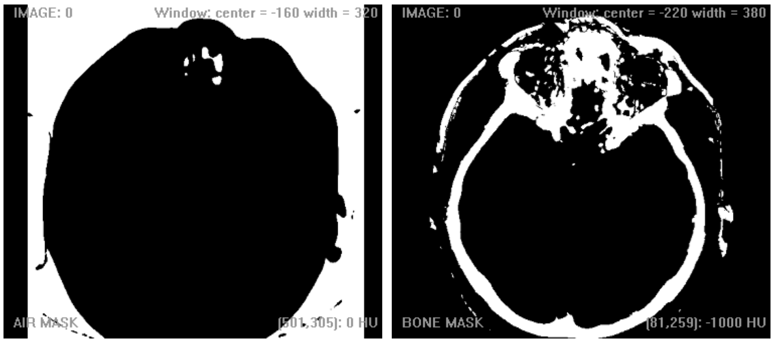
Při expanzi masky vzduchu není nutné zavádět práh pro gradient obrazové funkce, protože nehrozí expanze masky do nežádoucích oblastí.

5.5 Příklad použití

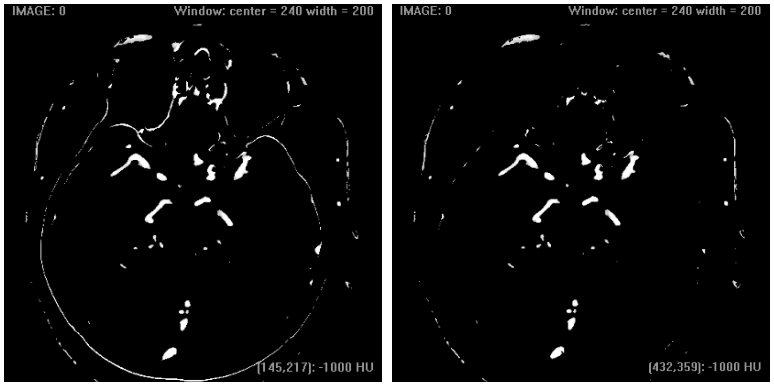
Uvažujme dvojici datasetů s prekontrastními a postkontrastními daty pořízenými při angiografickém vyšetření mozku. Předpokládejme, že vzhledem k podmínkám snímání jsou datasety dobře předregistrované a aplikujme prahování pro odstranění denzního povrchu pokožky, subtrakci a následné maskování.



Obrázek 5.1: Vstupní prekontrastní a postkontrastní série



Obrázek 5.2: Vygenerovaná maska vzduchu (vlevo) a maska kostí (vpravo)



Obrázek 5.3: Vlevo subtrahovaná data bez aplikace maskování, vpravo subtrahovaná data po aplikaci maskování

Kapitola 6

Implementace

6.1 Platforma

Podle požadavků kladených na možnosti využití aplikace *ctseg* byla jako cílová platforma zvolena pracovní stanice typu PC s procesorovou architekturou IA-32, operačním systémem Windows 2000 nebo vyšším, a operační pamětí alespoň 1 GB. Program neklade na software ani hardware cílové platformy žádné další požadavky. Předpokládá korektní vstupní data.

6.2 Architektura programu

Implementace je rozdělena do tří základních vrstev, čímž je zajištěna modularita a pružnost při úpravách jednotlivých komponent, resp. při případném přenosu na jinou hardwarovou či softwarovou platformu. Nejnižší vrstva zajišťuje vstupně-výstupní operace, střední vrstva obsahuje algoritmické jádro a nejvyšší vrstva implementuje uživatelské rozhraní.

Vrstva	**Modul**	**Typ**
Vstup a výstup	DICOMIO	staticky linkovaná knihovna (LIB)
Algoritmické jádro	ENGINE	dynamicky linkovaná knihovna (DLL)
Grafické uživatelské rozhraní	GUI	Win32 MFC aplikace

Implementace je provedena v prostředí Microsoft Visual C++ 6.0. Jednotlivé moduly jsou vedeny jako projekty sdružené ve společném workspace.

6.3 Modul DICOMIO

Knihovna DICOMIO implementuje vstup a výstup pro snímky a série standardu DICOM. Pro přístup k DICOM souborům využívá moduly *dcmdata* a *ofstd* z open-source projektu DICOM Toolkit (DCMTK). DICOMIO slouží jako wrapper nad strukturami DCMTK, navíc přidává pomocné funkce pro vyšší vrstvy (čtení a změna důležitých atributů DICOM hlavičky) a implementuje načítání sérií z adresářů.

Omezení V dodané implementaci jsou pro zjednodušení omezeny některé parametry vstupních DICOM souborů. Předpokládáme uložení pixelových dat v 16bitových slovech s 12bity obrazové informace (rescale intercept -1024) a ve fotometrické interpretaci MONOCHROME2. Zobrazení je optimalizováno pro rozměry řezů 512*512 pixelů. Jedná se o parametry smluvené s radiodiagnostickým pracovištěm FN Na Bulovce. Jejich změna nebo úprava systému pro větší množinu akceptovaných dat je snadná.

6.4 Modul ENGINE

6.4.1 Formát obrazových dat

Moduly algoritmického jádra pracují s 3D obrazovými daty uloženými ve třídě *ImageData*. Tato třída je šablonována typem voxelu, a umožňuje tak efektivně ukládat např. normalizovaná obrazová data nebo data z různých modalit. ImageData je (jediný) interface mezi algoritmy a knihovnou DICOMIO.

6.4.2 Schéma implementace algoritmů

Interface pro algoritmický modul je popsán abstraktní třídou *Algorithm*. Ta předepisuje každému algoritmu následující členské funkce:

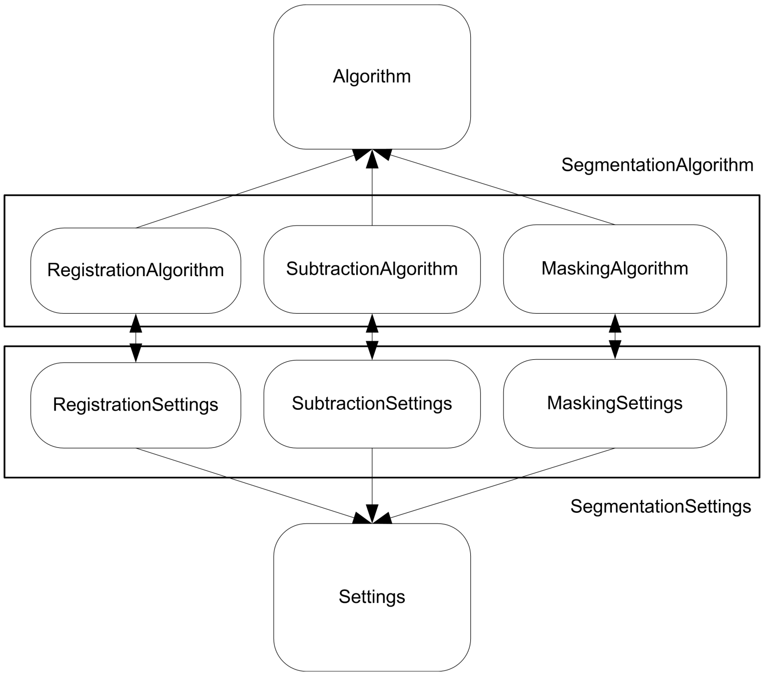
Název	**Parametr**	**Návratová hodnota**	**Popis**
init	Settings*	bool	Inicializuje datové struktury algoritmu.
run	void	bool	Spustí algoritmus s nastaveními uvedenými při volání init.

Třída *Settings* deklaruje vstupně-výstupní interface pro algoritmy. Předepisuje implementaci serializačních funkcí (*serialize* a *deserialize*), u nichž se předpokládá, že je využije modul uživatelského rozhraní pro snadné ukládání nastavení. Pokud třída Settings obsahuje členské objekty typu Settings, při serializaci volá jejich serializační funkce.

Ve většině případů jsou paměťové oblasti pro vstup a výstup algoritmů modulu ENGINE (instance potomků třídy Settings a další typy) alokovány v rutině, která spouští segmentační proces.

6.4.3 Segmentace odčítáním

Schéma segmentace odčítáním a sémantika modulů jsou uvedeny v 3.2. Moduly pro registraci (*RegistrationAlgorithm*), subtrakci (*SubtractionAlgorithm*) a maskování (*MaskingAlgorithm*) jsou implementovány ve shodě se schématem z předchozího odstavce jako potomci abstraktní třídy Algorithm s nastaveními v potomcích abstraktní třídy Settings.

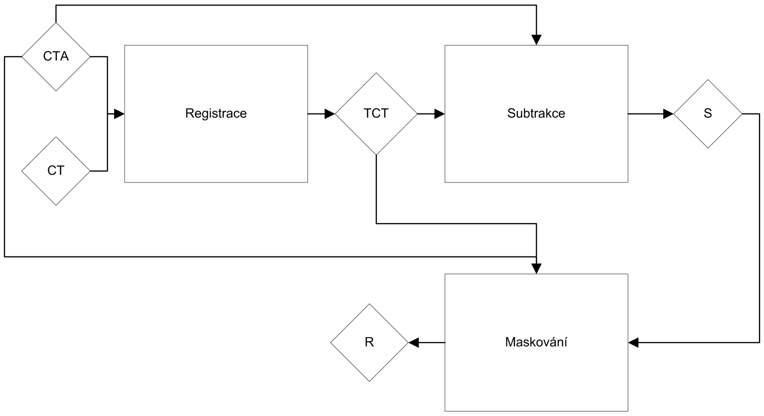


Obrázek 6.1: Schéma implementace segmentace subtrakcí

Třída *SegmentationSettings* obsahuje ukazatele na řídící datové struktury těchto modulů. Kterýkoliv modul je možné vyřadit z činnosti nastavením příslušného ukazatele na nulovou hodnotu. Na obrázku 6.2 je schéma z kapitoly 3.2 rozšířené o znázornění toku dat mezi moduly.

Registrace Registrační algoritmus je rozdělen do několika vlastních submodulů:

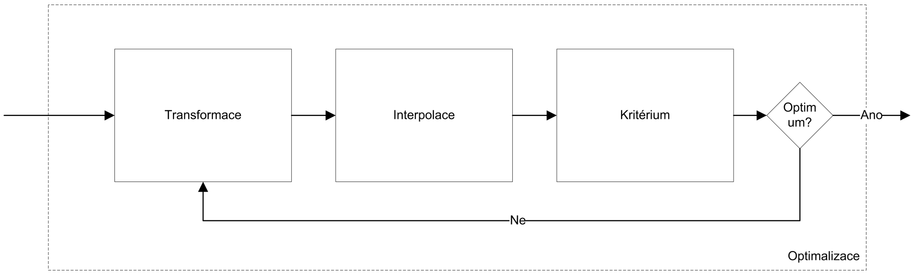
- optimalizace (Optimization)



Obrázek 6.2: Schéma implementace subtrakčního segmentačního algoritmu s vyznačením toku dat mezi moduly

- transformace (Transformation)
- interpolace (Interpolation)
- kritérium (Criterion)

Pro každý submodul je definován interface pomocí abstraktních tříd uvedených v závorkách.



Obrázek 6.3: Schéma implementace registračního algoritmu

Vstupem registračního algoritmu je:

- referenční série
- plovoucí série
- parametry pro thresholding sérií
- typy algoritmů použitých v submodulech

- vstupně-výstupní nastavení submodulů
- rozvrh hierarchického zpracování

Výstupem je:

- transformovaná plovoucí série
- výstupní stav submodulů

Vstupní i výstupní série musí být alokovány již při spuštění algoritmu. Při inicializaci je provedeno prahování a normalizace denzit prekontrastní a postkontrastní série na rozsah histogramu, což umožní efektivnější výpočty při opakovaném zpracování. Pro normalizovaná data je alokována samostatná paměťová oblast, původní série nejsou poškozeny. V průběhu registrace (při běhu optimalizačního algoritmu) je plovoucí série transformována on-the-fly a transformovaný obraz se jako celek nikam neukládá. Po skončení optimalizace je plovoucí série transformována a uložena do poskytnuté datové struktury.

6.5 Modul GUI

GUI (graphical user interface) implementuje front-end projektu jako grafickou aplikaci platformy Win32 s využitím knihovny MFC (Microsoft Foundation Classes). Implicitně linkuje DLL knihovnu s algoritmickým jádrem projektu, staticky je připojen modul DICOMIO.

Základní úkoly, které plní modul GUI, jsou:

- vizualizace vstupních i výstupních sérií
- organizace sérií ve studiích
- načtení dat, příprava datových struktur, spuštění algoritmického jádra a uložení výsledků

6.6 Prostorová složitost

Vzhledem k charakteru úlohy a obrazových dat, se kterými algoritmy pracují, vykazuje implementace netriviální prostorovou složitost. V tomto odstavci uvedeme odhad prostorové složitosti v případě, kdy jsou při výpočtu aktivovány všechny moduly segmentačního algoritmu a výpočet je spouštěn z grafického uživatelského rozhraní. Nechť jsou vstupem dvě DICOM série se dvěma sty snímků, velikost jednoho snímku je 512 * 512 pixelů. Denzita každého voxelu je uložena ve 12 bitech, což znamená dvoubytový záznam pro každý voxel. Tento vstup můžeme považovat za typický pro námi řešené úlohy na spolupracujícím radiodiagnostickém pracovišti. Obě vstupní série jsou načteny grafickým uživatelským rozhraním do paměti při otevření studie. Ke každému snímku je navíc připojena pomocná paměť pro rychlou vizualizaci (viz 2.3.1), její velikost je 512*512*1 bytů. Vstupní data tedy zabírají 512*512*(2 + 1)*200*2 = 314 572 800 B = 300 MB. Charakter registrace a použité metriky vyžaduje, aby celá tato data byla v paměti (alespoň v průběhu registrace). GUI dále alokuje paměťové bloky pro výstupní data: transformovaná CT data (TCT), výsledek subtrakce (S), masku kostí (AM) a vzduchu (BM). Maskování se aplikuje na paměťový blok S, nevyžaduje tedy nový paměťový blok. Velikost TCT dat je 512*512*2*200 bytů = 512 kB*200 = 100 MB, stejně tak pro S. Masky AM a BM jsou implementovány pomocí kontejneru vector<bool>, předpokládejme jeho specializaci pro tento datový typ, která ukládá jednotlivé položky jako bity. Tedy 512*512*200/8 = cca 6 MB. Registrační algoritmus si dále vytváří normalizované varianty CT a CTA dat, každá z nich zabírá 512*512*1*200 = 50 MB. Celkem musí být tedy alokováno 2*100 + 2*6 + 2*50 = 312 MB dat, což je celkem 612 MB. Připomeňme, že jsme neuvažovali některé pomocné datové struktury, jako např. struktura pointerů ve třídě ImageData pro rychlý přístup k voxelu, sdružený histogram pro kritérium NMI a některé další zanedbatelné položky.

Kapitola 7

Uživatelská příručka

7.1 Vstupní a výstupní data

Program jako svůj vstup očekává prekontrastní a postkontrastní DICOM sérii, jako výstup poskytuje několik DICOM sérií a konfiguračních souborů v závislosti na konkrétním nastavení. Každá DICOM série musí být uložena v samostatném adresáři. V tomto adresáři se nachází pouze soubory ve formátu DICOM, které představují jednotlivé snímky série. Názvy souborů mohou být libovolné, seřazení snímků se provede podle DICOM hlavičky.

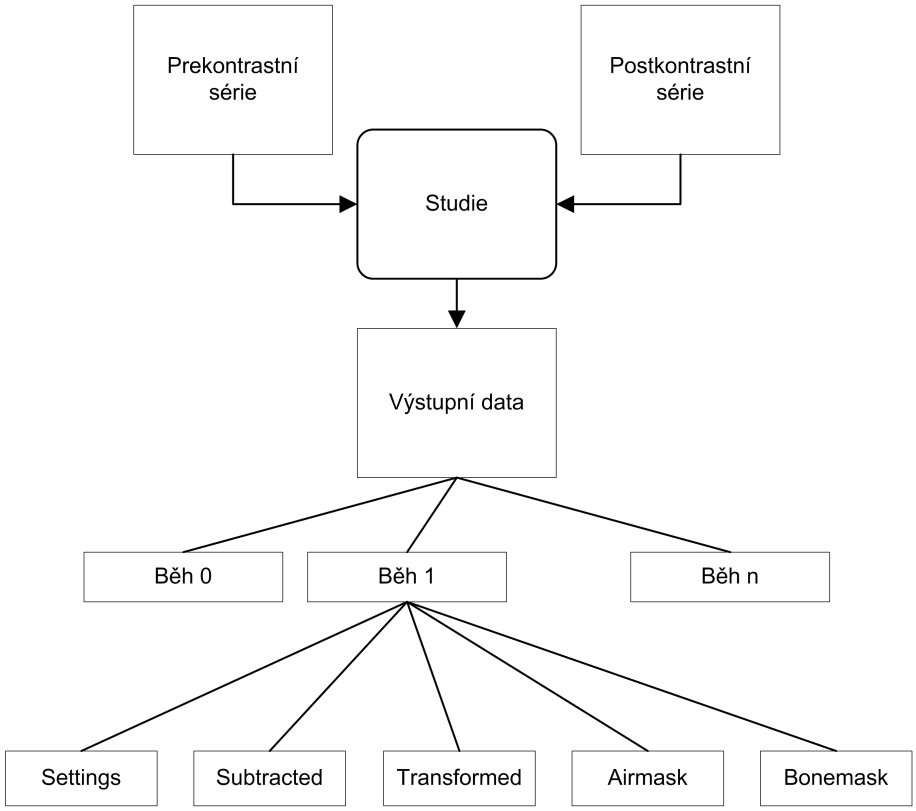
Identifikace výstupních dat Aby bylo možné používat série vytvořené programem *ctseg* v sytémech PACS (Picture Archiving and Communication Systems), sériím je přiřazeno nově vygenerované *Series Instance UID*, nové pořadové číslo série ve studii (začíná od hodnoty 777) a název popisující obsah.

7.2 Organizace dat

Práce s programem je organizována do *studií*. V rámci každé studie je definováno:

- umístění vstupních prekontrastních dat
- umístění vstupních postkontrastních dat
- umístění výstupních dat

Každé umístění je reprezentováno adresářem na disku. Výstupní data jsou rozdělena do *běhů*, které reprezentují jednotlivá spuštění segmentace. Tato organizace dat umožní uchovat a později analyzovat informace o všech spuštěních algoritmu s různými nastaveními.



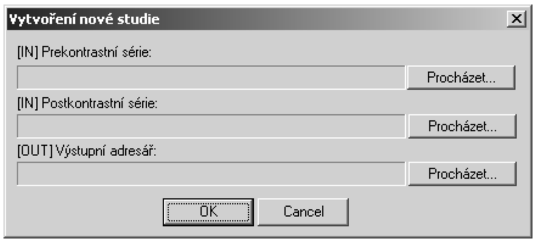
Obrázek 7.1: Schéma studie

Data	**Adresář**	**Obsah**
Prekontrastní série	libovolný	
Postkontrastní série	libovolný	
Výstupní data	libovolný (označme "out")	
Běh n	out/run.n	Výsledky n-tého běhu. Běhy jsou číslovány od nuly.
Subtracted	out/subtracted	DICOM série s výslednými odečtenými daty
Transformed	out/transformed	Transformovaná CT série
Air mask	out/airmask	DICOM série s maskou vzduchu
Bone mask	out/bonemask	DICOM série s maskou kostí
Settings	out/params	Konfigurační soubory s nastaveními příslušného běhu

DICOM série v jednotlivých adresářích mohou být využity pro další zpracování samostatně. Pro analýzu výsledků běhu a porovnání sérií mezi sebou je vhodné využít příslušné funkce programu ctseg, viz kapitola 7.8.

7.3 Vytvoření nové studie

Před vytvořením studie je nutné mít připravené vstupní DICOM série. Adresář pro výstupní data je možné vytvořit přímo v dialogu Procházet. Po stisknutí tlačítka OK vás program ctseg vyzve k zadání názvu souboru, který bude řídicím souborem studie. V tomto souboru jsou uloženy absolutní cesty k zadaným adresářům a další informace. Program žádným způsobem nedefinuje vzájemné umístění adresářů se vstupem a výstupem, což umožňuje lepší adaptabilitu při zapojení do existujícího řetězce zpracování. Je však třeba dbát na to, že uložené cesty jsou absolutní.



Obrázek 7.2: Dialog pro vytvoření nové studie

7.4 Otevření existující studie

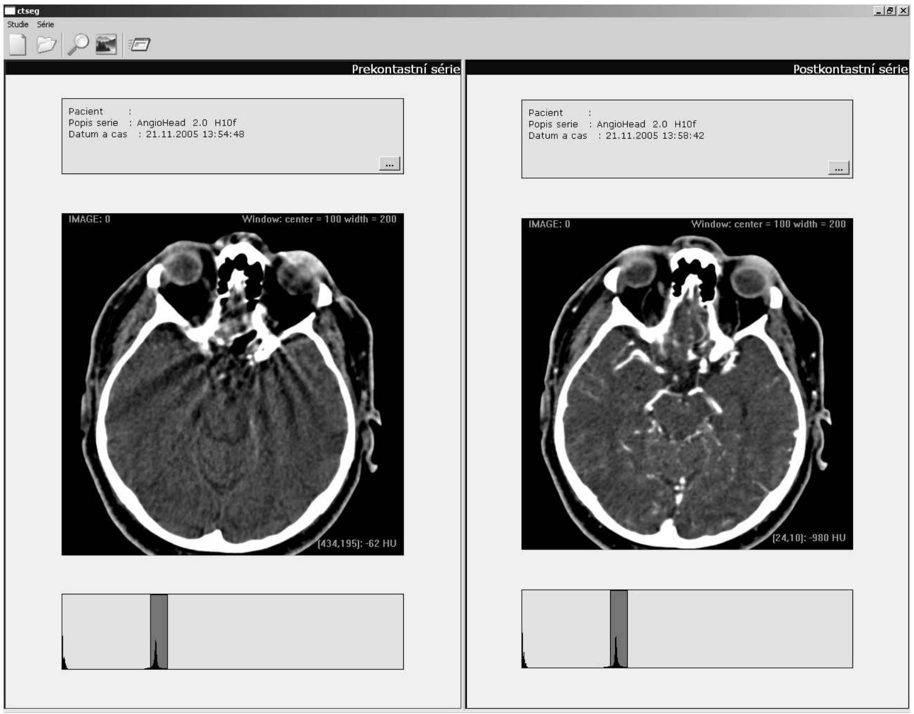
Provede se zvolením řídícího souboru studie v dialogu, který se zobrazí po volbě Studie - Otevřít studii.

7.5 Uložení studie

Po každém spuštění segmentace je uživateli nabídnuta možnost uložení výstupních dat do nového běhu. Pokud se na základě průběhu výpočtu rozhodnete neukládat výstup, nový běh nebude vytvořen a studie nebude nijak změněna. V opačném případě je automaticky provedeno uložení studie a zapsání dat.

7.6 Hlavní okno aplikace

Pokud je otevřená studie, v základním zobrazení je k dispozici náhled prekontrastní a postkontrastní série, spolu s histogramem a se základními informacemi uloženými v DICOM hlavičce. Před použitím příkazů z menu Série, které se vztahují k jednotlivým sériím (nikoli celé studii), je nutné se přesvědčit, že některá ze sérií je zvolena (její okno je ohraničeno červenou čarou).



Obrázek 7.3: Hlavní okno aplikace

7.6.1 Zobrazení DICOM série

Elementárním způsobem zobrazení série je procházení jednotlivých snímků ve dvourozměrném zobrazení. Okno programu ctseg se zobrazením DICOM série poskytuje tyto základní informace (zobrazeny zeleným písmem):

- číslo aktuálně zobrazeného snímku
- střed a šířka okna pro prahování
- denzita ve středu kurzoru (pokud se myš nachází nad oknem)
- popis série (volitelně)

Zobrazení je možné ovládat myší. Pokud je stisknuto levé tlačítko, probíhá posun zobrazených snímků. Pohybem myši směrem k hornímu okraji okna se zobrazení posunuje směrem ke snímkům s vyšším číslem, a analogicky při pohybu směrem dolů ke snímkům s nižším číslem. Při stisku pravého tlačítka se mění parametry prahování. Pohyb v horizontálním směru mění šířku okna při zachování jeho středu, pohyb ve vertikálním směru posunuje střed okna. V průběhu manipulace s oknem je prahování aplikováno jen na aktuálně zobrazený snímek, po uvolnění pravého tlačítka je dopočteno prahování pro celou sérii. Pokud je přiřazeno okno s histogramem, lze na něm sledovat aktuální umístění limitů pro thresholding. Cílem prahování je především usnadnění analýzy obrazu člověkem. Nastavené hodnoty se však přenášejí také do dialogu pro spuštění segmentace, kde slouží jako výchozí nastavení pro prahování použité jednotlivými moduly.

7.6.2 Histogram

Histogram umožňuje zobrazit aktuální nastavení limitů prahování v kontextu výskytu denzit ve snímku. Na vodorovné ose jsou vyneseny jednotlivé hodnoty denzit, které se mohou ve snímku vyskytovat, na svislé ose jejich relativní četnosti.

7.6.3 Samostatné otevření série

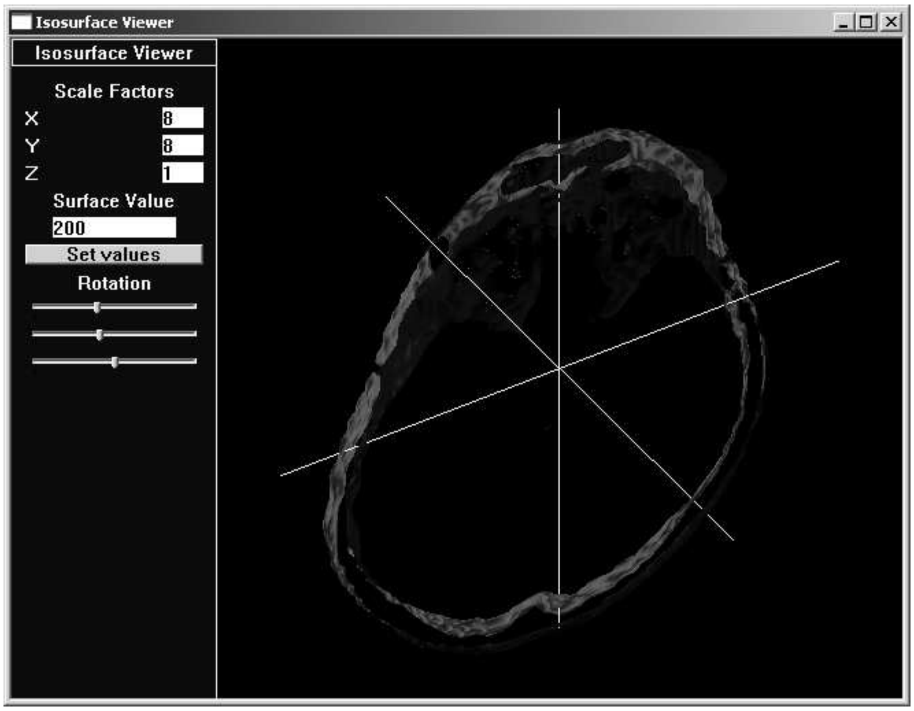
Při běžném způsobu práce se studií jsou vstupní série otevřeny automaticky. Program lze však použít i jako jednoduchý prohlížeč sérií bez zařazení do studie, pokud zvolíte v menu Série položku Otevřít z adresáře. Jestliže je otevřena studie, nelze některou z jejích vstupních sérií nahradit tímto způsobem. Studii je nutné nejprve zavřít, případně vytvořit novou studii.

7.6.4 Trojrozměrná vizualizace série

Program ctseg nabízí vizualizaci izoploch v tomografických datech pomocí algoritmu Marching cubes. Zvolte jednu ze vstupních sérií a zadejte v menu Série položku 3D vizualizace. Stiskem levého tlačítka a pohybem myši se nastavuje orientace modelu, stiskem pravého tlačítka se modifikuje denzita izoplochy. Po uvolnění pravého tlačítka program provede přepočítání celého modelu, což může trvat až několik minut.

7.7 Segmentace cév

Otevřete studii a zvolte položku Spustit segmentaci v menu Studie. Program nabídne dialog, ve kterém se nastavují parametry jednotlivých modulů segmentačního algoritmu. Každý modul kromě subtrakce lze vyřadit z činnosti. Pokud je například požadavek na rychlé vyhodnocení dat, které po zběžném prozkoumání vykazují jen malou misregistraci, je možné vyřadit z činnosti registrační modul, a nechat zapnuté maskování jako korekci artefaktů vzniklých po odečtení. Po



Obrázek 7.4: Trojrozměrná vizualizace série

nastavení všech parametrů se stiskem tlačítka Spustit v paměti vytvoří dočasné výstupní série, nad nimiž běží algoritmické jádro. V tomto kroku je alokováno velké množství paměti, odhad je uveden v 6.6. Průběh výpočtu je možné sledovat prostřednictvím textového výstupu, který jádro generuje, a který bude v případě uložení běhu připojen k výsledným sériím. Pokud chcete výpočet ukončit ještě před řádným zastavením, stiskněte Storno a vyčkejte, než algoritmus ohlásí, že byl korektně ukončen. Poté můžete dialog zavřít a studie nebude nijak změněna. Jestliže algoritmus skončí korektně, můžete výsledná data, která jsou zatím pouze v paměti, buď opět stornovat a zrušit, nebo stiskem Uložit vytvořit nový běh.

7.8 Analýza běhu

Jednotlivé moduly algoritmického jádra generují diagnostické výstupy, které jsou důležité pro analýzu mezivýsledků, které produkují, a pro zkoumání jejich vlivu na výsledná data. Pro tyto účely nabízí program speciální okno (viz obrázek 7.5), které poskytuje náhled na celkem sedm sérií:

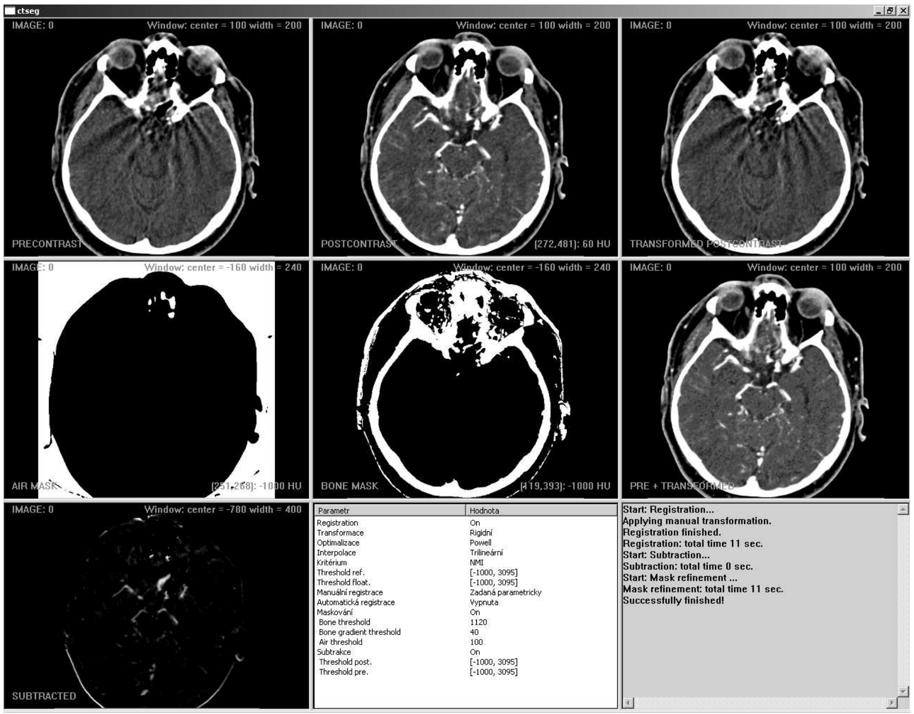
- prekontrastní série
- poskontrastní série

- transformovaná prekontrastní série
- maska vzduchu
- maska kostí
- transformovaná prekontrastní série a postkontrastní série v kombinaci
- odečtená série

Pokud jsou některé moduly v analyzovaném běhu vypnuty (např. registrace), pak jsou jim příslušející série nedostupné (např transformovaná série).
K dispozici jsou rovněž dvě okna s konfigurací běhu:

- prohlížeč parametrů
- textový výstup algoritmu

Analýzu běhu je možné vyvolat z menu Studie v hlavním okně programu.



Obrázek 7.5: Analýza běhu algoritmu

Kapitola 8

Závěr

8.1 Shrnutí dosažených výsledků

V této práci jsme chtěli nalézt a implementovat řešení, které by zlepšilo možnosti vizualizace cév neurokrania. Po průzkumu existujících segmentačních metod jsme se rozhodli co nejvíce využít specifika aplikační domény, a to rigidní charakter transformace vykázané mezi lebkou a vaskulárním systémem v prekontrastních a postkontrastních datech. Rozhodli jsme se nasadit postupy tradičně používané v digitální subtrakční angiografii (DSA) v CT angiografii podle [6]. Registraci dat řešíme pomocí automatizovaných metod založených pouze na hodnotách voxelů a moderním kritériu registrace NMI. Vyvinutý registrační engine je tak po drobných úpravách možné dále používat pro výzkum multimodalitní registrace. Součástí výsledků práce je grafická aplikace pro platformu Win32, která umožňuje bez dalších mezikroků aplikovat segmentační algoritmy na data získaná z CT přístroje a tato data dále používat a upravovat v profesionálním software.

8.2 Praktické použití

Segmentační algoritmy jsme v průběhu vývoje testovali na několika datasetech, které poskytlo Centrum pro výpočetní tomografii Fakultní nemocnice Na Bulovce. Uvádíme dva příklady dosažených výsledků, přičemž první příklad ilustruje situaci, kdy segmentační algoritmy poskytly přímo výsledky použitelné pro diagnostiku cév. Druhý příklad vyžaduje další úpravy.

8.2.1 Příklad 1

Parametry použitých dat jsou následující:

Rozměr řezu	512 * 512 pixelů
Počet řezů	40 (pre) resp. 34 (post)
Rozměry voxelu	0.39 * 0.39 * 2 mm

Program *ctseg* jsme spustili s těmito nastaveními:

Registrační modul	zapnutý
Transformace	rigidní
Optimalizace	Powellova metoda
Interpolace	trilineární
Kritérium	NMI
Prahování	vypnuto
Manuální registrace	vypnuta
Automatická registrace	rotace a translace
Tolerance argumentu kritéria	1
Tolerance hodnoty kritéria	0.1
Maskování	zapnuto
Práh denzity masky kostí	100
Práh gradientu masky kostí	40
Práh denzity masky vzduchu	-900

Výsledky algoritmu:

Translace	(-1.91,3.09,0) voxelů
Rotace	(0,0,0.06) stupňů
Hodnota kritéria v optimu	1.266
Čas výpočtu registrace	307 s

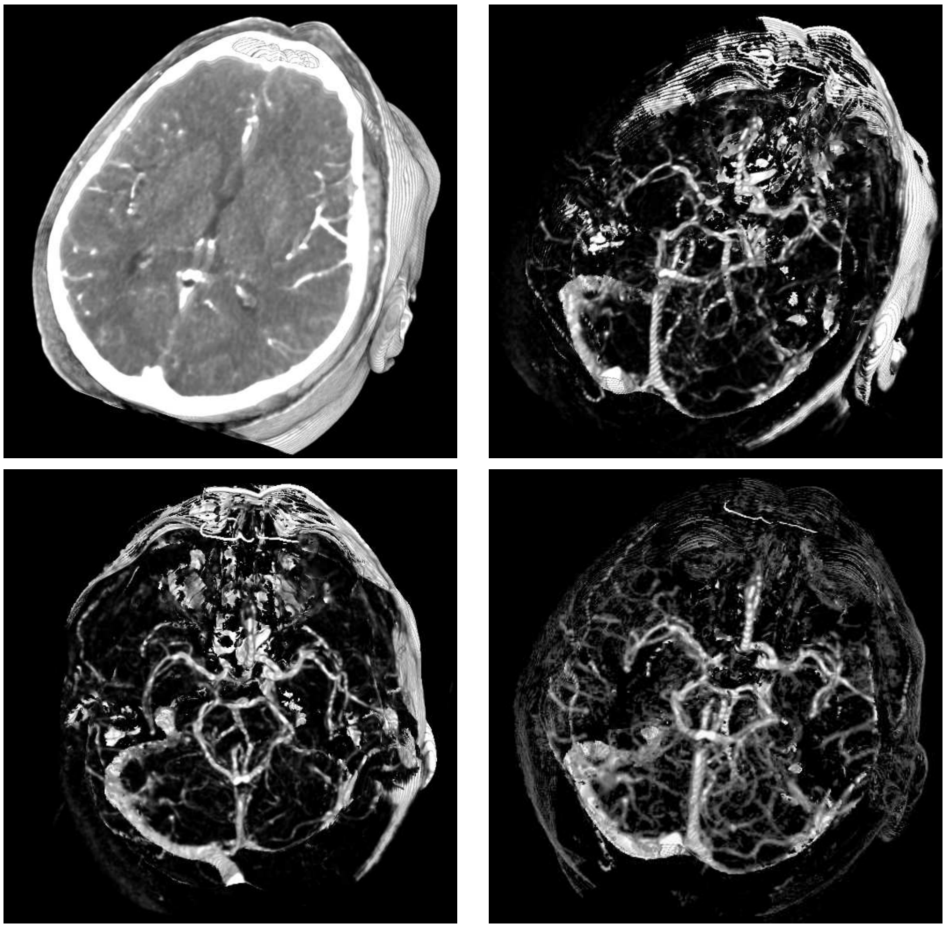
Výsledky jsou znázorněny na obrázku 8.1.

8.2.2 Příklad 2

Parametry použitých dat jsou následující:

Rozměr řezu	512 * 512 pixelů
Počet řezů	40 (pre) resp. 34 (post)
Rozměry voxelu	0.39 * 0.39 * 4 mm

Algoritmus je spuštěn s analogickými nastaveními i výsledky, jako v příkladě 1. Tento dataset však vykazuje malý rozdíl mezi denzitami kontrastního média



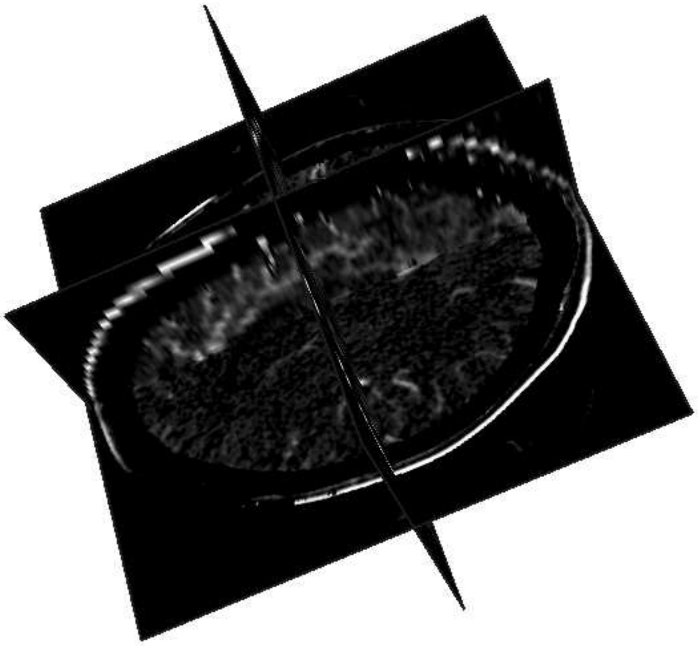
Obrázek 8.1: Příklad 1. Vlevo nahoře vidíme postkontrastní sérii, vpravo nahoře a vlevo dole objemové zobrazení subtrahovaných dat po aplikaci maskování. Vpravo dole je výsledek odečtení, jestliže před spuštěním procesu aplikujeme prahování.

v cévách a denzitou kostní tkáně, resp. povrchu pokožky. Navíc transformaci pokožky nelze aproximovat pouze rotacemi a translacemi. Z těchto důvodu dochází při odečtení ke vzniku výrazných artefaktů, které zhoršují vizualizaci dat při jejich přímém použití.

8.3 Možná vylepšení

V oblasti medicínské segmentace probíhá intenzivní výzkum, který se snaží nalézt univerzální, rychlé a robustní segmentační metody. Použitá skupina algoritmů je pouze jedním z přístupů k problematice. Ukazuje se, že umožňuje s malou interakcí uživatele podstatně zlepšit vizualizaci cév neurokrania. Nevýhodou však je:

- velká časová (a také prostorová) složitost



Obrázek 8.2: Příklad 2. V zobrazení MPR (multiplanar reformation) vidíme, že ve snímku zůstává povrch pokožky, který by při trojrozměrné vizualizaci zakrýval cévní struktury. Dále je možné si všimnout malého rozdílu denzit mezi cévami s kontrastním médiem a okolní tkání.

- v některých případech nedostatečné odstranění artefaktů
- neřeší tvorbu modelu cévní struktury

Nevýhody jsou vzhledem k zamýšlenému použití algoritmů seřazeny podle významu. Časová složitost je problematická zvláště u akutních indikací, kdy vzniká požadavek na téměř okamžité segmentační výsledky. Prostorová složitost v řádech stovek MB je dána charakterem dat a na moderních strojích již nezpůsobuje potíže. Nedostatečné odstranění artefaktů je nevýhoda plynoucí z redukce problému na rigidní transformaci a může snížit efekt segmentace. V neposlední řadě je nutné zmínit, že použité algoritmy segmentují zobrazení cév, ale nekonstruují žádným způsobem jejich popis ani počítačový model. Ten lze využít k další diagnostice nebo pokročilému zobrazení (např. [3]).

Časová složitost vyplývá z obecného řešení pomocí registrace, založené pouze na hodnotách voxelů. Nabízí se řešení:

- Adaptivní maskování [6]
- Celočíselná implementace
- Rychlejší optimalizační metoda (např. ASA)

Literatura

[1] Bühler K., Felkel P., La Cruz A.: *Geometric Methods for Vessel Visualization and Quantification - A Survey*, Tech. rep., VRVis Research Center, Vienna, Austria, 2002.

[2] Čížek J.: *Tomographic Data Processing in Nuclear Medicine*, Diploma Thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2001.

[3] Hahn H.K., Preim B., Selle D., Peitgen H.: *Visualization and Interaction Techniques for the Exploration of Vascular Structures*, Center for Medical Diagnostic Systems and Visualization, Bremen, Germany, 2001.

[4] Hlaváč V., Šonka M.: *Počítačové vidění*, Grada, 1992.

[5] Kirbas C., Quek F.: *A Review of Vessel Extraction Techniques and Algorithms*, Tech. rep., VisLab Wright State University, Dayton, Ohio, Nov 2000.

[6] Kwon S. M., Kim Y. S., Kim T., Ra J. B. : *Digital subtraction CT angiography based on efficient 3D registration and refinement*, Computerized Medical Imaging and Graphics 28 (2004), 391–400.

[7] Lorensen W.E., Cline H. E.: *Marching cubes: a high resolution 3D surface reconstruction algorithm*, Computer Graphics, volume 21, number 4, July 1987.

[8] Maintz J., Viergever M.: *An Overview of Medical Image Registration Methods*, Imaging Science Department, Imaging Center Utrecht, 1996.

[9] Microsoft Corporation: *MSDN Library*, http://msdn.microsoft.com/library/.

[10] NEMA: *DICOM Homepage*, http://medical.nema.org/.

[11] OpenGL ARB: *OpenGL Reference Manual, The Official Reference Document to OpenGL, Version 1.4*, Addison-Wesley Professional, 2004.

[12] Pluim J. P. W, Maintz J.B.A, Viergever M. A.: *Mutual information based registration of medical images: a survey*, IEEE Transactions on Medical Imaging, Vol. XX, No. Y, Month 2003.

[13] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P. : *Numerical Recipes In C: The Art of Scientific Computing*, Cambridge University Press, 1992.

[14] Sharman J.: *The Marching Cubes Algorithm*, http://www.exaflop.org/docs/marchcubes/.

[15] Studholme C., Hill D.L.G., Hawkes D.J.: *An overlap invariant entropy measure of 3D medical image alignment*, Pattern Recognition 32, 71-86, 1999.

[16] Wikipedia contributors: *Computed tomography*. Wikipedia, The Free Encyclopedia. http://en.wikipedia.org, 2006.

[17] Žára J., Beneš B., Sochor J., Felkel P.: *Moderní počítačová grafika*, Computer Press, Brno, 2004.

Příloha A

Obsah disku CD-ROM

K této práci je přiložen CD-ROM, jehož obsah je následující:

A.1 Zdrojové kódy

Implementace algoritmů je k dispozici v programu *ctseg*. Na přiloženém CD jsou jeho zdrojové kódy v adresáři */src/ctseg*, pokyny pro přeložení naleznete v souboru */src/readme.txt*.

A.2 Instalace

Instalátor s přeloženou release verzí programu *ctseg* je k dispozici v adresáři */inst*. Program vyžaduje operační systém Windows 2000 a vyšší, doporučujeme alespoň 1 GB operační paměti.

A.3 Ukázková data

Do adresáře */data* jsme umístili anonymizované DICOM série, se kterými je možné experimentovat v programu ctseg. Přiloženy jsou i dva příklady odečtených sérií, které zmiňujeme v závěrečné kapitole.

A.4 Dokumentace

V adresáři */doc* jsou uloženy zdrojové kódy této práce v systému LaTeX, včetně použitých obrázků.