

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

## **Bakalářská práce**

# **Registrace volumetrických dat pořízených heterogenními technikami**

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2019/2020

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Natálie VÍTOVÁ**  
Osobní číslo: **A17B0583P**  
Studijní program: **B3918 Aplikované vědy a informatika**  
Studijní obor: **Kybernetika a řídicí technika**  
Téma práce: **Registrace volumetrických dat pořízených heterogenními techniky**  
Zadávací katedra: **Katedra kybernetiky**

### Zásady pro vypracování

1. Seznamte se s technologiemi akvizice volumetrických dat zachycujících cévní soustavu v játrech a s obecnými metodami registrace obrazů a třírozměrných povrchových a volumetrických dat.
2. Navrhněte postup pro rigidní registraci volumetrických dat založený na hledání lokálního maxima hustoty v množině kandidátních transformací.
3. Implementujte navržený postup s ohledem na maximální možnou modularitu řešení umožňující nahrazení jednotlivých bloků alternativními algoritmy.
4. Otestujte navržené řešení na dostupných datech, identifikujte problematické vlastnosti navrženého řešení a diskutujte možnosti dalšího vývoje algoritmu.

Rozsah bakalářské práce: **30-40**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná**

#### Seznam doporučené literatury:

- Čihák, R. (2002). Anatomie 2. Praha: Grada Publishing.
- Nekula, J., Heřman, M., Vomáčka, J., & Köcher, M. (2005). Radiologie. Univerzita Palackého .
- Sonka, M., Hlavac, V., & Boyle, R. (2014). Image Processing, Analysis, and Machine Vision Second Edition. Thomson-Engineering.
- Hruďa, L., Dvořák, J., Váša, L.: On evaluating consensus in RANSAC surface registration, Computer Graphics Forum, Vol. 37(5), 2019.

Vedoucí bakalářské práce: **Ing. Miroslav Jiřík, Ph.D.**  
Výzkumný program 1

Datum zadání bakalářské práce: **15. října 2019**  
Termín odevzdání bakalářské práce: **25. května 2020**

*Radová*

---

**Doc. Dr. Ing. Vlasta Radová**  
děkanka



*Psutka*

---

**Prof. Ing. Josef Psutka, CSc.**  
vedoucí katedry

# Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 7. července 2020

Natálie Vítová

# Poděkování

Touto cestou bych chtěla poděkovat vedoucímu své bakalářské práce Ing. Miroslavu Jiříkovi, Ph.D. za skvělé vedení, korekturu práce a užitečné rady. Velký dík patří také Doc. Ing. Liboru Vášovi, Ph.D. za odborné diskuze, cenné rady a za odbornou a psychickou podporu při implementaci algoritmu. Své rodině děkuji za oporu.

## Abstrakt

Práce se zabývá registrací volumetrických dat pořízených mikro CT přístrojem na data pořízená klasickým lékařským CT. V prvních kapitolách je probrána teorie z anatomie jater, radiologie (CT a mikro CT) a registrace obrazů včetně popisu již známých a používaných metod pro registraci dvourozměrných, povrchových a objemových dat. Ve druhé části práce je popis návrhu vlastního algoritmu pro registraci volumetrických dat. Součástí práce je implementovaný algoritmus v programovacím jazyce C# dostupný na platformě GitHub.

## Klíčová slova

Registrace volumetrických dat, CT, Mikro CT, Játra, Rigidní registrace

## **Abstract**

The thesis deals with registration of volumetric data gathered by micro CT and classical medical CT devices. First chapters go through the theory of liver anatomy, radiology (CT and micro CT) and image registration including description of some already known and used methods for registration of two dimensional, surface and three dimensional data. In second part of the thesis there is own algorithm for registration of volumetric data described. The thesis also include implemented algorithm in the programming language C# which is available online on the platform GitHub.

## **Key words**

Registration of volumetric data, CT, Micro CT, Liver, Rigid registration

# Seznam obrázků

1.1	Anatomie jater . . . . .	3
2.1	3D model cévní soustavy jater prasete . . . . .	5
2.2	Princip fungování CT . . . . .	6
2.3	Princip fungování $\mu$ CT . . . . .	8
3.1	Srovnání různých transformací . . . . .	14
4.1	Blokové schéma algoritmu . . . . .	24
4.2	Přehled rozhraní a jejich metod . . . . .	24
4.3	Interpolace . . . . .	28
5.1	Řezy makro daty . . . . .	35
5.2	Řezy mikro daty . . . . .	35
5.3	Srovnání funkčnosti algoritmů pro výpočet feature vektorů .	37
5.4	Vliv náhodné volby směru rotace na výslednou odchylku . .	39
5.5	Srovnání metod <i>Sample()</i> a <i>SampleShifted()</i> . . . . .	41
5.6	Experimenty s použitím třídy <i>Sampler</i> . . . . .	42



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Játra . . . . .	2
<b>2</b>	<b>Technologie akvizice volumetrických dat</b>	<b>4</b>
2.1	Získání cévní soustavy jater . . . . .	4
2.2	Výpočetní tomografie (CT) . . . . .	5
2.3	Rentgenová mikrotomografie ( $\mu$ CT) . . . . .	7
<b>3</b>	<b>Obecné metody registrace obrazů</b>	<b>9</b>
3.1	Kritéria dělení metod registrací . . . . .	11
3.1.1	Dimenze dat . . . . .	11
3.1.2	Povaha registrační báze . . . . .	11
3.1.3	Transformace . . . . .	12
3.1.4	Obor transformace . . . . .	13
3.1.5	Interakce . . . . .	14
3.1.6	Optimalizační proces . . . . .	14
3.1.7	Modalita . . . . .	15
3.1.8	Subjekt . . . . .	15
3.1.9	Objekt . . . . .	15
3.2	Příklady registračních metod . . . . .	16
3.2.1	Korelační a sekvenční metody . . . . .	16
3.2.2	Metody založené na Fourierově transformaci . . . . .	18
3.2.3	Point mapping . . . . .	20

<b>4</b>	<b>Algoritmus multimodální registrace</b>	<b>23</b>
4.1	Návrh . . . . .	23
4.2	Implementace . . . . .	25
4.2.1	IData . . . . .	26
4.2.2	ISampler . . . . .	30
4.2.3	IFeatureComputer . . . . .	30
4.2.4	IMatcher . . . . .	31
4.2.5	ITransformer . . . . .	31
<b>5</b>	<b>Testování vybraných částí algoritmu</b>	<b>34</b>
5.1	VolumetricData . . . . .	35
5.2	IFeatureComputer . . . . .	36
5.3	Transformer3D . . . . .	38
<b>6</b>	<b>Závěr</b>	<b>43</b>
	<b>Literatura</b>	<b>44</b>

# 1 Úvod

Téma registrace obrazů je velmi aktuální, neboť transformování jednoho obrazu na druhý (ať už se jedná o dvourozměrné, povrchové, nebo objemové obrazy) je důležitou součástí mnoha oborů. Příkladem může být počítačové vidění, vojenské automatické detekování cíle, dálkový průzkum Země či medicínské zpracování obrazů. Motivací k zabývání se tématem registrace dat v této práci je snaha získat představu o tom, kolikrát se rozvětví cévy v játrech než přejdou do jaterních sinusoid.

Resekce jaterní tkáně (odstranění části jaterní tkáně) u pacienta je vyžadována v nejrůznějších případech, nejčastěji při výskytu zhoubných i nezahubných nádorů nebo metastáz. Pokud je operace úspěšná, může pacienta vyléčit, či mu významně prodloužit život, v opačném případě může pacient zemřít. Aby se snížilo riziko špatně provedené operace, využívají lékaři nejrůznějších moderních technologií, např. vizualizaci jater před vlastní operací a návrh řezu na tomto modelu.

Pro další usnadnění příprav na operaci je třeba zjistit počet rozvětvení cévy před jejím přechodem do jaterních sinusoid. Vzhledem k tomu, že model jater vychází z výpočetní tomografie (CT) snímků které nemají dostatečně vysoké rozlišení, je snaha získat chybějící informace z rentgenové mikrotomografie ( $\mu$ CT), které má rozlišovací schopnost mnohonásobně vyšší. Zařízení  $\mu$ CT však pracuje pouze s malými vzorky, což znamená, že játra by musela být nařezána. Pro testovací účely byl tedy vytvořen pryskyřicový model cévní soustavy jater prasete (jsou podobná lidským játrům), který mohl být nařezán.

Cílem této práce je navrhnout algoritmus pro registraci trojrozměrných (volumetrických) dat a implementovat navržený postup tak, aby umožňoval nahrazení jednotlivých bloků alternativními algoritmy. Ze zadání je zřejmé, že výsledný algoritmus by měl umožňovat rigidní registraci jakýchkoli volumetrických dat, avšak testování i teoretická část práce budou zaměřeny na registraci volumetrických dat pořízených  $\mu$ CT na volumetrická data pořízená lékařským CT.

## 1.1 Játra

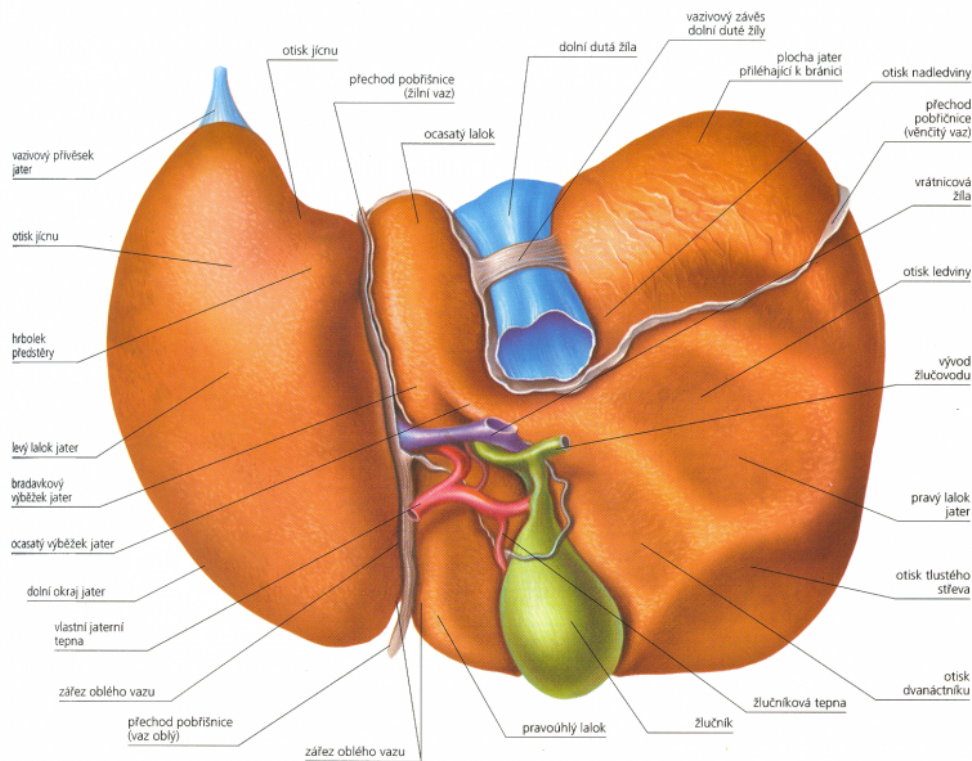
Údaje v této kapitole vycházejí z prací [1] a [2]. Játra (v medicíně ustálený řecký název - *hepar*) patří mezi nejdůležitější a největší vnitřní orgány. Nacházejí se pod pravou brániční klenbou a svojí levou částí zasahují až k pomyslné svislici spuštěné středem klíční kosti (medioklavikulární čára). V dospělém věku mají játra hmotnost v rozmezí 1 až 2.5 kg. V embryonálním stádiu slouží játra ke krvetvorbě a v pozdějším věku jsou velmi důležitou exokrinní žlázou účastnící se metabolismu. Jsou zásadní pro udržování stálosti vnitřního prostředí, podílejí se na detoxikaci organismu a jsou důležitá pro imunitní procesy. V játrech se vytváří žluč, která díky svému hydrotropnímu účinku (zvyšuje rozpustnost nepolárních sloučenin ve vodě) umožňuje snazší vstřebávání tuků a vylučování nepotřebných a škodlivých látek z těla. Při závažném onemocnění nebo poškození jater dochází k úmrtí.

Mezi základní metabolické funkce jater patří zejména syntéza látek jako např. glykogenu, aminocukrů nebo uronových kyselin. Díky glykogenu jsou játra schopna rychle regulovat hladinu glukózy v krvi, což je životně důležitý proces. Další nezbytné metabolické funkce jsou látková přeměna fruktózy a galaktózy a přeměna sacharidů na glukózu.

Dalším chemickým procesem, který probíhá pouze v játrech, je oxidace mastných kyselin s konečným cílem získat energii a produkce ketolátek. Mimo to jsou játra hlavním místem pro přeměnu nadbytečné glukózy na mastné kyseliny a triacylglyceroly, které jsou následně exportovány a uloženy do tukové tkáně. Mezi důležité látky, které jsou v játrech syntetizovány, patří cholesterol, fosfolipidy, VLDL (z anglického *very low density lipoprotein* - lipoproteiny s velmi nízkou hustotou) a HDL (z anglického *high density lipoprotein* - lipoproteiny s vysokou hustotou). Část cholesterolů a fosfolipidů je jako součást lipoproteinů vyloučena do žluči.

Mnoho neesenciálních aminokyselin je syntetizováno právě v játrech, jejichž úkolem je regulovat hladinu aminokyselin v plazmě. Při přeměně aminokyselin se však uvolňuje toxický amoniak, jehož detoxikaci zajišťuje tvorba močoviny probíhající také v játrech.

Játra jsou jaterními rýhami rozdělena na čtyři laloky a vstupují jimi do jater cévy a vystupují žlučové vývody. Vrátnicová žíla (*v. portae*) přivádí do jater krev z dutiny břišní, obsahující látky vstřebané z potravy ke zpracování jaterními buňkami. Jaterní tepnou (*a. hepatica propria*) přitéká okysličená krev vyživující např. stěny velkých žil nebo vazivo. Žíla s tepnou se dále větví až do center lalůčků, kde je krev sbírána do odtokových jaterních žil (*venae hepaticae*), které ústí do dolní duté žíly (*v. cava inferior*) (obrázek 1.1). Průměrně játry proteče 1.5 l krve za minutu.



Obrázek 1.1: Anatomie jater (převzato z [3]).

## 2 Technologie akvizice volumetrických dat

Volumetrická (objemová) data jsou nejčastěji získávána z medicínských přístrojů jako jsou např. výpočetní tomografie (CT), magnetická rezonance (MRI), rentgen (RTG) a další. V jiných odvětvích jsou objemová data získávána z 3D skenerů nebo jako výsledek počítačových výpočtů. Z názvu lze odvodit, že tato data popisují nejen povrch, ale i objem skenovaného objektu. V medicíně je pořizování volumetrických obrazů velmi časté kvůli zjednodušení vyšetření a zejména z důvodu možnosti vyloučení operace pacienta [4, 5].

Základní jednotkou objemových dat je tzv. *voxel* (objemový pixel, popř. objemový element), který se dá vyjádřit prostorovou funkcí  $\nu = \rho(x, y, z)$ . Samotná data jsou diskrétní - voxely rozmístěné na vrcholech pomyslné trojrozměrné mřížky, která má konstantní vzorkovací vzdálenost ve všech třech směrech. Pro každý směr však může být vzdálenost různá. Dalším pojmem je řez, kterým se rozumí podmnožina dat s jednou souřadnicí konstantní [4, 5].

### 2.1 Získání cévní soustavy jater

Pro tuto práci byla použita data získaná CT a  $\mu$ CT přístroji (viz. kapitoly 2.2 a 2.3) skenováním pryskyřicového modelu (tzv. korozivní preparát) prasečích jater. Prase představuje ve výzkumu jater modelový organismus z důvodu, že struktura prasečích jater je velmi podobná lidským játrům. Eberlová [6] popisuje získání zmiňovaného modelu.

Po vyjmutí prasečích jater do nich byla vrátnicovou žilou napuštěna pryskyřice *Biodur E20 Plus*. Po 24 h byla 15% roztokem chloridu draselného odstraněna jaterní tkáň, čímž vznikl 3D model cévní soustavy jater prasete (viz obrázek 2.1). Vzorky pro  $\mu$ CT byly pořízeny odříznutím části modelu v oblasti laloku, jeho propláchnutím alkoholem, zmražením v destilované vodě a následným nařezáním na požadované velikosti vzorků. Postup byl schválen komisí pro práci s experimentálními zvířaty a se zvířaty bylo nakládáno v souladu se zákonem České republiky a legislativou Evropské unie [6].

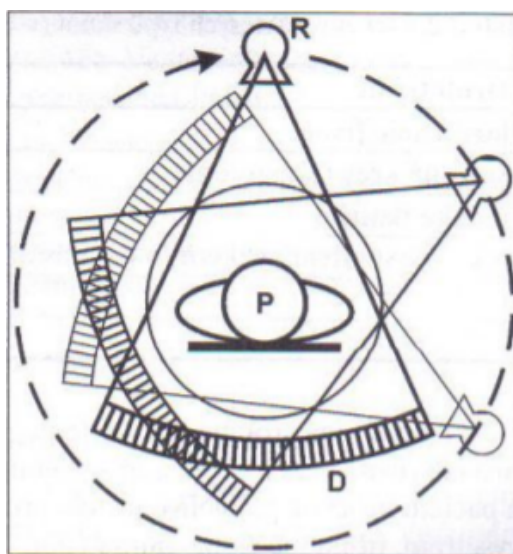


Obrázek 2.1: 3D model cévní soustavy jater prasete (převzato z [7]). Na obrázku je pořízený model jater, jehož CT snímky a následné  $\mu$ CT snímky jeho částí jsou použity v této práci.

## 2.2 Výpočetní tomografie (CT)

Kapitola o výpočetní tomografii čerpá z knih o radiologii [8] a [9]. Tato zobrazovací metoda má široké použití - mezi nejčastější důvody vyšetření CT přístrojem patří zranění lebky nebo páteře, zranění v oblastech hrudníku a břicha, cévní mozková příhoda a vyloučení či potvrzení přítomnosti nádorů. Další možností využití výpočetní tomografie je hledání pooperačních komplikací.

Jak už název napovídá, CT je řazeno do metod tomografických, tj. metod, které pořizují obrazy v řezech. Šířka jednotlivých vrstev (skenů) nabývá hodnot v rozmezí  $0.5 - 5 \text{ mm}$ . Princip snímkování spočívá v detekování intenzity prošlého rentgenového záření. Nejdůležitějšími částmi přístroje jsou rentgenka (vyzařuje rentgenové záření) a systém detektorů. Záření vysílané ve tvaru vějíře vychází z rentgenky, projde pacientem a dopadá na detektory umístěné naproti rentgence (viz obrázek 2.2).



Obrázek 2.2: Princip fungování CT (převzato z [8]). Rentgenka (R) se pohybuje naznačeným směrem a vysílá rentgenové záření, které projde pacientem (P) a je zachyceno systémem detektorů (D).

Tam je množství zachyceného záření převedeno na elektrický signál, který je dále zpracován počítačem. Pro zhotovení jednoho skenu se musí rentgenka otočit kolem pacienta o  $360^\circ$ , což trvá přibližně  $0.3 - 2$  s. Detektory, kterých bývá  $800 - 1200$ , jsou umístěny vedle sebe ve více tenkých řadách. Během rotace každý detektor zaznamená velké množství dat ( $720 - 1400$  měření).

Tomografické snímky jsou pořizovány buď konvenční nebo spirální technikou v axiální rovině. Při konvenčním vyšetření je stůl s pacientem posouván o konstantní vzdálenost. Mezi jednotlivými posuny je pořízen sken. Spirální vyšetření spočívá v posouvání pacienta stálou rychlostí přístrojem, přičemž jsou pořizována data, z nichž počítač rekonstruuje jednotlivé skeny.

Jednotlivé vrstvy jsou získávány v digitální podobě jako dvourozměrné pole hodnot, obvyklé rozměry jednoho skenu jsou  $512 \times 512$  bodů. Hodnoty vypočítané počítačem, označující míru oslabení záření, z pohledu volumetrických dat nazývané voxely, jsou ve výpočetní tomografii nazývány *denzita* a jsou udávány v *Hounsfieldových jednotkách - HU*. Densita nabývá hodnot od  $-1000 HU$  do  $3096 HU$  a její rozložení se řídí dvěma základními body - densitou vzduchu ( $-1000 HU$ ) a densitou destilované vody ( $0 HU$ ). V tabulce 2.1 jsou uvedeny denzity tkání nejčastěji vyšetřovaných CT metodou.



Druh tkáně	Denzita [HU]
<b>vzduch</b>	<b>-1000</b>
<b>voda</b>	<b>0</b>
kosti, kalcifikace	> 90
měkké tkáně	25 – 70
sražená krev	65 – 85
krev v cévách	40

Tabulka 2.1: Density některých typů tkání (podle [9]).

Řezy z CT dat jsou zobrazovány ve stupních šedi (oko rozliší 60 odstínů), na něž musí být denzity přeškálovány podle toho, které tkáně je potřeba zkoumat. Tomuto procesu se říká *windowing*.

## 2.3 Rentgenová mikrotomografie ( $\mu$ CT)

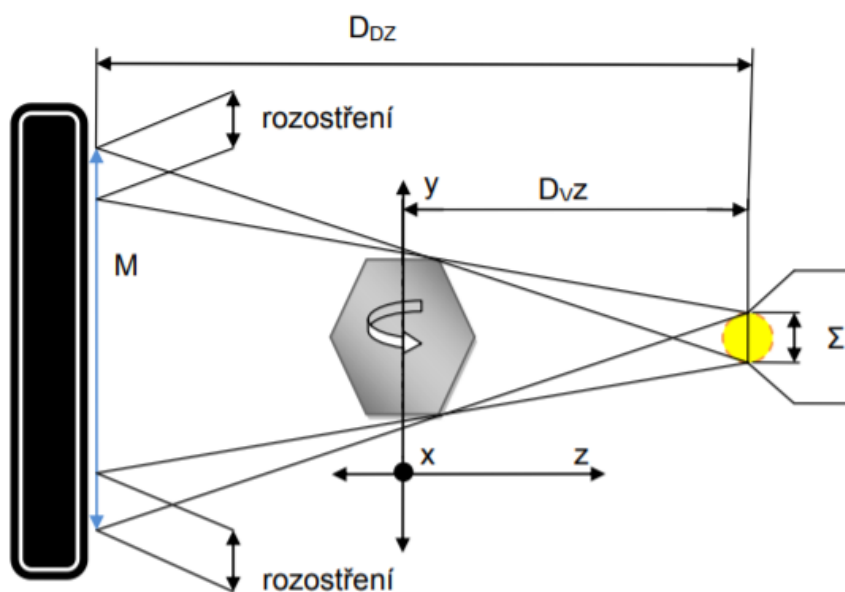
Podobně jako CT je i  $\mu$ CT (také mikro-CT) zobrazovací metodou, která však neslouží k lékařským vyšetřením, ale zejména v technických oblastech, např. pro zjišťování vad materiálů (pórovitost, trhliny) nebo měření rozměrů a geometrických tolerancí, pro něž jsou kontaktní měřicí metody nedostačující. V biomedicíně je dobrým příkladem zkoumání opotřebených endoprotéz. Ceněnou vlastností  $\mu$ CT skenování je jeho nedestruktivnost - metoda je tedy použitelná i pro vzácné artefakty [10, 11].

Z pohledu pořizování dat (snímků) je  $\mu$ CT velmi podobné počítačové tomografii - také se jedná o tomografickou metodu a snímkování je založeno na různém zeslabení rentgenového záření při průchodu různými materiály. Hlavními částmi tomografu jsou opět rentgenka a detektor, mezi nimi je polohovací zařízení pro umístění vzorku. Na rozdíl od CT je v případě  $\mu$ CT otáčeno vzorkem, nikoli rentgenkou. Zařízení otáčí objektem o  $360^\circ$  po krocích daných vstupním krokovým úhlem. Po každém pootočení je vytvořen dvourozměrný sken, stejně jako v případě CT.

Zvětšení zkoumaného objektu je dáno vztahem vzdálenosti mezi detektorem a rentgenkou (zdrojem) a vzdálenosti mezi vzorkem a rentgenkou, matematicky vyjádřeno následovně:

$$M = \frac{D_{DZ}}{D_{VZ}} \quad (2.1)$$

kde  $D_{DZ}$  je vzdálenost detektor-zdroj a  $D_{VZ}$  vzdálenost vzorek-zdroj (viz obrázek 2.3) [10, 11].



Obrázek 2.3: Princip fungování  $\mu$ CT (převzato z [10]). Vlevo na obrázku se nachází detektor, uprostřed je na polohovacím zařízení umístěn vzorek (na obrázku šestiúhelník), kterým je otáčeno a v pravo se nachází rentgenka (znázorněna žlutým kruhem), neboli zdroj. Zvětšení  $M$  je dáno podílem vzdálenosti detektor-zdroj ( $D_{DZ}$ ) a vzdálenosti vzorek-zdroj ( $D_{VZ}$ ).

Při pořizování snímků nedochází k normalizaci dat podle známých hodnot (jako je tomu v případě CT podle vzduchu a vody) a navíc nejsou hodnoty získávány v *Hounsfieldových jednotkách*. Obrazy jsou ovšem detailnější a díky vysokému rozlišení jsou na nich vidět i vlásečnice a jaterní sinusoidy, které dosahují průměrů v rozmezí  $7 - 50 \mu m$  [12] a které klasické medicínské CT není schopno rozlišit.

# 3 Obecné metody registrace obrazů

Registrace obrazů je hledání vztahu mezi dvěma obrazy definováním transformační funkce  $f$ , která transformuje obraz A na obraz B tak, aby se znaky obsažené v obou obrazech překrývaly. Podle [13] je registrace obrazů nejen hledání transformační funkce v prostoru, ale i z hlediska intenzity dat, která se různí například při použití různých senzorů nebo různou odrazivostí objektů. Registrace dvou obrazů lze tedy zapsat matematicky takto:

$$B(x, y) = g(A(f(x, y))) \quad (3.1)$$

pro 2D obrazy a analogicky

$$B(x, y, z) = g(A(f(x, y, z))) \quad (3.2)$$

pro 3D obrazy, kde funkce  $f$  je zmiňovaná prostorová transformace bodů  $x, y$  ( $x, y, z$ ) na  $x', y'$  ( $x', y', z'$ )

$$(x', y') = f(x, y) \quad (3.3)$$

(analogicky pro 3D) a funkce  $g$  je jednodimenziální transformace intenzity.

Registrace obrazů je využívána v mnoha sférách. Lisa G. Brown ve své práci [13] rozděluje problémy registrace do čtyř tříd:

- Multimodální registrace (*Multimodal registration*)

Tato třída zahrnuje problémy zabývající se registrací obrazů stejné scény (objektu), ale pořizovaných různými senzory.

Metody používané v této oblasti často, z pohledu zdroje dat, využívají modely sensorů nebo předzpracování intenzit pixelů/voxelů transformační funkcí  $g$  (viz funkce 3.1/ 3.2). Příkladem problému řadícího se do této třídy je analýza medicínských obrazů (CT, MRI, PET (pozitronová emisní tomografie)) nebo zpracování dálkově získaných obrazů (radar, infračervené záření, mikrovlny). V případě obrazů z lékařského prostředí se jedná o rozšíření informací poskytovaných obrazem - CT a MRI poskytují informace o struktuře, zatímco PET nebo SPECT (jednofotonová emisní výpočetní tomografie) poskytují informace funkční. Kombinací těchto informací jsou lékaři schopni přesněji nalézt místa metabolických změn. Ve druhém případě lze s pomocí registrace zlepšit klasifikaci geografických objektů jako jsou budovy, silnice nebo některé rostlinstvo.

- Registrace podle daného vzoru v obrazu (*Template registration*)  
Principem řešení problémů této třídy je hledání shodného obrazce se známým vzorem. Často se metody spadající do této třídy využívají při hledání map nebo modelových objektů. Metody jsou charakteristické přednastavenými parametry a známými vlastnostmi objektů. Příkladem použití je, stejně jako v předchozí třídě, zpracování dat pořízených dálkovými snímači, pokud jsou scény dobře definovány, tj. známe prvky, které scéna určitě bude obsahovat, jako např. ranveje na letišti. Dalším příkladem může být rozpoznávání vzoru např. podpisů nebo osob.
- Registrace obrazů pořízených z různých pohledů (*Viewpoint registration*)  
Nejčastější využití metod náležících do této třídy je rekonstrukce hloubek nebo tvarů. Využívají se v oblasti počítačového vidění kvůli zmíněnému obnovení nesrovnalostí v hloubkách nebo tvarech. Dalším využitím je např. sledování pohybu objektu.
- Vývoj obrazů v čase nebo po změně podmínek (*Temporal registration*)  
Obrazy, které jsou registrovány metodami z této třídy, byly pořízeny ze stejné scény, ale v jiném čase, nebo poté, co se změnila nějaké zásadní podmínky. To se používá např. pro sledování změn nebo růstu. Příklady problémů, pro které jsou metody sledování změn v čase používány, jsou detekování nádorů, sledování změn po podání radioizotopů a monitorování přírodních zdrojů či růst měst.

V této práci je řešen problém registrace objemových medicínských obrazů. Tento konkrétní problém spadá do třídy multimodální registrace, kdy jedna data jsou pořízena počítačovou tomografií a druhá data mikrotomografií.

Kromě dělení do tříd podle povahy problému se metody registrace obrazů dají dělit podle různých dalších kritérií. Základní rozdělení metod (viz [14–16]) zahrnuje devět kritérií. Ta jsou podrobněji popsána v následující části.

## 3.1 Kritéria dělení metod registrací

### 3.1.1 Dimenze dat

Nejčastější registrací z pohledu tohoto kritéria je registrace  $2D-2D$ , kdy oba registrované obrazy jsou dvoudimenzionální, např. porovnávání skenů z CT nebo klasických rentgenových snímků. Často je také používána registrace  $3D-3D$ , oba obrazy jsou trojtrozměrné, a může se jednat např. o porovnávání dat z CT nebo MRI. Méně častým případem je registrace  $2D-3D$ , jejímž příkladem je porovnání rentgenového snímku s daty pořízenými výpočetní tomografií. Kromě prostorových dimenzí může jednou dimenzí být i čas v případě sledování časové změny. Příkladem v medicíně je pozorování hojení pacienta či růst nádoru, v jiných oblastech to může být monitorování hadiny oceánů, dezertifikace či tání ledovců.

### 3.1.2 Povaha registrační báze

Z pohledu tohoto měřítka se může jednat o registraci vnitřní (*intrinsic*) a vnější (*extrinsic*). Vnější registrace je využívána zejména v medicíně, kdy je na pacienta umístěn vnější objekt, který má napomoci orientaci v obrazu a následné registraci dvou či více obrazů. Druhý zmíněný typ využívá pouze informací obrazu, nejčastěji v případě, že samotný obraz obsahuje vlíčovací body, např. roh nebo rozcestí. Při dálkovém průzkumu Země se za tyto body dají považovat třeba řeky, silnice, domy apod. v lékařství poté orgány, kosti nebo větvení cév. V jistých případech může jít i o registraci, která není založena na obrazech samotných, ale na parametrech senzorů, kterými byly obrazy pořízeny. Poté musí existovat vztah mezi daty generovanými jedním a druhým přístrojem.

### 3.1.3 Transformace

Transformační funkce je obvykle vyjádřena transformací (jejím maticovým zápisem) a jak již bylo zmíněno (viz rovnice 3.1 a 3.2), jejím úkolem je mapovat bod ze souřadnic jednoho obrazu do souřadnic obrazu druhého. Existují různé druhy transformací, mezi nejčastější patří rigidní, afinní, projektivní (perspektivní) a obecná polynomiální (elastická nebo transformace křivkami). Kapitola o transformacích vychází z prací [13, 17] a z prezentace [18].

Pro rigidní transformaci je předpokladem zachování velikosti a tvaru objektů na obou obrazech. Tvoří ji rotace a translace a zapisuje se následujícím způsobem:

$$\vec{y} = R\vec{x} + \vec{t} \quad (3.4)$$

kde  $\vec{y}$  je vektor souřadnic v prostoru obrazu  $B$  (do těchto souřadnic transformujeme) a  $\vec{x}$  je vektor souřadnic obrazu  $A$  (tento obraz transformujeme).  $R$  je rotační matice  $3 \times 3$  a  $\vec{t}$  vektor translace.

Maticový zápis rigidní transformace je následující:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \left( \begin{array}{ccc|c} & & & t \\ R & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} \quad (3.5)$$

Rotační matice je definována jako součin rotací kolem os  $x$ ,  $y$  a  $z$ :

$$R = R_x R_y R_z \quad (3.6)$$

$$\begin{aligned} R_x &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_1 & -\sin \alpha_1 \\ 0 & \sin \alpha_1 & \cos \alpha_1 \end{pmatrix}, \\ R_y &= \begin{pmatrix} \cos \alpha_2 & 0 & \sin \alpha_2 \\ 0 & 1 & 0 \\ -\sin \alpha_2 & 0 & \cos \alpha_2 \end{pmatrix}, \\ R_z &= \begin{pmatrix} \cos \alpha_3 & -\sin \alpha_3 & 0 \\ \sin \alpha_3 & \cos \alpha_3 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (3.7)$$

Afinní transformace jsou obecnější a nezachovávají vzdálenosti, tj. mapují na sebe rovnoběžné přímky, které však nemusejí být stejně dlouhé. Možný zápis takové transformace je následující:

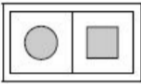
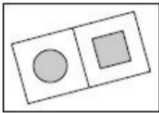
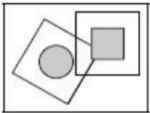
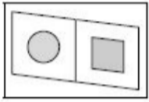
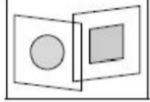
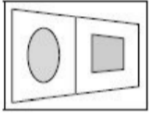
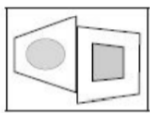
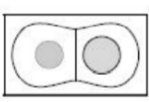
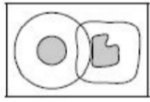
$$\vec{y} = sR\vec{x} + \vec{t} \quad (3.8)$$

kde  $s$  je kladný skalár, který zmenšuje či zvětšuje obraz.

Projektivní a obecnější perspektivní transformace počítají s deformacemi a mapují na sebe přímky (nikoli rovnoběžné). Polynomiální transformace jsou nejobecnější a počítají s nejrůznějšími deformacemi v obrazu, mapují přímky na křivky.

### 3.1.4 Obor transformace

Toto kritérium záleží na povaze obrazů. Pokud je třeba transformovat celý obraz, je transformace nazývána globální. V případě, že je nutné transformovat dílčí části obrazu zvlášť, je každá jednotlivá transformace označována jako lokální.

Originál	Transformace aplikovaná globálně	Transformace aplikovaná lokálně	Typ transformace
			Rigidní
			Afinní
			Perspektivní
			Elastická

Obrázek 3.1: Srovnání různých transformací (převzato z [18]). Na obrázku je znázorněn rozdíl mezi jednotlivými transformacemi a zejména rozdíl mezi obory transformace.

### 3.1.5 Interakce

Registrační metody, z hlediska jejich interakce s uživatelem, jsou děleny na automatické, poloautomatické nebo také semiautomatické a interaktivní. V případě automatických metod je od uživatele vyžadováno pouze zadání vstupních obrazů, případně rozšiřujících informací o senzorech, kterými byla data pořízena, či čas pořízení. Do samotného procesu registrace však uživatel nezasahuje. Poloautomatické metody vyžadují buď zadání počátečního řešení, nebo potvrzení řešení vypočítaného algoritmem. Interaktivní registraci řídí prakticky sám uživatel, algoritmus ho pouze kontroluje (např. zadávání parametrů) nebo dopočítá dílčí části.

### 3.1.6 Optimalizační proces

Při hledání konečné transformace může algoritmus narazit na lokálně správné, ale globálně zcela nesprávné transformace, proto je většinou celý proces registrace prováděn iterativně. Optimalizace je tedy volba kritériální funkce, která pomáhá rozhodnout, kdy je vypočítaná transformace pravděpodobně správná a tedy může být algoritmus ukončen.



### 3.1.7 Modalita

Z hlediska modality (typ zobrazovací metody) jsou rozeznávány dva druhy registrace - monomodální a multimodální. Pokud všechny zahrnuté obrazy vznikly stejnou metodou, jedná se o monomodální metody registrace. Naopak, pokud jsou zdroje obrazů různé, jedná se o registrační metody multimodální. Výhodou multimodálních metod je možnost zobrazit na jednom obraze struktury, které by žádné zařízení najednou pořídit nemohlo.

### 3.1.8 Subjekt

Toto kritérium se týká zejména lékařských obrazů a dělí registrace na intrasubjektivní, intersubjektivní a registraci podle atlasu. První případ zahrnuje data pouze od jednoho pacienta, naopak druhý případ zahrnuje data od více pacientů. Atlasová registrace porovnává obrazy jednoho pacienta s atlasem - teoreticky ideálním souborem dat, který byl pořízen od velkého množství pacientů.

### 3.1.9 Objekt

Posledním kritériem dělení registrací je samotný objekt zájmu. V lékařství se jedná o jednotlivé části těla nebo orgány, při dálkovém průzkumu Země může být objektem celá Země, nebo její části - pouště, oceány, města, atd. Podle zvoleného objektu je nutné vybrat modalitu a typ transformace. Příkladem může být lebka, která se za normálních okolností nedeformuje a játra, jejichž snímky se (pokud jsou pořizovány zaživa) mění. Pro lebku by tedy stačila rigidní transformace, zatímco pro játra by bylo nutné volit transformaci křivkovou, která by vykompenzovala nelinearity vzniklé metabolickými změnami.

## 3.2 Příklady registračních metod

Následující text čerpá z práce [13].

### 3.2.1 Korelační a sekvenční metody

Základním statistickým přístupem k registraci obrazů je křížová korelace, která však sama o sobě není registrační metodou. Korelačními jsou nazývány takové metody, pro jejichž algoritmy je výpočet křížové korelace nezbytný a zároveň je jejich nejdůležitějším prvkem. Pro vzor  $T$  (z anglického *template*) a obraz  $I$  (z anglického *image*) je vzorec pro výpočet normované křížové korelace pro každou translaci  $(u, v)$  následující:

$$C(u, v) = \frac{\sum_x \sum_y T(x, y) I(x - u, y - v)}{\sqrt{[\sum_x \sum_y I^2(x - u, y - v)]}}. \quad (3.9)$$

Pokud je vyžadována absolutní míra, je výhodné použít korelační koeficient, který se vypočítá následujícím způsobem:

$$\frac{\text{cov}(I, T)}{\sigma_I \sigma_T} = \frac{\sum_x \sum_y (T(x, y) - \mu_T)(I(x - u, y - v) - \mu_I)}{\sqrt{\sum_x \sum_y (I(x - u, y - v) - \mu_I)^2 \sum_x \sum_y (T(x, y) - \mu_T)^2}}, \quad (3.10)$$

kde  $\mu_T$  a  $\mu_I$  jsou střední hodnoty a  $\sigma_T$  a  $\sigma_I$  jsou směrodatné odchylky. Korelační koeficient nabývá hodnot z intervalu  $[-1, 1]$  a je mírou důvěry v to, že obraz je instancí vzoru.

Často jsou tyto metody používány při hledání vzorů v obrazu nebo k rozpoznávání vzorců. Užitečné jsou pro registraci obrazů s malou rigidní nebo afinní transformací. Algoritmus na malém rozsahu translací, rotací a koeficientů přeškálování provede transformaci pro všechny kombinace zmíněných operací. Ta transformace vzoru, která má nejvyšší korelaci (nebo jinou míru, např. korelační koeficient) s obrazem, je na daném rozsahu optimální. Se zvyšujícími se rozsahy se ovšem zvyšuje časová náročnost algoritmu a to je důvod, proč se nedá použít pro větší transformace.

Pro snížení časové náročnosti algoritmu lze použít namísto celého obrazu jeho části a to nejlépe ty, které jsou výrazné oproti jiným částem, jako např. hrany tvaru Y nebo T.

Pokud jsou v obrazu významná zkruslení, která nelze odstranit transformací, nebude maximum korelace jasně rozeznatelné. Pro některé druhy šumů, jako je např. bílý šum, se jako filtr křížové korelace použije vzor ( $T$ ). V ostatních případech musí obraz projít dvojitou filtrací, tj. před vypočítáním jeho korelace se vzorem je ještě předzpracován. První filtr může být založen na vlastnostech šumu, pokud se dá šum alespoň částečně popsat statisticky. Metody, které jsou založené na této dvojitou filtraci obrazu se nazývají techniky spojených filtrů (z anglického *matched filter techniques* - MFT) a jejich cílem je maximalizovat korelaci s ohledem na šum. Jejich nevýhodou je možná výpočetní náročnost a obtížná splnitelnost statistických předpokladů šumu.

Z tohoto pohledu efektivnější algoritmy jsou algoritmy pro detekci sekvenční podobnosti (z anglického *sequential similarity detection algorithms* - SSDAs), které mají oproti MFT dvě zásadní vylepšení. Prvním je volba jiné podobnostní míry  $E(u, v)$ , která je výpočetně mnohem rychlejší než křížová korelace a její normalizovaný předpis je následující:

$$E(u, v) = \sum_x \sum_y |T(x, y) - \hat{T} - I(x - u, y - v) + \hat{I}(u, v)|, \quad (3.11)$$

kde  $\hat{T}$  a  $\hat{I}$  jsou průměrné intenzity vzoru a obrazu, respektive některé jeho části.

Druhým vylepšením SSDAs je sekvenční vyhledávací strategie. Tato strategie se různí dle požadované přesnosti a rychlosti a také dle složitosti dat. Pro jednoduchou translaci může být vyhledávací strategie např. sekvenční prahování. V tom případě je obraz rozdělen na okna (podle testované translace a velikosti vzoru) a velikosti měř bodů jsou sčítány, dokud tato suma nepřekročí daný práh. Počet otestovaných bodů do překročení prahu pro každé okno poté ukazuje na to, jak moc je daná translace správná. To okno s nejvyšším počtem otestovaných bodů je tedy považováno za nejlepší translaci.

Ačkoli sekvenční metody zvyšují rychlost registrace, pořád zůstává problém u složitých transformací, které mnohonásobně zvyšují počet průchodů algoritmem. Sekvenční algoritmus je v tomto okamžiku velice důležitý pro zachování rozumné doby výpočtu transformace, na druhou stranu se může stát, že vynechá důležité body a tedy nenajde optimální transformaci.

Ve srovnání s korelací jsou sekvenční metody řádově efektivnější, avšak výsledky metod se mohou různit a to ve prospěch obou zmíněných. Nedostatkem obou metod je neschopnost zpracovat obrazy, které nejsou identické, tj. kdyby vzor byl např. světlejší nebo tmavší. V takových případech je vhodné použít metody založené na Fourierově transformaci, které budou popsány v následující kapitole.

### 3.2.2 Metody založené na Fourierově transformaci

Stejně jako v případě korelačních a sekvenčních metod, jsou i tyto vhodné zejména pro malé transformace. Oproti nim však nemohou být použity pro afinní a obecné transformace, ale mohou být aplikovány pouze pokud Fourierova transformace transformovaného obrazu matematicky souvisí s původním obrazem. Vhodné jsou obzvláště pro obrazy s nízkou frekvencí nebo obrazy s šumem závislým na frekvenci, tj. pro obrazy s bílým šumem jsou tyto techniky nevhodné.

Na rozdíl od předchozích metod, metody založené na Fourierově transformaci hledají optimální transformaci na základě informací ve frekvenční oblasti. Díky tomu jsou odolné proti korelovanému šumu a šumu závislému na frekvencích.

Fourierova transformace obrazu  $f(x, y)$  je komplexní funkce vyjádřená následovně:

$$F(\omega_x, \omega_y) = R(\omega_x, \omega_y) + iI(\omega_x, \omega_y), \quad \forall (\omega_x, \omega_y) \text{ na frekvenčním spektru,} \quad (3.12)$$

kde  $i$  je imaginární jednotka,  $R(\omega_x, \omega_y)$  reálná část a  $I(\omega_x, \omega_y)$  imaginární část.

Tuto rovnici lze vyjádřit také jako součin amplitudy a fáze frekvenční transformace:

$$F(\omega_x, \omega_y) = |F(\omega_x, \omega_y)| e^{i\phi(\omega_x, \omega_y)}, \quad (3.13)$$

kde  $|F(\omega_x, \omega_y)|^2 = R^2(\omega_x, \omega_y) + I^2(\omega_x, \omega_y)$  je rovna energii na každé frekvenci obrazu a  $\phi(\omega_x, \omega_y) = \tan^{-1}[I(\omega_x, \omega_y)/R(\omega_x, \omega_y)]$  je fázový posun.

Základní metodou vycházející z Fourierovy transformace je tzv. fázová korelace (z anglického *phase correlation*), která registruje obrazy posunuté vůči sobě. Pro dva obrazy  $f_1$  a  $f_2$ , které se liší o posun  $(d_x, d_y)$ , tj.

$$f_2(x, y) = f_1(x - d_x, y - d_y), \quad (3.14)$$

je vztah jejich Fourierových transformací popsán následující rovnicí:

$$F_2(\omega_x, \omega_y) = e^{-i(\omega_x d_x + \omega_y d_y)} F_1(\omega_x, \omega_y)(\omega_x, \omega_y). \quad (3.15)$$

Z toho je vidět, že oba obrazy mají stejnou amplitudu a fázový rozdíl odpovídá jejich posunu. Fázový rozdíl je shodný s fází výkonového spektra, která po převodu z frekvenční do prostorové oblasti pomocí inverzní Fourierovy transformace znázorňuje impuls v posunu, který je optimální registrací obrazů. Tedy fázová korelace je hledání maxima Fourierovy inverzní transformace fáze výkonového spektra. Tato metoda je robustní vůči změnám jasů a částečně i vůči použití různých senzorů pro zachycení téže scény.

Pro registraci obrazů posunutých a zároveň zrotovaných vůči sobě je nutné použít rozšířenou metodu fázové korelace. Tato metoda nejdříve naleznе úhel rotace a poté až posun. Vzhledem k tomu, že rotace je pro Fourierovu transformaci neměnná, pokud je znám úhel rotace lze zrotovat výkonné spektrum a získat posun klasickou fázovou korelací.

Toho se dá využít a vypočítat fázi výkonného spektra jako funkci rotačního úhlu za použití polárních souřadnic  $(r, \theta)$ . Rotační úhel musí být takový, aby inverzní Fourierova transformace této funkce byla co možná nejlepší aproximací impulzu a poté umístění tohoto impulzu je hledaný posun.

Metody zmíněné v této kapitole a kapitole 3.2.1 jsou vhodné pro malé rigidní nebo afinní transformace, ale pouze v případě, že mezi obrazy neexistuje větší množství lokálních nesrovnalostí. Pro tyto případy je vhodnější zvolit metody z následující kapitoly.

### 3.2.3 Point mapping

Nejčastěji používanou technikou k registraci dvou obrazů, u nichž neznáme podstatu jejich transformace, je tzv. *point mapping*<sup>1</sup> nebo také *landmark mapping*. Existuje mnoho metod používajících tuto techniku, zde však budou uvedeny jen některé.

Základní metoda se skládá ze tří fází. V prvních dvou fázích jsou nalezeny vlíčovací body a pro body z referenčního obrazu jsou nalezeny jejich odpovídající protějšky v druhém, datovém, obrazu. Ve třetí fázi jsou za pomoci odpovídajících si bodů vytvořeny funkce (většinou dvě dvourozměrné), které transformují datový obraz na referenční. Vzhledem k tomu, že jsou obrazy složeny z pixelů/voxelů, je při transformaci jednoho na druhý vyžadováno kromě transformační funkce i použití interpolace.

Důležitým prvkem těchto metod jsou vlíčovací (kontrolní) body. Jak již bylo naznačeno v kapitole 3.1.2, vlíčovací body mohou být součástí sledovaných dat, nebo mohou být dodány externě při pořizování snímků. Externí vlíčovací body mohou být například chemické značky v MRI, nebo stereotaktický rám v PET. Tyto značky se dají v datech označit buď ručně, nebo automaticky. Pro ruční určení je třeba někoho, kdo se orientuje v dané problematice, a také lze použít pouze na nepříliš rozsáhlá data. V případě rozsáhlých dat je hledání kontrolních bodů automatizováno a jako kontrolní body jsou nejčastěji hledány rohy, rozcestí, lokální maxima obvodových křivek a další. Čím vyjimečnější body jsou, tím lépe se registrují. Důležitým aspektem je počet hledaných bodů, který musí být dost vysoký kvůli přesnosti vypočítané transformace a zároveň nesmí být moc vysoký kvůli efektivitě výpočtů.

---

<sup>1</sup>*point mapping* - nastavení bodů jednoho obrazu na body druhého obrazu

Získané vlícovací body v obou obrazech jsou následně spojeny se svými protějšky, tj. pro každý bod v jednom obraze je nalezen odpovídající bod v obraze druhém. V případě ručního hledání vlícovacích bodů probíhá spojování bodů zároveň s jejich hledáním. Hledání odpovídajících bodů je, vzhledem k vysokému požadavku na přesnost určení párů, často prováděno lokálně pomocí jiné z registračních metod, např. křížovou korelací. Některé metody nehledají protějšky body ale rovnou protějšky shluky bodů. V takových případech lze zároveň na sebe natočit shluky bodů a tím najít požadovanou transformaci.

Pro případy, kdy hledání vlícovacích bodů a také hledání jejich odpovídajících protějšků je náročné, se používají metody se zpětnou vazbou. Tyto metody jsou iterační a předává se výstupní transformace zpět metodě hledající páry bodů, která na základě této zpětné vazby vyhledá páry bodů znovu a pokud možno lépe. Díky tomuto přístupu jsou metody schopné registrovat obrazy s rigidní, afinní a v některých případech i složitější transformací.

Dalšími metodami, které používají techniku *point mapping* jsou globální polynomiální metody. Parametry jedné optimální transformace, která je jejich výstupem, mohou být nalezeny aproximací, kde je snahou, aby spárované body měly po transformaci jednoho na druhý co nejmenší odchylku. Na to je nejčastěji používána metoda nejmenších čtverců nebo shlukování. Tím pádem nalezená transformace neodpovídá přesně spárovaným bodům, ale je jejich nejlepší aproximací. Parametry transformace mohou být také nalezeny interpolací a to tak, že každému parametru transformace odpovídá přesně jeden pár bodů. Bohužel se zvyšujícím se počtem bodů se zvyšuje počet parametrů, které musí být vypočítány, a tím se i zvyšuje řád polynomiální funkce, která se při vysokém řádu chová nepředvídatelně. V takových případech se používá aproximace metodou nejmenších čtverců nebo spline křivkami.

Protože povaha transformace je neznámá, je obecně používána bivariační polynomiální transformace předepsána následujícími vztahy:

$$u = \sum_{i=0}^m \sum_{j=0}^i a_{ij} x^i y^{j-i} \quad (3.16)$$

$$v = \sum_{i=0}^m \sum_{j=0}^i b_{ij} x^i y^{j-i}, \quad (3.17)$$

kde  $(x, y)$  jsou indexy referenčního obrazu,  $(u, v)$  indexy datového obrazu a  $a_{ij}$  a  $b_{ij}$  jsou parametry polynomu, které musí být vypočítány.

Největším omezením globálních transformací s technikou *point mapping* je, že tyto metody nemohou být použity na místní zkreslení způsobené například vlivem senzorů nebo pořízením obrazů z různých úhlů. Aby se předešlo tomuto omezení, existují lokální metody, které používají interpolaci po částech a zmírňují zkreslení.

V lokálním přístupu je oproti globálnímu vypočítáno mnoho transformací a to buď pro jednotlivé části obrazů, nebo iterativně pro celé obrazy najednou. Tyto metody staví na tom, že pouze body blízké danému okolí mohou přispět do výsledné transformace tohoto okolí. Nejčastěji jsou takové transformace aproximovány množinou spline křivek tvořící povrch a definovanou následovně:

$$S(x, y) = \sum_{i,j} V_{i,j} B_{i,j}(x, y), \quad (3.18)$$

kde  $B_{i,j}(x, y)$  jsou funkce a  $V_{i,j}$  kontrolní body. Existuje množství přístupů k nalezení parametrů těchto funkcí, ale těmi se tato práce nezabývá. V porovnání s ostatními metodami zmíněnými v této práci jsou tyto metody jedny z nejpřesnějších ale také jedny z nejpomalejších.

Nejnovějším přístupem k registraci obrazů je elastická registrace založená na modelu, kdy modelem deformace obrazů jsou deformace elastického materiálu. Jsou velmi podobné metodám používajícím interpolaci po částech, neboť i tyto metody mohou používat spline křivky.



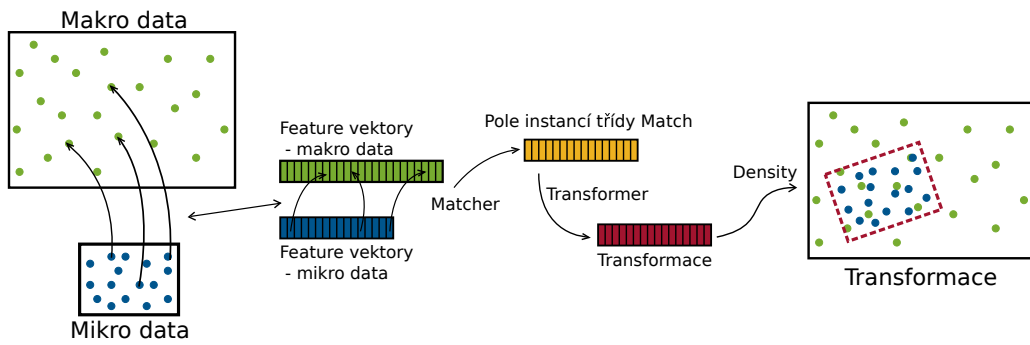
# 4 Algoritmus multimodální registrace

## 4.1 Návrh

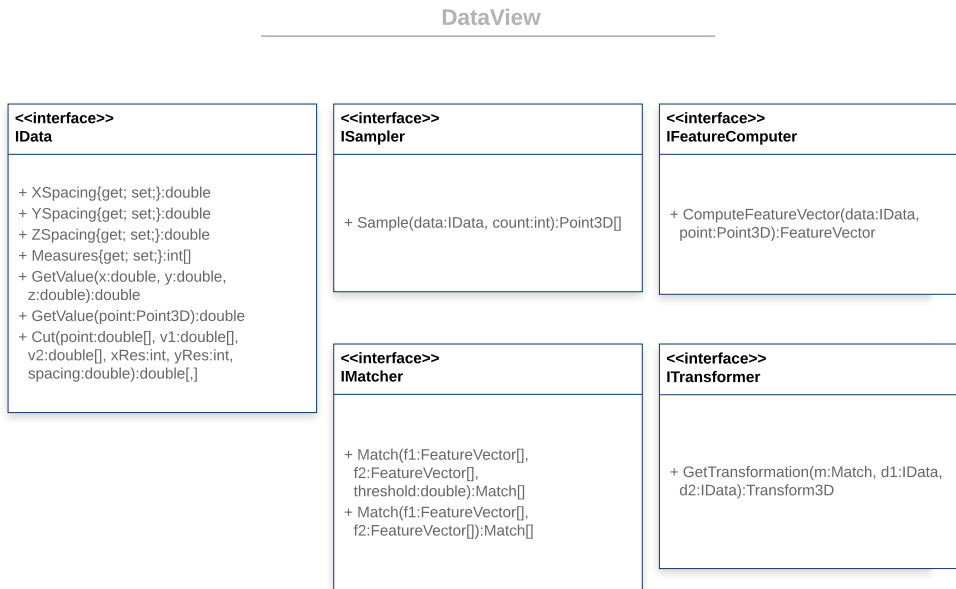
V této práci bude popsán algoritmus registrace dat pořízených  $\mu$ CT přístrojem (dále mikro data, v kódu *micro data* nebo jen *micro*) na data pořízená klasickým lékařským výpočetním tomografem (dále makro data, v kódu *macro data* nebo jen *macro*) nazvaný *DataView*. Cílem algoritmu je z předaných vstupních dat (soubory k mikro a makro datům) vypočítat rigidní transformaci registrující mikro data na makro data. K nalezení jedné optimální transformace využívá *DataView* část algoritmu popsaného v práci [19] (dále označovanou jako *Density*).

*Density* a zmíněný algoritmus jsou navrženy pro globální povrchovou 3D registraci na základě metody RANSAC (*Random Sample Consensus*). Některé algoritmy, také založené na metodě RANSAC, hledají optimální transformaci z kandidátních transformací (vypočítaných některou z metod zmíněných v předchozí části na základě kontrolních bodů v obrazech) volbou té, po jejímž použití se budou obrazy nejvíce překrývat. Algoritmus použitý v práci [19] vychází z předpokladu, že tento přístup k určení optimální transformace nemusí být nejlepším možným zejména kvůli náročnosti ohodnocení výsledku. K tomuto účelu byla navržena právě metoda *Density*, která staví na pozorování, že optimální transformace formuje hustotní maximum v prostoru transformací. Nutnou podmínkou pro správné určení výstupu je využití informací ze vstupních dat, jako jsou rozměry obrazů, k porovnávání jednotlivých (kandidátních) transformací.

Cílem této práce je aplikovat *Density* i pro registraci volumetrických dat namísto pouze povrchových. Kvůli dosažení nejvyšší možné modularity je algoritmus složen z bloků, které na sobě téměř nezávisí a pouze si navzájem předávají potřebné informace. Schéma algoritmu je zobrazeno na obrázku 4.1 a požadavky jednotlivých rozhraní jsou zobrazeny na obrázku 4.2.



Obrázek 4.1: Blokové schéma algoritmu. Schéma znázorňuje směr předávání informací v algoritmu od vstupních dat (mikro a makro data) k výpočtu výsledné transformace.



Obrázek 4.2: Přehled rozhraní a jejich metod. Díky rozhráním lze kterýkoli blok algoritmu kdykoli vylepšit nebo nahradit a nebudou tím nijak narušeny či ovlivněny ostatní části.

Úkolem uživatele je zadat vstupní parametry, kterými jsou soubory .mhd mající informace o mikro a makro datech, která na sebe mají být registrována. *IData* (makro a mikro) mohou být buď reálná nebo umělá data pro testování. V každém případě musí mít instance tohoto rozhraní informace o rozměrech mřížky dat (*Measures*) a také o vzorkování v každém ze tří směrů (*XSpacing*, *YSpacing* a *ZSpacing*). Dále musí instance definovat metodu *GetValue()*, která vrací denzitu ve specifikovaném bodě zadaném v reálných souřadnicích.

Třída implementující rozhraní *ISampler* nemusí mít jinou metodu než metodu *Sample()*, která vrátí pole zadaného počtu bodů nalezených v metodou přebíraných datech (viz modré a zelené body v obrázku 4.1). Metoda *ComputeFeatureVector()* poskytovaná rozhraním *IFeatureComputer* vypočítá pro všechny předané body (nalezené metodou *Sample()*) z dat, jejichž odkaz je také předán metodě, vektor, který popisuje bod a jeho okolí, tzv. feature vektor. Na základě těchto vektorů jsou následně metodou *Match()* rozhraní *IMatcher* nalezeny páry bodů. Vrácený pár, neboli *Match*, obsahuje odkazy na feature vektory bodů a také procentuální důvěru v to, jak moc jsou si dané body podobné, tzv. *similarity*. Metoda *Match()* má ještě přetížení s parametrem *threshold*, který udává procento párů, které budou vráceny (řazeno od nejlepší shody).

Nalezené páry přebírá metoda *GetTransformation()* rozhraní *ITransformer*, která na základě vstupních dat hledá transformaci mezi body v páru a transformaci pro každý pár ukládá do pole transformací. Toto pole poté předá již zmiňované třídě *Density*, která najde v prostoru transformací hustotní maximum a tuto transformaci prohlásí za optimální a vrátí ji jako výstup celého algoritmu.

## 4.2 Implementace

V následující kapitole budou popsány jednotlivé části kódu a to podle blokového schématu (obrázek 4.1). Nejprve však budou popsány pomocné třídy, které jsou využívány ve více částech algoritmu:

- *Configuration* - načítá konfigurační soubor a ukládá ho do proměnné, kterou lze předat jiným třídám či metodám
- *FeatureVector* - uchovává bod, pět čísel popisujících jeho okolí (tzv. znaky) a průměr koule, na základě které je popis okolí vypočítán; také

implementuje metody na výpočet své velikosti a vzdálenosti k jinému feature vektoru, výchozí hodnota feature vektoru je 0 pro bod  $[0, 0, 0]$

- *Match* - entita obsahující dva feature vektory a míru důvěry v jejich shodu, ta musí být předána z rozhraní *IMatcher* nebo je dána výchozím nastavením na hodnotu 0
- *PictureMaker* - přebírá dvourozměrné pole desetinných čísel a disponuje metodou, která z tohoto pole vytvoří bitmapu ve stupních šedi (hodnoty 0 - 255)
- *Point3D* - reprezentace bodu v prostoru, implementuje metody na rotaci a translaci bodu

Třídy, které nejsou uvedené v seznamu a nebudou rozebírány v následující části ale jsou implementovány v kódu, nejsou pro algoritmus stěžejní, byly v hlavním kódu nahrazeny jinými třídami, nebo byly určeny pouze pro dočasné testovací účely.

### 4.2.1 IData

V programu jsou dvě třídy implementující rozhraní *IData*. Jedná se o třídy *VolumetricData* a *TransData*, kdy první zmíněná načítá a zpracovává reálná data, zatímco druhá je pomocná pro testovací účely. Následuje popis načtení reálných dat.

Veškeré základní informace o CT nebo  $\mu$ CT datech jsou uloženy v textovém .mhd souboru, jehož příklad je demonstrován v následující ukázce (listing 4.1). Pro načítání souborů jsou důležité poslední čtyři údaje, tj. vzorkování ve všech třech směrech (*ElementSpacing*), dimenze ve všech třech směrech (*DimSize*), formát uložení hodnoty (*ElementType*) a soubor, ve kterém jsou uloženy hodnoty v binární podobě (*ElementDataFile*). *ElementType* nabývá hodnot MET\_USHORT (dva bajty) a MET\_UCHAR (jeden bajt).

---

```

ObjectType = Image
NDims = 3
BinaryData = True
BinaryDataByteOrderMSB = False
CompressedData = False
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = 0 0 0
CenterOfRotation = 0 0 0
AnatomicalOrientation = RAI
ElementSpacing = 0.053059 0.053059 0.053058
DimSize = 1012 1024 1014
ElementType = MET_USHORT
ElementDataFile =
    P01_a_MikroCT-nejhrubsi_rozliseni_DICOM_liver-1st-
    important_Macro_pixel-size53.0585um.raw

```

---

Listing 4.1: Ukázka .mhd souboru.

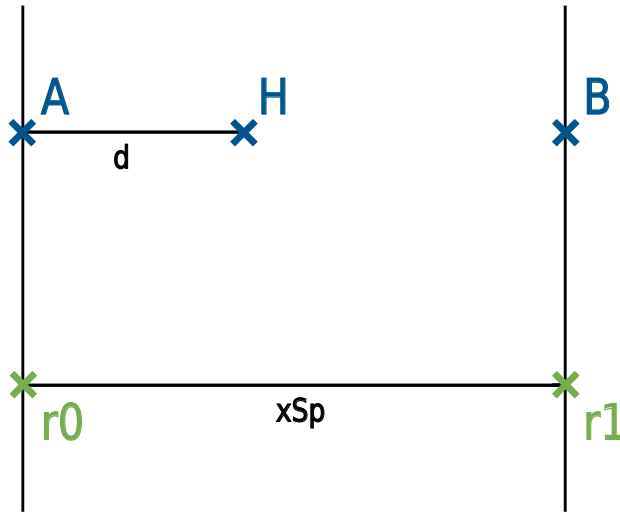
Tyto základní informace jsou uloženy v instanci pomocné třídy *Data*, jejímž jediným úkolem je načíst informace, uložit je a poskytnout je, pokud jsou vyžadovány. Druhou třídou pro zpracování dat je již zmiňovaná třída *VolumetricData*. Ta načítá binární data z .raw souboru a ukládá je do pole dvourozměrných polí (to je z důvodu paměťové náročnosti). Výsledkem je tedy kvádr denzit ( $VData[z][x,y]$ ), který je soukromou proměnnou třídy a je dále zpracován jejími metodami. Nejdůležitější metodou (je implementována z rozhraní) je metoda *GetValue()*, která využitím interpolace vrátí hodnotu denzity v libovolném reálném bodě. Interpolace v jedné dimenzi pro libovolný bod  $H$  a jeho hodnotu  $h$  je dána vztahem:

$$d = x_H - x_A, \quad (4.1)$$

$$r = d/xSp, \quad (4.2)$$

$$h = r \cdot b + (1 - r) \cdot a, \quad (4.3)$$

kde  $xSp$  je vzdálenost dvou vzorků v datové mřížce v dimenzi  $x$ ,  $x_H, x_A$  jsou souřadnice bodů  $H, A$  a  $a, b$  jsou hodnoty v bodech  $A, B$  (viz obrázek 4.3). V případě, že  $H = A$ , je  $r = 0$  a interpolační metoda vrátí správně hodnotu  $a$  (znázorňuje bod  $r0$ ) a totéž platí pro  $H = B$ , kdy  $r = 1$  a interpolace vrátí hodnotu  $b$  (znázorňuje bod  $r1$ ). Tímto postupem je následně vypočítána hodnota bodu v trojrozměrných datech přes všechny dimenze.



Obrázek 4.3: Na obrázku je graficky znázorněna interpolační metoda, která počítá denzitu v bodě  $H$  na základě denzit okolních bodů  $A$  a  $B$ . V případě označeném jako  $r_0$  je hodnota v bodě  $H$  rovna hodnotě v bodě  $A$  a to samé platí v případě  $r_1$  akorát pro bod  $B$ . Vzdálenost  $xSp$  značí vzdálenost dvou vzorků v datové mřížce voxelů.

Poslední velmi důležitou metodou je metoda  $Cut()$ , která dělá dvourozměrný výřez z dat o daných rozměrech z daného bodu a v daných směrech (vektory směrů jsou však ortogonalizovány). Metoda také přebírá nové vzorkování, které bude v obou směrech stejné.

Druhou zmiňovanou třídou přebírající rozhraní  $IData$  je třída  $TransData$ , která vyrobí pro jakoukoli instanci třídy  $VolumetricData$  mikro data definováním matice rotace, vektoru posunu a rozměrů nových dat. Tím, že je transformace známá, se usnadňuje testování, které tím pádem může být založeno na výpočtu odchylky vypočítané transformace od očekávané transformace. Matice rotace je vypočítána na základě definovaného úhlu rotace  $\theta$  a rotační osy  $u = (u_x, u_y, u_z)$ :

$$R = \begin{bmatrix} \cos \theta + u_x^2(1 - \cos \theta) & u_x u_y(1 - \cos \theta) - u_z \sin \theta & u_x u_z(1 - \cos \theta) + u_y \sin \theta \\ u_y u_x(1 - \cos \theta) + u_z \sin \theta & \cos \theta + u_y^2(1 - \cos \theta) & u_y u_z(1 - \cos \theta) - u_x \sin \theta \\ u_z u_x(1 - \cos \theta) - u_y \sin \theta & u_z u_y(1 - \cos \theta) + u_x \sin \theta & \cos \theta + u_z^2(1 - \cos \theta) \end{bmatrix} \quad (4.4)$$

Třída disponuje i metodami pro výpočet úhlu a osy jakékoli matice rotace, což může být použito pro testování. Pokud definujeme rotační matici následovně

$$R = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad (4.5)$$

poté rotační osa může být definována vzorcem:

$$u = \begin{bmatrix} h - f \\ c - g \\ d - b \end{bmatrix} \quad (4.6)$$

a rotační úhel vzorcem:

$$|\theta| = \arccos\left(\frac{\text{Tr}(R) - 1}{2}\right) \quad (4.7)$$

kde  $\text{Tr}$  (*trace*) je součet prvků na diagonále dané matice.

Z důvodu, že jedna a ta samá rotace může být popsána různými rotačními maticemi, implementuje třída metodu *GetAlpha()*, která nejprve vypočítá rotaci z předpokládané rotace do vypočítané rotace a teprve z výsledné určí rotační úhel, který udává odchylku vypočítané rotace od předpokládané rotace. Dále má třída vlastní metodu *Sample()*, která má stejný účel jako stejnojmenná metoda rozhraní *ISample*. Její výhodou je, že navzorkuje body v uměle vytvořených mikro datech na stejná místa, na kterých jsou body metodě předané umístěné v makro datech. Podobnou metodou je metoda *SampleShifted()*, která navzorkované body posune o náhodnou velikost po všech osách, aby se tak lépe modelovala reálná situace, kdy body nebudou totožné, ale pokud možno velmi blízké.

## 4.2.2 ISampler

Rozhraní *ISampler* implementuje pouze třída *Sampler*, která navzorkuje body náhodně v celých datech. Tento přístup je ovšem neefektivní a pro zvýšení šance na to, že se body v mikro i makro datech navzorkují blízko sebe, je nutné zvýšit počet bodů, čímž se zvyšuje časová náročnost algoritmu, až je v reálném čase neupočitatelný. Výsledky experimentu, který na toto poukazuje jsou ukázány v následující kapitole.

Možným vylepšením je využití programu na hledání cévního stromu, který reprezentuje cévu jako úsečku s dvěma hraničními body (uzly) a poloměrem. Právě tyto uzly by se daly využít jako vlíčovací body a mohly by vykazovat lepší výsledky, než časově náročná metoda třídy *Sampler*. Naznačené řešení však není součástí této práce a jeho rozvedení může být využito pro další vývoj programu.

## 4.2.3 IFeatureComputer

V programu jsou dvě třídy implementující rozhraní *IFeatureComputer*. Jedna definuje metodu *GetSphere()* a druhá metodu *GetRing()*. Metoda *GetSphere()* najde daný počet bodů v kulovém okolí zadaného bodu, definovaném požadovaným poloměrem. Oproti tomu metoda *GetRing()* náhodně najde daný počet bodů v kulové slupce určené dvěma poloměry kolem zadaného bodu. Jak již bylo zmíněno, metoda *ComputeFeatureVector()*, kterou každá ze tříd musí implementovat, vrací instanci třídy *FeatureVector*, která přebírá bod a pět znaků popisujících jeho okolí.

Třída *FeatureComputer* počítá znaky přes soustředné koule o konstantně se zvyšujícím poloměru. Každý znak je poté průměr denzit bodů generovaných metodou *GetSphere()* v dané kouli. Počet generovaných bodů se se zvětšujícím se poloměrem také konstantně zvyšuje. Výsledný vektor je normovaný na jednotkový vektor.

Jiný přístup je zvolen ve třídě *FeatureComputerNormedRings*, která namísto soustředných koulí využívá soustředná mezikulí, aby tak lépe mohl vyniknout průměr každého jednotlivého okolí.



#### 4.2.4 IMatcher

Třída *Matcher*, implementující rozhraní *Matcher*, využívá pomocné třídy *KDTree* k nalezení nejbližších vektorů v prostoru feature vektorů. Třída *KDTree* je reprezentace stromu pro ukládání k-dimenzionálních dat. Jeho předností je rychlost vyhledávání, která je zapříčiněna tím, že algoritmus nemusí procházet všechny prvky. K-dimenzionální strom je konstruován tak, že vždy pro jednu dimenzi je prostor rozdělen na dvě části na základě mediánu a tento medián je přidán do stromu jako list. Další úroveň listů je vypočítána stejným způsobem, ale pro následující dimenzi. Takto algoritmus pokračuje, dokud všechny předané prvky netvoří strom. Jakmile je strom vytvořen, lze v něm jednoduchým a rychlým způsobem vyhledávat vektory nejbližší zadanému vektoru. Algoritmus *KDTree* byl převzat a upraven od [20].

Po vytvoření instance třídy *KDTree* najde metoda *Match()* pro každý feature vektor vypočítaný pro mikro data nejbližší vektor z pole feature vektorů vypočítaných pro makro data. Následně je pro všechny sdružené páry vektorů vypočítána jejich vzdálenost a tou je vyjádřena procentuální míra důvěry v to, že body odpovídající vektorům jsou tytéž (nebo se alespoň nacházejí dosti blízko sebe) pouze navzorkované v jiných datech.

#### 4.2.5 ITransformer

Posledním blokem algoritmu, kterému se ještě tato práce věnuje, je třída *Transformer3D* rozhraní *ITransformer*, která najde transformaci pro zadanou instanci třídy *Match*. Důležité je, že transformace není komutativní operace a v tomto algoritmu jsou transformována mikro data na makro data, nikoli naopak. Podobně jako rozšířená metoda fázové korelace (zmíněná v kapitole 3.2.2) i metoda *GetTransformation* nejprve vypočítá rotaci (tu uchovává v podobě rotační matice nikoli rotačního úhlu), poté zrotuje bod z mikro dat a vypočítá posun mezi zrotovaným bodem a jeho odpovídajícím protějškem v makro datech.

Pro výpočet rotace je vytvořena třída *RotationComputer*, která přebírá odkazy na instance mikro i makro dat a daný pár bodů, pro který je rotace počítána. Třída byla implementována ve spolupráci s doc. Ing. Liborem Vášou, Ph.D. Algoritmus výpočtu rotace se skládá ze dvou částí. Nejprve je pro oba body (mikro i makro) vypočítána symetrická matice popisující bázi prostoru v okolí daného bodu a z té jsou metodou *Evd()* získány vlastní vektory, které jsou díky symetrii matic ortogonální.

S pomocí vlastních vektorů obou matic je následně vypočítána přechodová matice, která reprezentuje rotační matici, neboli rotaci z jedné báze do druhé.

Symetrická matice pro daný bod je vypočítána na základě velkého množství bodů (zadáno jako parametr metody, řádově desítky tisíc bodů) navzorkovaného metodou *GetSphere()*. V tomto shluku bodů je nalezena báze prostoru tak, že jedna souřadná osa je dána směrem největších denzit, druhá souřadná osa je kolmá na první a je dána směrem nejmenších denzit a třetí souřadná osa je kolmá na první dvě. Směr souřadných os je generován náhodně, ale tak, aby výsledná báze byla pravotočivá. V praxi to probíhá tak, že daný shluk je vycentrován kolem počátku a je vypočítána autokorelační matice přes všechny body. Z té jsou spočítány vlastní vektory a tyto vektory tvoří výše popsanou bázi.

Báze mikro dat je tedy dána maticí vlastních vektorů  $V'$  definovanou následovně:

$$V' = \begin{bmatrix} v'_1 & v'_2 & v'_3 \end{bmatrix} \quad (4.8)$$

a báze makro dat je definována podobně maticí  $V$ :

$$V = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}, \quad (4.9)$$

poté, vzhledem k tomu že jsou obě matice unitární (řádkové vektory matice jsou ve smyslu skalárního součinu ortonormální), platí následující úprava rovnice s požadovanou maticí rotace  $R$ :

$$\begin{aligned} V &= R \cdot V' \\ V \cdot V'^{-1} &= R \\ R &= V \cdot V'^T \end{aligned}$$

Pokud jsou v datech nalezeny blízké body, *Matcher* na základě vektorů poskytnutých třídou *FeatureComputer* správně přiřadí odpovídající si body a *RotationComputer* vypočítá rotaci s malou odchylkou vůči reálné rotaci, bude mít i translace malou odchylku od té reálné. Testy jednotlivých částí algoritmu jsou uvedeny v následující kapitole.

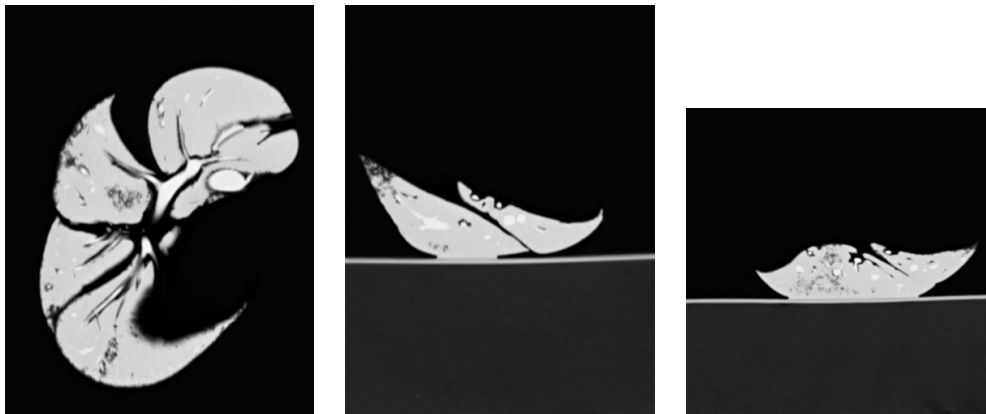
# 5 Testování vybraných částí algoritmu

Vzhledem ke komplexnosti navrženého algoritmu a také nemožnosti určit, zda je transformace vypočítaná na reálných datech správná, je nutné testování navrženého řešení provést na uměle vytvořených datech, resp. z části uměle vytvořených. Jak již bylo zmíněno dříve, pro testování algoritmu existuje třída *TransData*, která simuluje mikro data pro jakákoli reálná data.

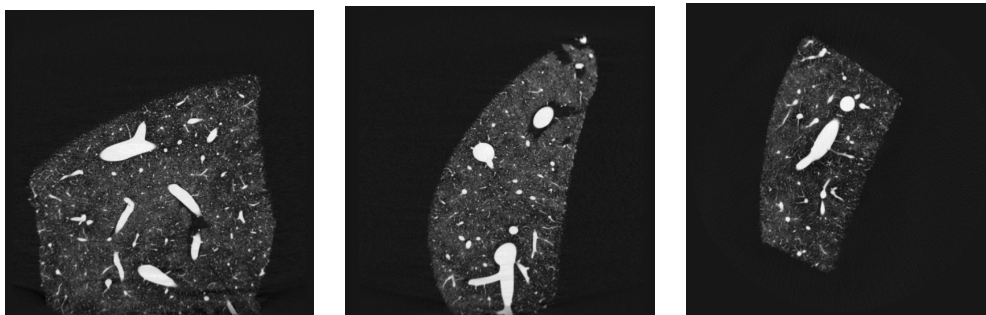
Třída *Sampler* zjevně nemusí být testována, protože sama o sobě pouze generuje náhodné souřadnice bodů. Taktéž *Matcher* nebude testován, protože se jedná o implementaci již známého a tedy testovaného algoritmu. V této kapitole bude ukázána funkčnost třídy *VolumetricData* a jejích metod *GetValue()* a *Cut()*. Dále budou diskutovány třídy počítající feature vektory (*FeatureComputer* a *FeatureComputerNormedRings*) a následně bude uvedena a rozebrána série testů pro třídu *Transformer3D* a její metodu *GetTransformation()*.

## 5.1 VolumetricData

Funkčnost metod *GetValue()* a *Cut()* byla verifikována vytvořením několika řezů daty, které byly následně vizuálně ověřeny. Na obrázcích 5.1 a 5.2 jsou ukázky řezů makro a mikro daty pořízených metodou *Cut()*.



Obrázek 5.1: Na obrázcích jsou řezy makro daty provedené základními rovinami přibližně uprostřed dat. Roviny řezů jsou následující: (zleva) rovina daná vektory rovnoběžnými s osami  $y$  a  $z$  a zvoleným bodem  $X$ , rovina daná vektory rovnoběžnými s osami  $x$  a  $z$  a zvoleným bodem  $Y$ , rovina daná vektory rovnoběžnými s osami  $x$  a  $y$  a zvoleným bodem  $Z$ .



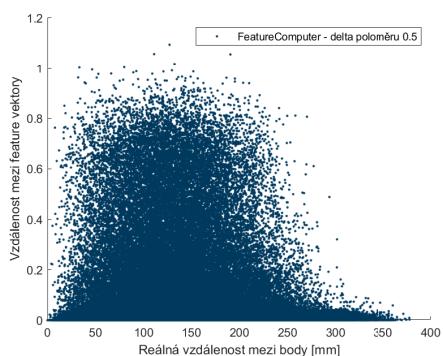
Obrázek 5.2: Na obrázcích jsou řezy mikro daty provedené základními rovinami přibližně uprostřed dat. Roviny řezů jsou následující: (zleva) rovina daná vektory rovnoběžnými s osami  $y$  a  $z$  a zvoleným bodem  $X$ , rovina daná vektory rovnoběžnými s osami  $x$  a  $z$  a zvoleným bodem  $Y$ , rovina daná vektory rovnoběžnými s osami  $x$  a  $y$  a zvoleným bodem  $Z$ .

## 5.2 IFeatureComputer

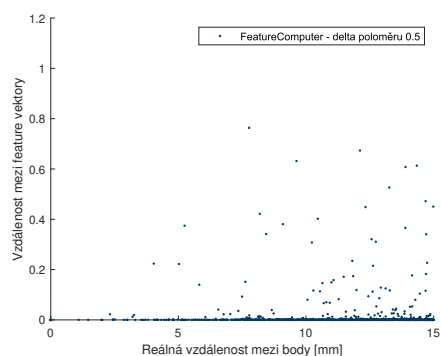
Pro testování funkčnosti dvou tříd implementujících rozhraní *IFeatureComputer* (*FeatureComputer* a *FeatureComputerNormedRings* - na obrázku značena pouze jako *NormedRings*) byl zvolen test závislosti vzdálenosti dvou feature vektorů na vzdálenosti bodů, ke kterým byly přidruženy. Vychází z předpokladu, že velmi blízké body by si měly být hodně podobné a tedy jejich feature vektory by si měli být blízké. Ideální stav by tedy byla zřejmě přímka  $y = x$ , na druhou stranu je nutné si uvědomit, že vzdálené body mohou mít také podobné okolí. Tedy ideální stav by byl shluk bodů, kde pro malé  $x$  (např. 0 – 10) je  $y$  rovno 0 a pro větší  $x$  je  $y$  větší než 0. Důležitější je aby pro blízké body byla vzdálenost feature vektorů malá (téměř 0), než aby vzdálené body neměly vzdálenost feature vektorů malou. Proto se experiment zaměřuje na okolí blízkých bodů a nezabývá se body, jejichž feature vektory jsou blízké.

Na obrázku 5.3 jsou graficky znázorněny výsledky jednotlivých experimentů pro jeden milion náhodných párů, které byly generovány v jedné a těch samých datech. Vzhledem k tomu, že výsledek je ovlivněn zvolenými poloměry soustředných koulí/mezikulí, byly pro každou třídu provedeny dva experimenty, každý s jinými poloměry. Poloměr (ať už se jedná o kouli nebo mezikulí) je roven proměnné  $\delta$  pro první znak a následně je pro každý další znak zvětšen o  $\delta$ . V případě mezikulí je druhý poloměr (ten vnitřní) dán nulou pro první znak a následně opět zvětšován o  $\delta$  pro každý další. Pro každou třídu byl proveden experiment s  $\delta = 0.5$  a druhý s  $\delta = 1$ . Na obrázcích vpravo (tj. 5.3b, 5.3d, 5.3f, 5.3h) je přiblížení grafu v oblasti zájmu, tedy v oblasti, kde jsou si body blízké.

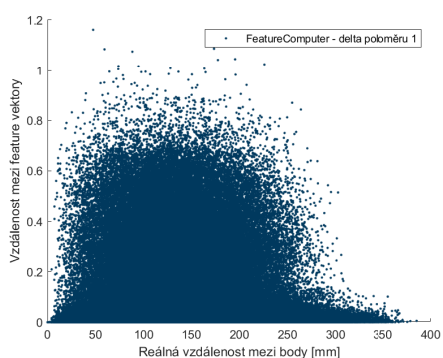
Na obrázku je vidět, že vyšší  $\delta$  má lepší výsledky pro obě třídy. To je zřejmě způsobeno tím, že je vzato v potaz větší okolí bodu. Pro moc vysoké  $\delta$  by však bylo potřeba zjistit denzitu více bodů, aby byla dostatečně pokryta daná oblast, a byla by tak zvýšena časová náročnost algoritmu. Z obou tříd se jeví jako výhodnější třída *FeatureComputerNormedRings*, která, jak již bylo zmíněno v kapitole 4.2.3, lépe popisuje jednotlivá okolí bodu.



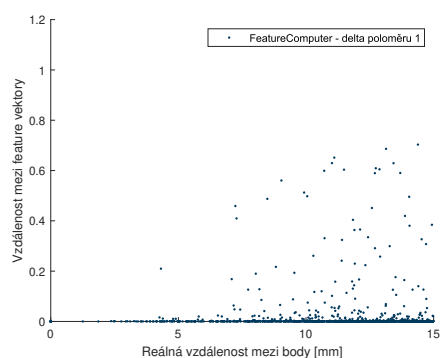
(a) *FeatureComputer* -  $\delta = 0.5$



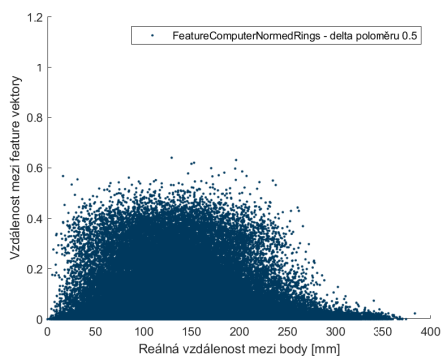
(b) *FeatureComputer* -  $\delta = 0.5$



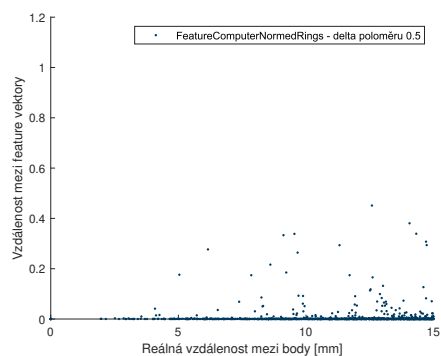
(c) *FeatureComputer* -  $\delta = 1$



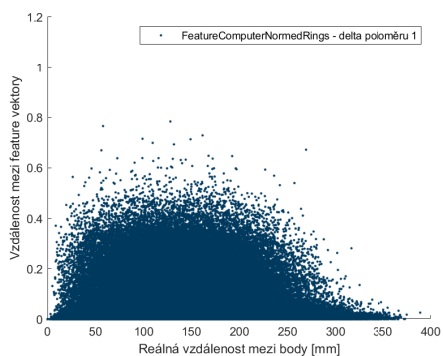
(d) *FeatureComputer* -  $\delta = 1$



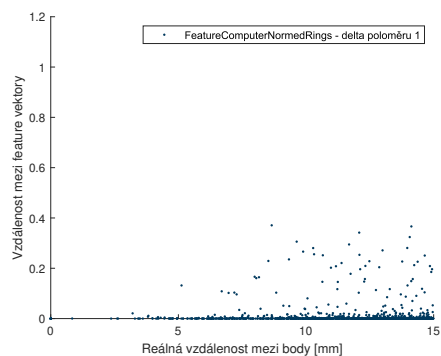
(e) *NormedRings* -  $\delta = 0.5$



(f) *NormedRings* -  $\delta = 0.5$



(g) *NormedRings* -  $\delta = 1$



(h) *NormedRings* -  $\delta = 1$

Obrázek 5.3: Srovnání funkčnosti algoritmů pro výpočet feature vektorů. Test byl proveden pro jeden milion náhodných párů bodů v jedné a těch samých datech. Vpravo je vždy přiblížení levého grafu na ose x na interval 0 – 15.

## 5.3 Transformer3D

Pro testování funkčnosti výpočtu rotace byly zvoleny dva různé testy. Ve všech jsou počítány odchylky všech vypočítaných rotací od předpokládané (zadané) rotace. V grafech je poté v procentech znázorněno kolik odchylek je menších než odchylka na ose  $x$ . Ideálním stavem by tedy byl skok na 100% v bodě  $[0.1, 0]$  (procentuální zastoupení je počítáno s krokem  $0.1^\circ$ ).

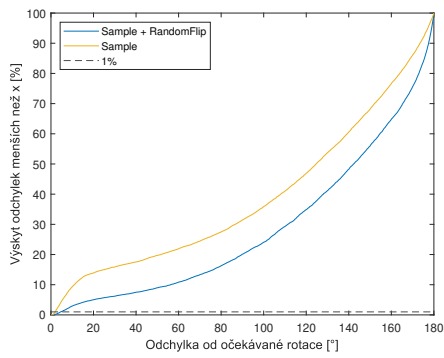
Bylo zjištěno, že se zvyšujícím se rotačním úhlem klesá zastoupení malých odchylek ( $0^\circ$ ) na úkor velkých ( $180^\circ$ ). Jak již bylo zmíněno, při výpočtu rotace jsou nalezeny báze okolí obou bodů, směry jednotlivých vektorů jsou však zvoleny metodou a to tak, aby byla výsledná báze pravotočivá. Proto byla navržena metoda *RandomFlipBasis()*, která náhodně otočí směry vektorů báze se zachováním pravotočivosti. Předpoklad byl, že vlivem náhody, se rozloží zastoupení nízkých a vysokých odchylek, tedy by měla křivka znázorněná v grafech být pro různé rotace stejná. Oproti původní křivce bylo očekáváno zhoršení u nízkých rotací a zlepšení u rotací vyšších. V prvním experimentu (viz obrázek 5.4) jsou tedy dvě křivky pro každou zvolenou rotaci - jedna (žlutá) znázorňuje výsledky hledání rotace algoritmem bez metody *RandomFlipBasis()* a druhá (modrá) značí výsledky hledání rotace se zmíněnou metodou. Jak je vidět již u rotace o  $120^\circ$ , metoda *RandomFlipBasis()* splňuje předpoklad vylepšení výsledků pro vyšší rotace. Experiment byl proveden s využitím třídy *TransData* a její metodou *Sample()*, která generuje body v mikro i makro datech na stejných místech vůči sobě.

Důležitým faktem je, že *Density* hledá v prostoru transformací, tj. rotací včetně k nim dopočítaných translací, tím pádem rotace s malými odchylkami budou generovat podobné translace, zatímco špatné rotace budou generovat různé translace, čímž se vytvoří chtěný hustotní vrchol v prostoru transformací v okolí požadované transformace. Díky tomu stačí najít málo rotací (v obrázcích naznačeno přímkou 1%) s malou odchylkou a *Density* přesto najde správnou transformaci. V tabulce 5.1 jsou zaznamenány výsledné odchylky rotací vypočítaných v jednotlivých částech experimentu.

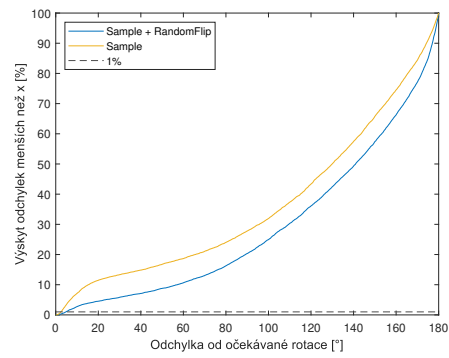
	$0^\circ$	$30^\circ$	$60^\circ$	$90^\circ$	$120^\circ$	$180^\circ$
Bez <i>RandomFlipBasis()</i>	0.11	0.47	1.00	2.44	3.45	2.75
S <i>RandomFlipBasis()</i>	0.58	0.86	2.48	3.29	1.64	3.47

Tabulka 5.1: Srovnání výsledných odchylek pro test bez a s využitím metody *RandomFlipBasis()* pro uvedené rotace. Výsledky jsou ve  $^\circ$  zaokrouhleny na dvě desetinná místa.

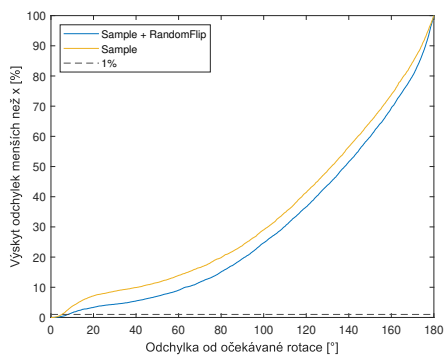




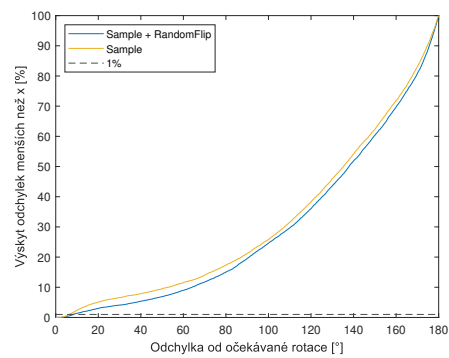
(a) Rotace o  $0^\circ$



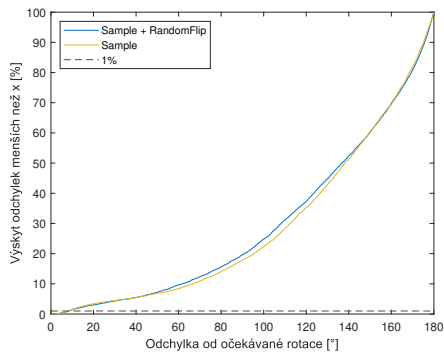
(b) Rotace o  $30^\circ$



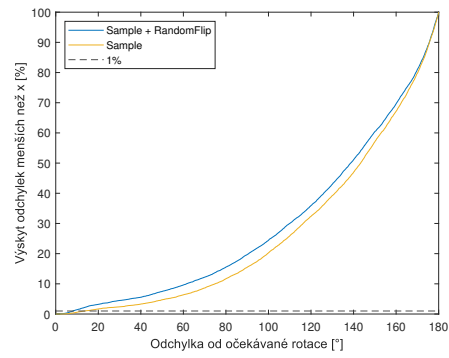
(c) Rotace o  $60^\circ$



(d) Rotace o  $90^\circ$



(e) Rotace o  $120^\circ$



(f) Rotace o  $180^\circ$

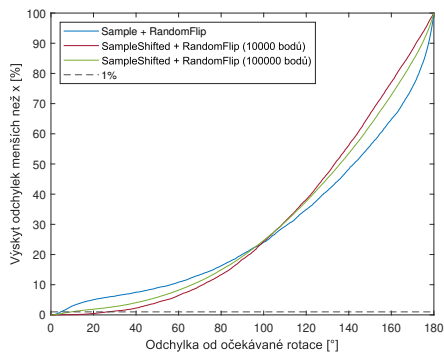
Obrázek 5.4: Vliv náhodné volby směru rotace na odchylku výsledné rotace od očekávané rotace.

Ve druhém experimentu je snahou simulovat ještě více reálné podmínky a to použitím metody *SampleShifted()*. Jak již bylo zmíněno, ta na rozdíl od metody *Sample()* body po jejich vygenerování náhodně posune. Na obrázku 5.5 je zobrazen experiment pro stejné rotace, jako experiment předchozí. Srovnává test s použitím metod *Sample()* a *RandomFlipBasis()* (modrá křivka) se dvěma testy s použitím metod *SampleShifted()* a *RandomFlipBasis()* (červená a zelená křivka). Druhá křivka (červená) odpovídá testu pro 10000 generovaných bodů, zatímco třetí (zelená) odpovídá testu pro 100000 generovaných bodů. Je vidět velké zhoršení druhých křivek oproti první křivce, ale to se dá vzhledem k posunu předpokládat. Na druhou stranu je vidět mírné zlepšení třetí křivky vůči druhé, což naznačuje, že zvýšení počtu generovaných bodů může mít nemalý vliv na výslednou rotaci. Výsledné odchylky rotací jsou uvedeny v tabulce 5.2.

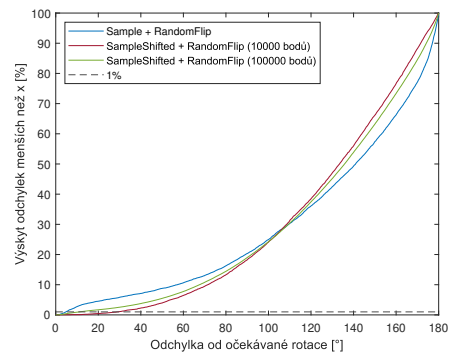
	0°	30°	60°	90°	120°	180°
<i>Sample()</i>	0.58	0.86	2.48	3.29	1.64	3.47
<i>SampleShifted()</i> (10000 bodů)	179.41	4.72	139.87	177.03	154.77	48.65
<i>SampleShifted()</i> (100000 bodů)	0.47	0.34	0.67	2.41	1.08	2.24

Tabulka 5.2: Srovnání výsledných odchylek pro testy s využitím metod *Sample()* a *SampleShifted()* pro uvedené rotace. Všechny experimenty též používají metodu *RandomFlipBasis()*. Výsledky jsou ve ° zaokrouhleny na dvě desetinná místa.

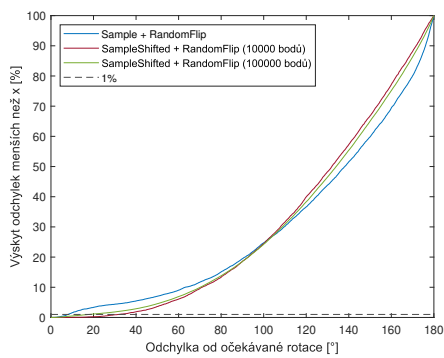
Jak je vidět v tabulce 5.2, počet generovaných bodů je velmi důležitý, neboť v tomto případě má test s metodou *SampleShifted()* lepší výsledky odchylek než test s metodou *Sample()*. Toto také potvrzuje předpoklad, že stačí málo relativně správných rotací a třída *Density* najde také relativně správnou výslednou transformaci.



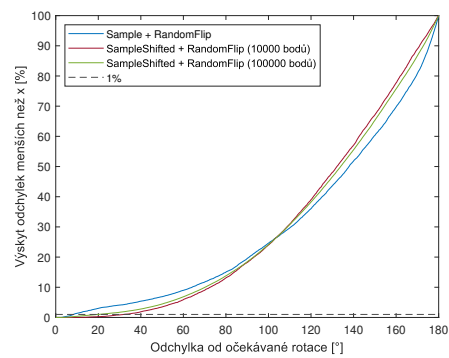
(a) Rotace o  $0^\circ$



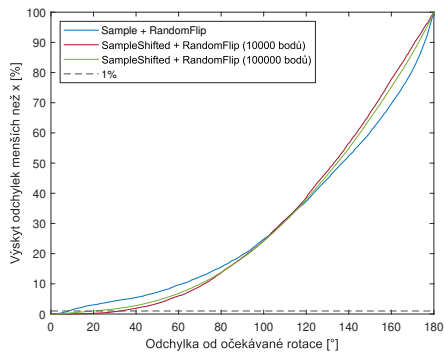
(b) Rotace o  $30^\circ$



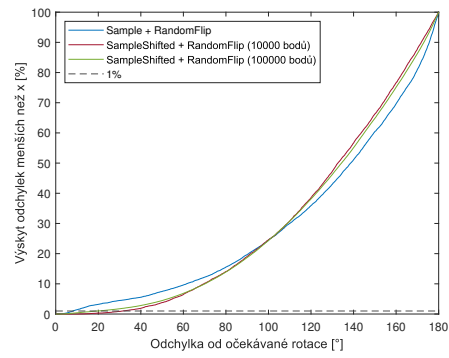
(c) Rotace o  $60^\circ$



(d) Rotace o  $90^\circ$



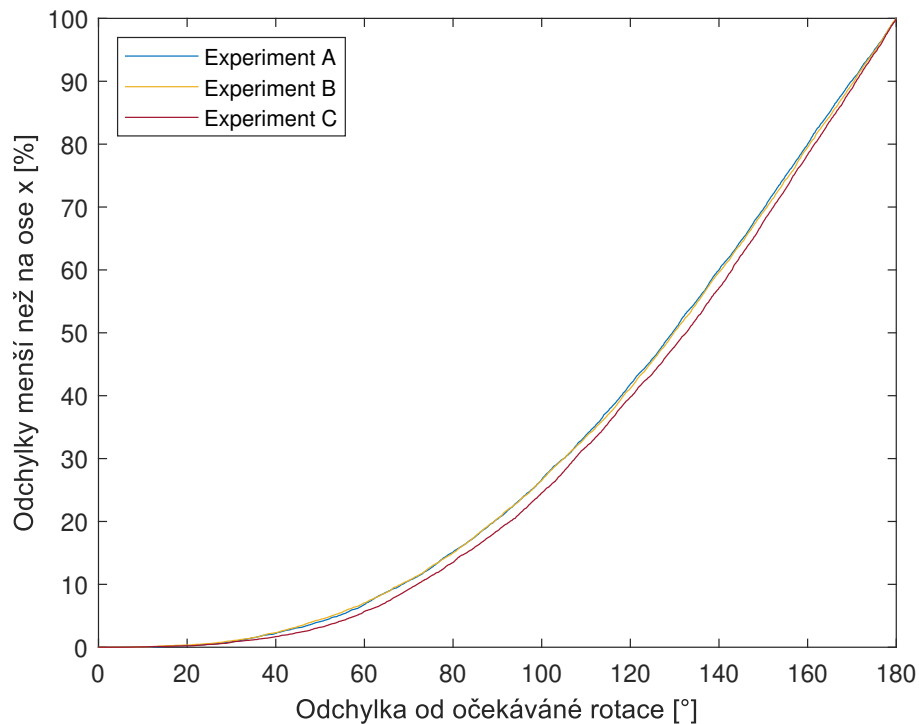
(e) Rotace o  $120^\circ$



(f) Rotace o  $180^\circ$

Obrázek 5.5: Srovnání metod *Sample* a *SampleShifted*.

Na obrázku 5.6 je ukázán experiment, který poukazuje na to, že pro fungování algoritmu je nutné implementovat novou třídu, která by nahradila momentálně používanou třídu *Sampler*. Je vidět, že pro více generovaných bodů se výsledek částečně zlepšuje, ale vzhledem k rostoucí časové náročnosti nemohla být tato teorie ověřena pro více bodů.



Obrázek 5.6: Experimenty s použitím třídy *Sampler*. Experimenty byly provedeny pro následující počet bodů generovaných v mikro a makro datech: jeden milion bodů v mikro datech a dva miliony bodů v makro datech (experiment A, modrá křivka), jeden milion bodů v mikro datech a pět milionů bodů v makro datech (experiment B, žlutá křivka), čtyři miliony bodů v mikro datech a osm milionů bodů v makro datech (experiment C, červená křivka).

## 6 Závěr

Registrace volumetrických dat získaných  $\mu$ CT a CT přístroji je důležitá, protože umožňuje doplnění chybějících informací v CT snímcích mnohonásobně detailnějšími snímky z  $\mu$ CT. Tato práce přibližuje teorii z oblasti biologie, týkající se fungování jater a problematiku získávání modelu cévní struktury jater prasete, který byl použit k pořízení dat. V textu byly popsány, pro tuto práci stěžejní, zobrazovací metody, kterými se volumetrická data pořizují - CT a  $\mu$ CT. Práce se také zabývala problematikou registrace, jejím dělením a registračními metodami používanými v praxi.

Vlastní práce byla zaměřena na návrh algoritmu pro registraci dat a implementaci jeho hlavního rozhraní. Algoritmus je dostupný na platformě GitHub [21]. Cílem bylo ověřit schopnost daného algoritmu nalézt výslednou transformaci a objevit jeho nejproblematictější části. Algoritmus je tvořen pěti bloky. Práce byla soustředěna zejména na blok pro výpočet feature vektorů a blok počítající rotaci. Výsledky jednotlivých experimentů ukazují na dobrou volbu metody pro výpočet feature vektorů a na funkční výpočet rotace na základě dvou bodů, každého z jedněch poskytnutých dat. Experimentálně bylo ukázáno, že nejslabším článkem algoritmu je třída *Sampler*, která byla napsána pouze jako náhodný generátor souřadnic a tato práce se na ni nezaměřila.

Logickým rozšířením práce je zdokonalení implementace zbývajících bloků. Možným vylepšením bloku pro generování bodů (momentálně třída *Sampler*) by bylo využití vlíčovacích bodů získaných jinými, již implementovanými algoritmy, či napsání nového algoritmu přímo pro tento program.

# Literatura

- [1] ČIHÁK, R. *Anatomie 2, Druhé, upravené a doplněné vydání*. Grada Publishing, a.s., 2002. ISBN 978-80-247-0143-1.
- [2] FONTANA, J. – ŠAJDÍKOVÁ, M. – MAĎA, P. 4. *Játra a biotransformace xenobiotik* [online]. Funkce buněk a lidského těla (multimediální skripta), 2016-2018. [cit. 2020/06/28]. Dostupné z: <http://fbt.cz/skripta/ix-travici-soustava/5-jatra-a-biotransformace-xenobiotik/>.
- [3] *Játra (pohled zezadu)* [online]. <https://latinsky.estranky.cz/>, 2010. [cit. 2020/06/28]. Dostupné z: <https://latinsky.estranky.cz/fotoalbum/vylucovací-soustava/vylucovací-soustava/jatra--pohled-zezadu-.png-.html>.
- [4] UHER, V. Vizualizace volumetrických dat. Diplomová práce, VŠB - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2013.
- [5] LYSOŇ, J. Moderní zobrazování volumetrických dat v reálném čase. Diplomová práce, VŠB - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky, 2016.
- [6] JIŘÍK, M. et al. Stereological quantification of microvessels using semiautomated evaluation of X-ray microtomography of hepatic vascular corrosion casts. *International Journal of Computer Assisted Radiology and Surgery*. 2016, 11, 10, s. 1803–1819. ISSN 18616429. doi: 10.1007/s11548-016-1378-3.
- [7] EBERLOVA, L. et al. The use of porcine corrosion casts for teaching human anatomy. *Annals of Anatomy-Anatomischer Anzeiger*. 2017, 213, s. 69–77.
- [8] NEKULA, J. et al. *Radiologie*. Univerzita Palackého, 2003. ISBN 80-244-0672-1.
- [9] HEŘMAN, M. *Základy radiologie*. Univerzita Palackého, 2014. ISBN 978-80-244-2901-4.
- [10] BUDIŠ, M. Srovnání možností uplatnění rentgenové mikrotomografie v metrologii s konvenčními přístupy. Diplomová práce, Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013.
- [11] GAJDOŠÍK, R. Charakterizace porézních materiálů pomocí RTG mikrotomografie. Bakalářská práce, Univerzita Tomáše Bati ve Zlíně, Fakulta technologická, 2014.

- [12] JIRKŮ, K. *Kardiovaskulární soustava Apparatus angiologia (prezentace)* [online]. Jihočeská univerzita, Přírodovědná fakulta. [cit. 2020/07/07]. Dostupné z: [https://www.prf.jcu.cz/data/files/766/767/773/4823fa\\_kardiovaskula%CC%81rni%CC%81\\_soustava.pdf](https://www.prf.jcu.cz/data/files/766/767/773/4823fa_kardiovaskula%CC%81rni%CC%81_soustava.pdf).
- [13] GOTTESFELD BROWN, L. A Survey of Image Registration Techniques. *ACM Computing Surveys*. 1992, 24, 4, s. 326–376. doi: 10.1007/0-306-48608-3\_13.
- [14] MAINTZ, J. B. – VIERGEVER, M. A. A survey of medical image registration. *Medical Image Analysis*. 1998, 2, 1, s. 1–36. ISSN 13618415. doi: 10.1016/S1361-8415(01)80026-8.
- [15] ANTOŠOVSKÝ, J. Registrace obrazu. Bakalářská práce, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, 2013.
- [16] DLUHOŠ, P. Metaheuristické optimalizační metody pro registraci obrazů z magnetické rezonance. Bakalářská práce, Masarykova univerzita, Přírodovědná fakulta, 2011.
- [17] BULKA, M. Fúze výsledků segmentace multimodálních obrazových dat. Bakalářská práce, Západočeská univerzita v Plzni, Fakulta aplikovaných věd, 2016.
- [18] TŮMA, T. *Segmentace obrazu v CT (prezentace)* [online]. Karlova univerzita, Skupina počítačové grafiky. [cit. 2020/07/07]. Dostupné z: <https://cgg.mff.cuni.cz/~pepca/ref/angiografie.pdf>.
- [19] HRUDA, L. – DVOŘÁK, J. – VÁŠA, L. On evaluating consensus in RANSAC surface registration. *Computer Graphics Forum*. 2019, 38, 5, s. 175–186. ISSN 14678659. doi: 10.1111/cgf.13798.
- [20] VÁŠA, L. KD-Tree [algoritmus]. *Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky*. 2020.
- [21] VÍTOVÁ, N. – VÁŠA, L. – ŠMÍD, M. *Registration of volumetric data* [online]. GitHub, 2020. [cit. 2020/07/07]. Dostupné z: [https://github.com/tadySnek/registration\\_of\\_volumetric\\_data](https://github.com/tadySnek/registration_of_volumetric_data).