



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

REGISTRACE OBRAZŮ S VYUŽITÍM OPTIMALIZACE POMOCÍ AUTOMATICKÉ DIFERENCIACE

IMAGE REGISTRATION USING OPTIMIZATION WITH AUTOMATIC DIFFERENTIATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. David Slavíček

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Vičar, Ph.D.

BRNO 2023

Diplomová práce

magisterský navazující studijní program **Biomedicínské inženýrství a bioinformatika**

Ústav biomedicínského inženýrství

Student: Bc. David Slaviček

ID: 211212

Ročník: 2

Akademický rok: 2022/23

NÁZEV TÉMATU:

Registrace obrazů s využitím optimalizace pomocí automatické diferenciaci

POKyny PRO VYPRACOVÁNÍ:

1) Prostudujte metody pro automatickou registraci obrazových dat se zaměřením na optimalizaci transformačních parametrů pomocí metod využívajících gradient. 2) Proveďte rešerši existujících knihoven pro registraci obrazů pomocí metod využívajících gradient, které využívají automatickou diferenciaci knihoven pro hluboké učení (PyTorch, Tensorflow, atd.). 3) Navrhněte a implementujte vlastní jednoduchou pro afinní registraci s optimalizací využívající automatické diferenciaci a jeho funkčnost ověřte na několika synteticky deformovaných obrázcích. 4) Vytvořte vhodné databáze pro testování úspěšnosti registrace. 5) Zdokonalte a rozšiřte vytvořenou metodu pro registraci a jednotlivé modifikace vyhodnoťte na vytvořených databázích. 6) Výsledky vhodně diskutujte a srovnajte s dostupnými metodami registrace.

DOPORUČENÁ LITERATURA:

[1] ZITOVÁ, Barbara a Jan FLUSSER. Image registration methods: a survey. Image and Vision Computing. 2003, 21(11), 977-1000. ISSN 02628856.

[2] MATTES, David, David R. HAYNOR, Hubert VESSELLE, Thomas K. LEWELLYN, William EUBANK, Milan SONKA a Kenneth M. HANSON. Nonrigid multimodality image registration. 2001-7-3, 1609-1620.

[3] NAN, Abhishek, Matthew TENNANT, Uriel RUBIN a Nilanjan RAY. DRMIME: Differentiable Mutual Information and Matrix Exponential for Multi-Resolution Image Registration. In: Proceedings of Machine Learning Research. PMLR, 2020, s. 527-543. ISSN 2640-3498.

Termín zadání: 6.2.2023

Termín odevzdání: 14.8.2023

Vedoucí práce: Ing. Tomáš Vičar, Ph.D.

prof. Ing. Valentine Provazník, Ph.D.

předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá využitím automatické diferenciaci v registraci obrazů. V teoretické části byly popsány kroky registrace obrazů a možné způsoby jejich realizace. Dále byly popsány různé algoritmy sloužící k registraci medicínských obrazů. V praktické části byl navržen algoritmus pro registraci afinně transformovaných obrazů s využitím automatické diferenciaci. Algoritmus byl implementován. Byly navrhnuty dvě modifikace algoritmu. Dále byla sestavena testovací databáze sestávající z RTG snímků hrudníku, na které byl algoritmus otestován. Další testování proběhlo na částech databáze snímků sítnice FIRE. Na částech této databáze byl také algoritmus srovnán s jinými algoritmy.

KLÍČOVÁ SLOVA

registrace obrazů, automatická diferenciaci, optimalizace, zpracování obrazu

ABSTRACT

This thesis deals with image registration using optimization with automatic differentiation. In the teoretical part were decribed steps of image registration and possibilities of their realization. Other algorithms for medical image registration were also described. The practical part consists of design of algorithm for registration of afinely transformed images and implementation of it. Two modifications of the algorith were proposed. Database of xray images was designed and the algorithm was tested on this database. Another testing database was FIRE database of retinal images. Algorithm was compared to other existing registration algorithms using parts of the FIRE database.

KEYWORDS

image registration, automatic differentiation, optimalization, image processing

SLAVÍČEK, David. *Registrace obrazů s využitím optimalizace pomocí automatické diferenciace*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2023, 62 s. Diplomová práce. Vedoucí práce: Ing. Tomáš Vičar, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. David Slaviček
VUT ID autora:	211212
Typ práce:	Diplomová práce
Akademický rok:	2022/23
Téma závěrečné práce:	Registrace obrazů s využitím optimalizace pomocí automatické diferenciaci

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Tomáši Vičarovi Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Tato práce vznikla za podpory projektu e-INFRA CZ (ID:90254) podporovaného ministerstvem školství, mládeže a tělovýchovy České republiky

Obsah

Úvod	12
1 Registrace obrazů	13
1.1 Kriteriální funkce	14
1.1.1 Střední kvadratická chyba (MSE)	14
1.1.2 Střední absolutní chyba (MAE)	14
1.1.3 Vzájemná informace	14
1.1.4 Diferencovatelná vzájemná informace	15
1.1.5 Kovariance	15
1.1.6 Normalizovaná křížová korelace	15
1.2 Metody založené na ploše a metody založené na charakteristických rysech	16
1.3 Transformace	16
1.3.1 Afinní transformace	16
1.3.2 Transformace s využitím kontrolních bodů	16
1.4 Optimalizace	16
1.5 Algoritmy pro registraci obrazů	16
1.5.1 H-M	16
1.5.2 REMPE	18
1.5.3 Harris-PIIFD	19
1.5.4 GDB-ICP	21
1.5.5 DRMIME	21
1.5.6 Nerigidní multimodální registrace obrazů - Mattes a kolektiv	22
1.5.7 DIRNet	22
1.6 Software pro registraci obrazů	23
1.6.1 Elastix	23
1.6.2 Simple Elastix	23
1.6.3 Kornia	23
1.6.4 AIRLab	25
2 Navržený algoritmus	26
2.1 Popis algoritmu	26
2.2 Pyramidový přístup	26
2.3 Změna kroku učení pomocí scheduleru	26
2.4 Implementace navrženého algoritmu	27

3	Testovací databáze	29
3.1	Rentgenové snímky hrudníku	29
3.2	FIRE: Fundus Image Registration Dataset	30
3.2.1	Parametry snímků	30
3.2.2	Popis částí databáze	30
3.2.3	Doporučená metrika	31
4	Testování hyperparametrů na rentgenových snímcích	34
4.1	Metrika testování	34
4.2	Testování hyperparametrů základní verze algoritmu	34
4.2.1	Testování kroku učení v základní verzi algoritmu	34
4.2.2	Testování počtu iterací v základní verzi algoritmu	35
4.3	Testování hyperparametrů algoritmu modifikovaného s využitím pyramidového přístupu	36
4.3.1	Testování počtu úrovní pyramidy	36
4.3.2	Testování počtu iterací	36
4.3.3	Testování kroku učení	37
4.4	Testování hyperparametrů algoritmu modifikovaného s využitím pyramidového přístupu a scheduleru	38
4.5	Subjektivní zhodnocení výsledku	38
5	Testování algoritmů pro jednotlivé dílčí transformace na rentgenových snímcích	40
6	Testování algoritmu na databázi FIRE	42
6.1	Předzpracování	42
6.2	Optimalizace hyperparametrů	43
6.2.1	Provedení optimalizace hyperparametrů	44
6.2.2	Výsledky optimalizace hyperparametrů	45
6.2.3	Příklad registrace	45
6.3	Porovnání s jinými algoritmy	47
6.3.1	Část databáze A	47
6.3.2	Část databáze S	49
7	Diskuze	51
	Závěr	52
	Literatura	53
	Seznam symbolů a zkratk	57

A	Grafy testování algoritmu a jeho modifikací	58
B	Ukázka předzpracovaného snímku sítnice	62

Seznam obrázků

1.1	H-M 16 model oka	17
1.2	REMPE schéma	18
1.3	Harrisův detektor	19
1.4	PFIID okol9 a histogram	20
1.5	Kornia registrace	24
2.1	Schéma algoritmu	27
2.2	Schéma algoritmu	28
3.1	RTG snímek	29
3.2	FIRE část A ukázka	30
3.3	FIRE část P ukázka	31
3.4	FIRE část S ukázka	32
3.5	FIRE metrika AUC	32
4.1	Vizualizace registrace rentgenových snímků	39
5.1	Testování posunu v X	40
5.2	Testování zkosení	41
6.1	Rozdíl snímků	42
6.2	Rozdíl snímků	44
6.3	Optimalizace A	46
6.4	Optimalizace S	46
6.5	FIRE část A ukázka	47
6.6	Rozdíl snímků	48
6.7	Rozdíl snímků	49
A.1	Kriteriální funkce	58
A.2	Testování posunu v Y	59
A.3	Testování rotace	59
A.4	Testování zvětšení v X	60
A.5	Testování zvětšení v Y	60
A.6	Testování zkosení v Y	61
B.1	Ukázka předzpracovaného snímku	62

Seznam tabulek

4.1	Testování různých kroků učení	35
4.2	Testování optimálního počtu iterací	35
4.3	Testování různých úrovní podvzorkování	36
4.4	Testování různých počtů iterací	37
4.5	Testování různých kroků učení	37
4.6	Testování parametru gamma	37
6.1	Plochy pod křivkami na části A databáze FIRE	48
6.2	Plochy pod křivkami na části S databáze FIRE	50

Úvod

Registrace obrazů nachází uplatnění v mnoha aplikacích. V medicíně umožňuje lékařům například lépe porovnat snímky jednoho pacienta v čase, nebo vytvořit zfúzovaný obraz z několika zobrazovacích modalit. [1, 2] Tato práce pracuje mimo jiné s databází snímků sítnice, kde jsou registrací řešeny tři hlavní úlohy. Vytvoření snímku s vyšším rozlišením spojením obrazů s velkým překryvem, neboli takzvané superrozlišení, vytvoření většího obrazu z několika částečně se překrývajících snímků a již zmíněné porovnávání snímků v čase.

Cílem této práce je sestavit algoritmus pro registraci obrazů s využitím automatické diferenciací, navrhnout modifikace, algoritmus otestovat a porovnat s dostupnými algoritmy. Vhodné je aby byl algoritmus poměrně univerzální. Díky využití automatické diferenciací může algoritmus efektivně počítat gradient na grafických kartách a tím výrazně snížit své výpočetní nároky.

1 Registrace obrazů

Registrace obrazů je úloha při které jsou zadány dva nebo více obrazů a je třeba jeden nebo několik z nich transformovat tak, aby pixely výsledných obrazů odpovídaly stejným, nebo co nejbližším místům v prostoru. [3] Mattes a kol. definují pro účely registrace medicínských dat v definici místo prostoru "anatomický prostor". [2] Registraci obrazů můžeme rozdělit na dva základní druhy - monomodální a multimodální. Monomodální registrace zarovnáva obrazy ze stejné modalit (například dva rentgenové snímky), multimodální obrazy z různých modalit (například CT a MRI data).[4]

Obvykle jeden z obrazů považujeme za nedeformovaný a tedy jej netransformujeme. Označme nedeformovaný obraz B a obraz, který k němu chceme registrovat A. Dále označme X_A zorné pole obrazu A, tedy soubor pozičních vektorů \mathbf{x}_A a X_B zorné pole obrazu B, tedy soubor pozičních vektorů \mathbf{x}_B . Poziční vektory jsou zobrazeny na odpovídající hodnoty intenzity $A(\mathbf{x}_A)$, $B(\mathbf{x}_B)$.

$$X_A = \{\mathbf{x}_A\}, X_B = \{\mathbf{x}_B\}; X_A \rightarrow \{A(\mathbf{x}_A)\}, X_B \rightarrow \{B(\mathbf{x}_B)\} \quad (1.1)$$

Při registraci využíváme zvolenou transformaci T_α , kde α je vektor parametrů transformace. Transformací transformujeme obraz A na A', který pak prostorově odpovídá obrazu B.

$$T_\alpha : \mathbf{x}_{A'} = T_\alpha(\mathbf{x}_A), A'(\mathbf{x}_{A'}) = A(\mathbf{x}_A), \mathbf{x}_{A'} = \mathbf{x}_B \quad (1.2)$$

Protože si zorná pole neodpovídají, je porovnání obrazů možné dělat jen v průniku zorných polí $X_{A'}$ a X_B , který označíme Ω_α . Průnik Ω_α závisí na vektoru transformačních parametrů α .

$$\Omega_\alpha = X_{A'} \cap X_B, X_{A'} = \{T_\alpha(\mathbf{x}_A)\}, X_B = \{\mathbf{x}_B\} \quad (1.3)$$

Cílem registrace je nalézt vektor parametrů α , tak aby byla optimalizována kriteriální funkce $c(B(\mathbf{x}_B), A'(T_\alpha(\mathbf{x}_A)))$.

$$\alpha_0 = \arg \max_\alpha c(B(\mathbf{x}_B), A'(T_\alpha(\mathbf{x}_A))) \in \Omega_\alpha \quad (1.4)$$

[3]

Před provedením registrace může být vhodné data předzpracovat, zejména odstraněním šumu. Poté je třeba rozhodnout, které transformaci odpovídá deformace obrazu. Dále je třeba stanovit vhodnou kriteriální funkci, pomocí které je průběžně odhadováno, jak kvalitně jsou obrazy zarovnány. Při použití metod založených na extrakci charakteristických rysů je třeba zvolit, které rysy budou extrahovány a vytvořit postup jejich extrakce. Následuje volba optimalizačního algoritmu. Poté optimalizační algoritmus iterativně hledá nejvhodnější zarovnání. [4]

1.1 Kriteriaální funkce

Kriteriaální funkce slouží ke kvantifikování kvality zarovnání obrazů. Při registraci lze využít gradientu kriteriaální funkce k hledání optimálního zarovnání.

1.1.1 Střední kvadratická chyba (MSE)

Jednoduchá a často využívaná metrika[1].

$$MSE = \sum_i (x_i - y_i)^2$$

x_i ve vzorci značí pixel na i -té pozici v zarovnávaném obraze, y_i značí pixel na i -té pozici v referenčním obraze

1.1.2 Střední absolutní chyba (MAE)

Jednoduchá metrika, méně využívaná, než *střední kvadratická chyba* (MSE).

$$MAE = \sum_i |x_i - y_i|$$

x_i ve vzorci značí pixel na i -té pozici v zarovnávaném obraze, y_i značí pixel na i -té pozici v referenčním obraze

1.1.3 Vzájemná informace

Tato metoda je vhodná pro multimodální registraci obrazů. Vzájemná informace je původně definována jako míra závislosti dvou náhodných proměnných. [2, 4, 5] Můžeme ji vyjádřit pomocí vzorce:

$$MI = H_A + H_B - H_{AB}$$

Kde H_A , H_B a H_{AB} jsou dány vztahy:

$$H_A = - \sum_{k=0}^{q-1} P(k) \log P(k)$$

$$H_B = - \sum_{k=0}^{r-1} P(k) \log P(k)$$

$$H_{AB} = - \sum_{k=0}^{q-1} \sum_{l=0}^{r-1} P(k, l) \log P(k, l)$$

kde $q-1$ a $r-1$ jsou maximální intenzity v jednotlivých obrazech a $P(x)$ pravděpodobnost výskytu dané intenzity v obraze. [3]

1.1.4 Diferencovatelná vzájemná informace

Vzájemná informace není diferencovatelná, což může komplikovat její využití jako kritériální funkce. V její základní podobě ji proto nelze využít spolu s optimalizací pomocí automatické diferenciace, která je předmětem této práce. Vzájemnou informaci však lze nahradit podobnou diferencovatelnou funkcí. [6, 7]

U standardní vzájemné informace využíváme k výpočtu pravděpodobnosti výskytu pixelu dané intenzity v obraze histogram. Pokud chceme, aby byla automatická diferenciace diferencovatelná, nahradíme ho takzvaným měkkým histogramem. [7]

U klasického histogramu přispívá každý pixel obrazu do jednoho kyblíku histogramu, do kterého patří dle svojí intenzity. V případě měkkého histogramu, přispívá každý pixel obrazu podle své intenzity do několika kyblíků histogramu. [8]

1.1.5 Kovariance

Kovariance je dána vzorcem:

$$COV = \sum_i (x_i - \bar{x})(y_i - \bar{y})$$

kde jeden obraz je označen X, intenzita jeho i-tého pixelu x_i a průměrná intenzita jeho pixelu \bar{x} , druhý obraz je označen Y, intenzita jeho i-tého pixelu y_i a průměrná intenzita jeho pixelu \bar{y} . [3]

1.1.6 Normalizovaná křížová korelace

Normalizovaná křížová korelace je dána vzorcem:

$$NCC = \frac{\sum_i (x_i y_i)}{\sqrt{\sum_i (x_i^2)} \sqrt{\sum_i (y_i^2)}}$$

kde intenzita i-tého pixelu prvního obrazu je označena x_i a intenzita i-tého pixelu druhého obrazu je označena y_i [3]

Pokud obrazy uvažujeme jako vektory, kde každý pixel je jedna složka vektoru, pak normalizovaná křížová korelace odpovídá kosinu úhlu mezi vektorem prvního obrazu a vektorem druhého obrazu. Proto bývá někdy normalizovaná křížová korelace označována jako kosinové kritérium. Výhodou metriky je invariance k zesvětlení nebo ztmavení a k lineárním změnám kontrastu. [3]

1.2 Metody založené na ploše a metody založené na charakteristických rysech

Při registraci obrazu můžeme využít metody založené na ploše, nebo metody založené na extrakci charakteristických rysů. Metody založené na extrakci charakteristických rysů poskytují výhodu při registraci multimodálních obrazů. [1]

1.3 Transformace

1.3.1 Afinní transformace

Afinní transformace sestávají z posunutí (translace), otočení, zvětšení nebo zmenšení v jednotlivých osách a zkosení. Afinní transformaci lze vyjádřit transformační maticí. Pokud jsou známy souřadnice bodu před transformací a je třeba znát souřadnice bodu po transformaci, lze využít následující postup. Souřadnice bodu před transformací jsou zapsány jako sloupcový vektor. K vektoru je zdola přidán jeden řádek obsahující hodnotu jedna. Vektor je vynásoben transformační maticí. Výsledkem jsou souřadnice bodu po afinní transformaci.

1.3.2 Transformace s využitím kontrolních bodů

Další možností transformace je určit kontrolní body, popsat na kterou pozici se transformuje každý kontrolní bod a definovat pravidlo pro transformaci ostatních bodů v závislosti na transformaci kontrolních bodů v jejich okolí. Mattes a spol. využívají tohoto principu a body, které nejsou kontrolními dopočítávají pomocí kubických B-splínů. Tento postup je vhodný pokud není deformace rovnoměrná po celém obraze. Například u tomografických snímků kde hraje roli pohyb bránice při dýchání. [2]

1.4 Optimalizace

K optimalizaci lze využít gradientních metod nebo evolučních algoritmů. V práci je využit gradientní sestup, implementovaný pomocí automatické diferenciací.

1.5 Algoritmy pro registraci obrazů

1.5.1 H-M 16

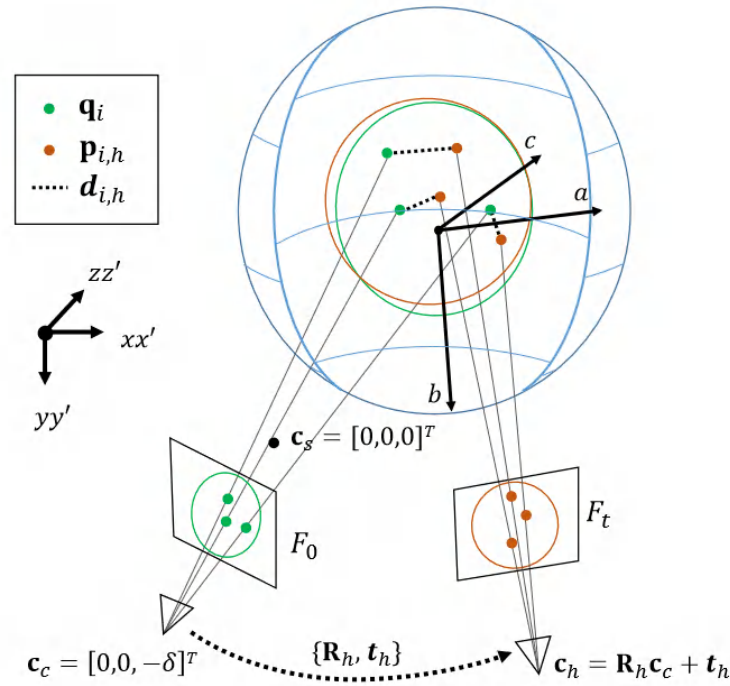
Algoritmus je vyvinut specificky pro registraci snímků sítnice. Předpokládá sférický, nebo elipsoidní tvar lidského oka a odhaduje polohy kamer, ze kterých byly pořízeny

snímky. Využívá extrakce charakteristických bodů. Všichni autoři algoritmu, jsou rovněž autory databáze FIRE. Databáze FIRE je popsána v kapitole 3.2 a je stěžejní pro testování algoritmu navrženého v této práci a jeho porovnávání s ostatními algoritmy. [9]

Prvotní odhad polohy kamer je získán pomocí metody RANSAC a s využitím sférického modelu oka. Následně je využito rojové optimalizace k nalezení přesné polohy kamer a odhadu délek poloos oka, čímž je sférický model nahrazen elipsoidním. [9]

Algoritmus využívá metody SIFT k extrakci významných bodů. V dřívějších verzích algoritmu byla použita metoda SURF, ta byla nahrazena z důvodu přesnější lokalizace významných bodů při použití SIFT. [9]

Uspořádání kamer a oka je definováno dvanácti parametry. Tři parametry udávají délky poloos oka, tři udávají rotaci oka. Další tři parametry udávají vzájemnou translaci kamer v 3D prostoru a poslední tři parametry určují vzájemnou rotaci kamer. Při registraci je tedy potřeba optimalizovat 12D funkci.[9]



Obr. 1.1: Model oka a umístění kamer H-M 16 [9]

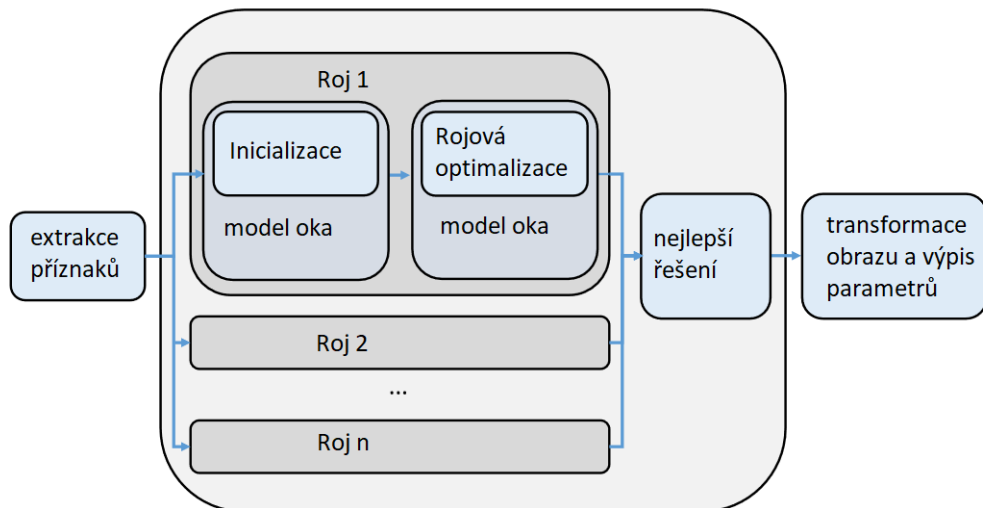
Model oka a umístění kamer využívaný algoritmem H-M 16 je zachycen na obrázku 1.1. Referenční snímek je označen F_0 , pohyblivý obraz F_t . Počátek souřadnic je označen c_s , souřadnice první kamery c_c . Odhadované souřadnice druhé kamery v

h -té iteraci jsou označeny c_h . Poloosy oka jsou označeny a , b , c . R_h značí rotační matici v h -té iteraci, t_h vektor posunu mezi kamerami v h -té iteraci. Označení q_i (zeleně) nese místo na sítnici, kam se promítá i -tý příznak z referenčního obrazu. Označení $p_{i,h}$ (oranžově) pak nese místo na sítnici, kam se promítá i -tý příznak z posuvného obrazu v h -té iteraci. Přerušovanou čarou a písmeny $d_{i,h}$ jsou označeny rozdíly, mezi místem na sítnici na které se promítá i -tý příznak z referenčního obrazu a z posuvného obrazu. Sumu těchto rozdílů se snaží algoritmus minimalizovat.

1.5.2 REMPE

Algoritmus REMPE je pokračováním vývoje algoritmu H-M 16 od stejných autorů. Algoritmus byl vyvíjen a testován s využitím databáze FIRE.[10, 11]

Při vylepšení algoritmu prošli autoři mnoho různých způsobů optimalizace, kombinací použitých příznaků a několik různých modelů oka. Nejpatrnějším zlepšením oproti výše zmíněnému algoritmu je použití kombinace příznaků SIFT v kombinaci s detekcí bifurkací cév na sítnici. [11]



Obr. 1.2: Schéma fungování algoritmu REMPE [11]

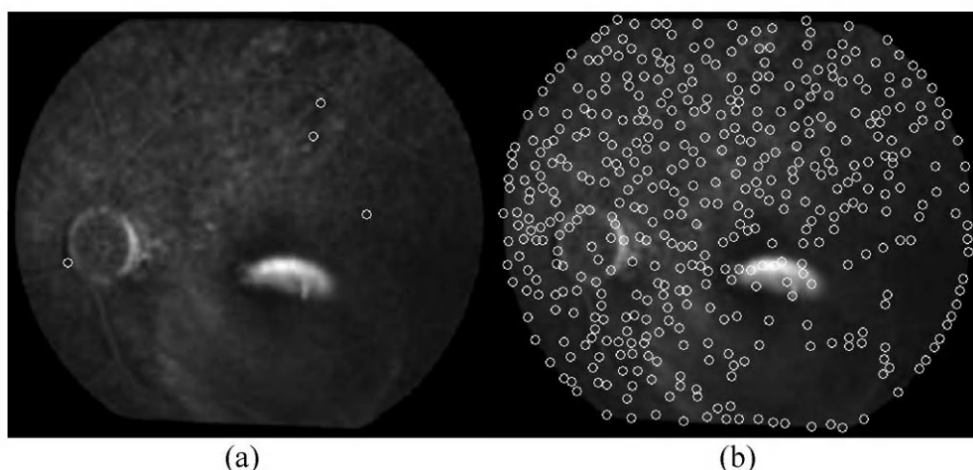
Princip fungování algoritmu je znázorněn na obrázku 1.2. Nejprve jsou extrahovány příznaky pomocí SIFT a nalezeny bifurkace cév. Dále je inicializován model oka a umístění kamer pomocí algoritmu RANSAC. Parametry modelu jsou dále optimalizovány rojovou optimalizací. Protože je rojová optimalizace stochastický proces, je inicializováno několik rojů souběžně a nakonec je z nich vybráno nejlepší řešení. [11]

Algoritmus byl implementován v programovacím jazyce C++ s využitím knihovny OpenCV. [11]

1.5.3 Harris-PIIFD

Algoritmus si klade za cíl registrovat multimodální obrazy sítnice, které jsou výrazně postiženy šumem. Algoritmus využívá vlastní extrakce příznaků pojmenované PIIFD (Partial intensity invariant feature descriptor, překlad: *Deskriptor příznaků částečně invariantní k intenzitě*). PIIFD je invariantní k afinní transformaci, jasů a změnám perspektivy. Algoritmus pracuje v sedmi krocích. [12]

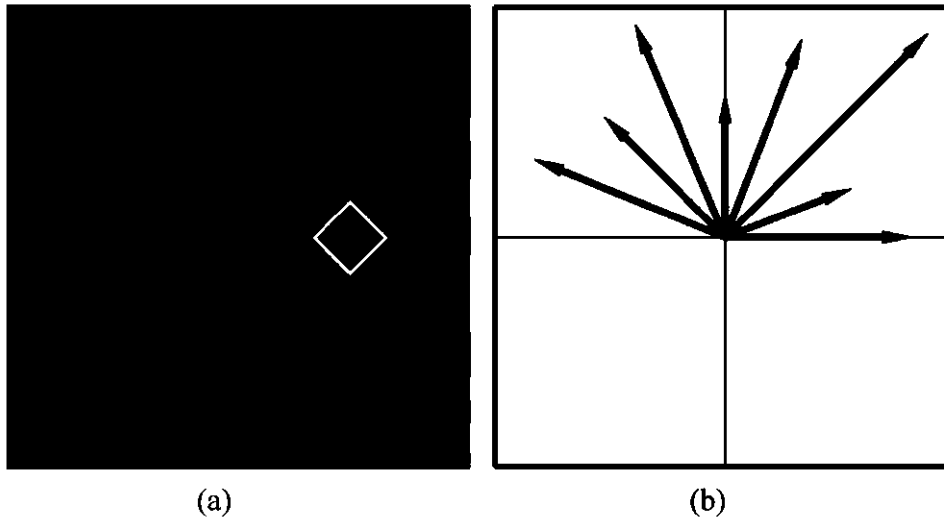
V prvním kroku jsou detekovány rohové body Harrisovým detektorem. Detekováno je přibližně 200 rohových bodů. Tyto nejsou využity jako příznaky pro registraci, ale na jejich pozicích jsou pak hledány příznaky pomocí PIIFD. Metoda funguje i pokud jsou rohové body nahrazeny body vygenerovanými náhodně, jen se sníženou přesností. Na obrázku 1.3 je zobrazena detekce rohových bodů Harrisovým detektorem (b) v porovnání s detekcí bifurkací cév (a) na zašuměném snímku sítnice. Je zde patrné, že na zašuměných snímcích může detekce bifurkací selhávat, zatímco Harrisův detektor stále umožňuje detekci mnoha rohových bodů. [12]



Obr. 1.3: Detekované bifurkace cév (a) a rohové body detekované Harrisovým detektorem (b) na zašuměném snímku sítnice. [12]

Ve druhém kroku je každému rohovému bodu přiřazena hlavní orientace. Určení hlavní orientace rohových bodů umožňuje určovat orientaci příznaků detekovaných dále PIIFD vzhledem k hlavní orientaci a tím dosáhnout invariance vzhledem k rotaci celého obrazu. Pro určení hlavní orientace je využit průměrovaný kolmý směr gradientu (v anglickém originále averaged perpendicular direction of gradient), který nabývá hodnot $\langle 0, \pi \rangle$. Využití směru v rozmezí $\langle 0, \pi \rangle$ usnadňuje registraci multimodálních obrazů, kde směr klasického gradientu může být opačný, hlavní orientace jsou ale stejné.

[12]



Obr. 1.4: Ukázka vymezení okolí kandidátního bodu (a) a lokálního degradovaného histogramu směrů[12]

Ve třetím kroku jsou extrahovány příznaky pomocí PIIFD. Extrakce příznaků funguje podobně jako u algoritmu SIFT. Nejprve je vymezeno okolí rozdělené do čtverců 4×4 . Okolí je vždy orientováno ve směru hlavní orientace a jeho velikost byla empiricky stanovena na 40×40 pixelů. V každém ze 16ti čtverců tvořících okolí je pak zvlášť vypočítán histogram směrů gradientu. Histogram má 16 kbelíků, ale aby bylo zamezeno náhlým změnám směru, jsou gradienty přiřazovány do dvou nejbližších kbelíků a to rozděleny bilineární interpolací. [12]

Aby bylo zajištěno částečné invariance k intenzitě jsou hodnoty v histogramu gradientů normalizovány. Normalizace spočívá v nahrazení nejvyšších 20% velikostí gradientů hodnotou 1, dalších 20% hodnotou 0,75 a tak dále až nejnižších 20% je nahrazeno nulou. [12]

Dále jsou protější směry histogramů sečteny, čímž jsou vytvořeny degradované histogramy s osmi kbelíky. Ze získaných 16ti degradovaných histogramů je pak dalšími úpravami popsanými v [12] získán deskriptor PIIFD. [12]

Ve čtvrtém kroku jsou deskriptory registrovaných obrazů mapovány na sebe bilaterální spojovací metodou. [12]

V pátém kroku jsou vyřazovány nesprávně namapované body. Je očekáváno, že správně namapované body budou mít přibližně podobný rozdíl v orientaci. Dvojice spojených bodů, které mají rozdíl v orientaci výrazně odlišný od většiny ostatních bodů jsou tedy vyřazeny. Dále algoritmus předpokládá, že eukleidovské vzdálenosti mezi spojenými body budou u všech správně spojených dvojic přibližně podobné. Spojené dvojice které mají mezi sebou výrazně odlišnou vzdálenost než většina ostat-

ních jsou tedy taky vyřazeny. [12]

V šestém kroku jsou zpřesněny polohy nalezených bodů. V okolí 5×5 pixelů od každého ze spojených bodů jsou vypočítávány deskriptory PIIFD. Nakonec jsou zvoleny body jejichž deskriptory vykazují nejvyšší podobnost. [12]

V sedmém a posledním kroku je zvolen druh transformace podle množství spojených dvojic bodů. Podporovány jsou lineární transformace, afinní transformace a polynomiální transformace druhého řádu. Jediným kritériem pro výběr transformace je nalezení dostatečného počtu spojených dvojic, aby byla transformace jednoznačně definována. [12]

1.5.4 GDB-ICP

Algoritmus GDB-ICP slouží k registraci širokého spektra obrazových dat. byl testován na snímcích budov, krajiny i různých medicínských datech. Při vývoji byl kladen důraz nejen na správnou registraci, ale i na vyhodnocení, kdy registrace není možná z důvodu absence překryvu nebo příliš velké deformace obrazu. Algoritmus funguje ve třech blocích, kterými jsou inicializace, odhad a rozhodovací blok. [13]

V bloku inicializace jsou detekovány příznaky a pro každou dvojici odpovídajících si příznaků je vypočtena lokální transformace pro malé okolí. V bloku odhadu probíhá zvětšování lokálních okolí. V rozhodovacím bloku je pak vyhodnoceno, zda odhadnutá transformace odpovídá správnému zarovnání obrazů. [13]

1.5.5 DRMIME

<https://www.overleaf.com/project/634d1e3093086462a879a83f>

Algoritmus DRMIME slouží k registraci monomodálních i multimodálních dat. Využívá metriky mutual information(MI) a je plně diferencovatelný. Metriku MI počítá metodou MINE, založenou na neuronových sítích. [4]

Využívá pyramidového přístupu, který je však upraven, tak aby nepracoval od nejhrubšího rozlišení k nejjemnějšímu, ale se všemi úrovní pyramidy současně. Kvůli drobnému posunu struktur v obraze při rozmazání a podvzorkování, používá algoritmus mírně upravené transformační matice pro transformaci nejjemnější rozlišení v obrazové pyramidě. [4]

K ušetření místa v paměti grafické karty algoritmus vybere v každé iteraci jen část pixelů, které se účastní výpočtu kritériální funkce. Autoři algoritmus implementovali ve dvou verzích, kdy jedna vybírá k výpočtu náhodně 10% pixelů, druhá najde hrany Cannyho detektorem a používá pouze pixely v okolí hran. [4]

Algoritmus byl testován na databázi FIRE[10] obsahující monomodální data a na databázi ANHIR[14] obsahující multimodální data. [4]

1.5.6 Nerigidní multimodální registrace obrazů - Mattes a kolektiv

Algoritmus slouží k registraci 3D snímků hrudníku z PET a CT. Motivací pro registraci těchto snímků je vysoké rozlišení obrazů z CT a fyziologický význam dat z PET. [2]

Při PET vyšetření je těsně před snímáním radiofarmaka pořízen prozařovací snímek. Získaný snímek je velmi podobný CT, ale je získán zářením o vyšší intenzitě, což způsobuje horší kontrast tkání a má oproti CT snímkům horší rozlišení. Algoritmus tedy neregistruje CT snímky k obrazu získanému vyzařováním radiofarmaka, ale k prozařovacím snímkům, získaným těsně před snímáním radiofarmaka. [2]

Algoritmus využívá kontrolních bodů umístěných v pravidelné mřížce a interpolaci B-spliny. Kriteriaální funkcí je vzájemná informace v upravené spojitě podobě. K upravení vzájemné informace do spojitě podoby jsou histogramy, ze kterých je vzájemná informace počítána vytvořeny s použitím Parzenova okna. Derivace kriteriační funkce jsou vyjádřeny analyticky. Optimalizace probíhá na základě gradientu. [2]

K urychlení výpočtu a snížení pravděpodobnosti uvíznutí v lokálním minimu je využita registrace v několika fázích, kde se mění rozmazání vstupních obrazů Gaussovým konvolučním jádrem, počet kontrolních bodů a další parametry. Jedná se o postup podobný pyramidovému přístupu. [2]

1.5.7 DIRNet

Algoritmus DIRNet využívá k registraci neuronové sítě. Tvoří jej tři hlavní části, kterými jsou neuronová síť, prostorový transformátor a resampler. Neuronová síť je učena bez učitele, po naučení se je schopna registrovat obrazy na jeden průchod. [15]

Referenční a pohyblivý obraz nejprve vstupují do konvoluční neuronové sítě. Neuronová síť analyzuje výřezy z referenčního a pohyblivého obrazu a vypočítá lokální deformační parametry pro prostorový transformátor. Prostorový transformátor vygeneruje pole vektorů posunu. To vstupuje do resampleru, který pak transformuje pohyblivý obraz. Ve fázi učení je dále vypočítána metrika podobnosti a jsou upraveny váhy neuronové sítě. [15]

Algoritmus byl implementován s použitím knihoven Theano a Lasagne [16]. Testován byl úspěšně na datasetu ručně psaných číslic MNIST a na MRI snímcích srdce. [15]

1.6 Software pro registraci obrazů

1.6.1 Elastix

Elastix je software pro rigidní a nonrigidní transformaci obrazů, zejména mediálních. Využívá knihovnu Insight Toolkit (ITK)[17]. Elastix umožňuje uživateli vytvořit registrační algoritmus pomocí parametrového souboru. Z příkazové řádky pak uživatel program spustí s parametrovým souborem a danými obrazy a ten provede registraci. [18]

Elastix podporuje různé transformace, například rigidní, afinní nebo transformaci s využitím kontrolních bodů. Nabízí mnoho kritériálních funkcí jako například MSE, křížovou korelaci nebo vzájemnou informaci. Program podporuje využití pyramidového přístupu a to ve dvou formách - rozmazání Gaussovým konvolučním jádrem s podvzorkováním a bez podvzorkování. K urychlení výpočtu umožňuje program využití pouze některých pixelů k výpočtu kritériální funkce. Pixely mohou být zvoleny například v okolí hran v referenčním obraze, nebo náhodně. [18]

1.6.2 Simple Elastix

Simple Elastix je toolkit pro Python, Javu, C# a další jazyky, obsahující knihovnu Elastix. Navíc nabízí předkonfigurované metody registrace, které mohou sloužit jako výchozí bod při návrhu registračního algoritmu. [19]

1.6.3 Kornia

Kornia je diferencovatelná knihovna pro programovací jazyk Python umožňující snadnou integraci metod klasického strojového učení do modelů pro hluboké učení. Sestává z několika modulů, které slouží například pro práci s barvami, kritériální funkce nebo extrakci znaků. [20]

Kornia je silně inspirována knihovnou OpenCV. Hlavními výhodami oproti OpenCV je diferencovatelnost a kompatibilita s PyTorch. Nevýhodou je nižší optimalizovanost pro použití v produkci a embedded zařízeních. Záměrem autorů tedy je, aby Kornia sloužila k trénování neuronových sítí, které pak budou nasazeny pomocí jiných knihoven, například OpenCV. [20]

Knihovna Kornia je strukturována do několika modulů podle různých témat z oblasti počítačového vidění. Vlastnosti jednotlivých modulů jsou shrnuty níže. [20]

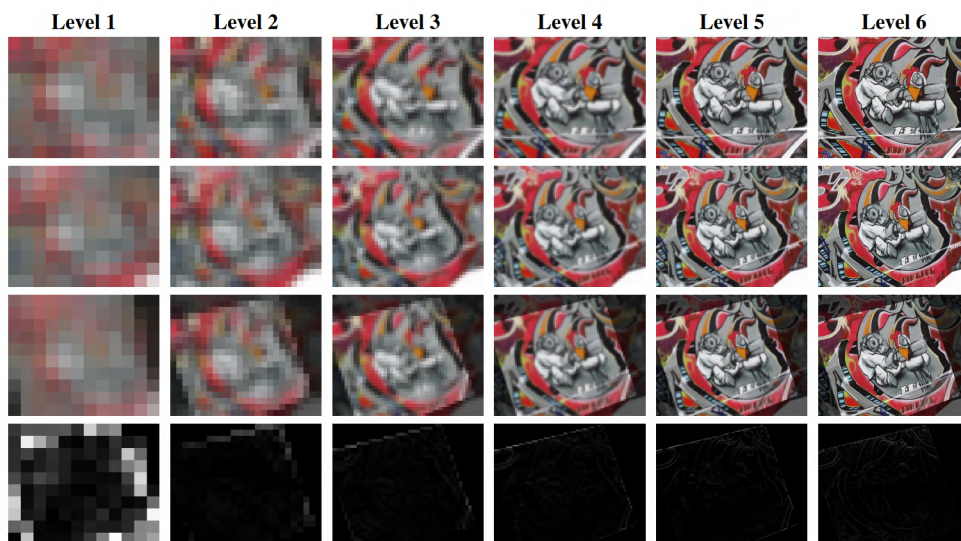
Modul *kornia.augmentation* obsahuje funkce k augmentaci dat pro trénování neuronových sítí. Podporuje práci s dávkami (batch). Příklady funkcionalit jsou: náhodné rotace, afinní a perspektivní transformace, různé náhodné transformace intenzit barev, přidání šumu a rozmazání pohybem. [20]

Modul *kornia.color* umožňuje konverze mezi barevnými prostory. Funkce modulu jsou podobné jako některé funkce knihoven OpenCV a Scikit-image, ale Kornia zavádí modifikace umožňující podporovat výpočty s plovoucí desetinnou čárkou. Umožňuje přechod například mezi následujícími barevnými prostory: šedotónový, RGB, RGBA, BGR, HSV. Dále umožňuje výpočet barevných histogramů, úpravu kontrastu, gamma korekci a jiné. [20]

Modul *kornia.features* obsahuje funkce pro detekci charakteristických rysů (features). Modul obsahuje například tyto funkce: Harrisův rohový detektor, Shi-Tomasiho rohový detektor, detekci charakteristických rysů metodou SIFT a detekci charakteristických rysů deeplearningovými metodami HardNet a SOSNet. Funkce jsou implementovány diferencovatelně. [20]

Modul *kornia.filters* zahrnuje různé lineární i nelineární filtry. Obsahuje například funkce pro rozmazání pomocí Gaussiánu, nebo obdelníkovým oknem, mediánový filtr, nebo Sobelův hranový detektor. Všechny zmíněné funkce jsou diferencovatelné. [20]

Dalšími moduly jsou moduly *kornia.geometry*, *kornia.losses* a *kornia.contrib*. První zmíněný model umožňuje geometrické transformace včetně afinních transformací. Druhý zmíněný model nabízí různé kriteriální funkce. Poslední zmíněný model obsahuje různé experimentální operátory. [20]



Obr. 1.5: Registrace s využitím pyramidového přístupu a metody gradientního sestupu pomocí knihovny Kornia [21]

Na obrázku 1.5 jsou znázorněny výsledky registrace metodou gradientního sestupu pomocí knihovny Kornia. Každý sloupec reprezentuje jednu úroveň pyramidy.

První řádek reprezentuje pohyblivý obraz, druhý referenční obraz, třetí registrovaný obraz a čtvrtý chybu registrace.[21]

1.6.4 AIRLab

AIRLab je knihovna pro programovací jazyk Python zaměřená na registraci medicínských obrazů. AIRLab využívá knihovny PyTorch a umožňuje výpočty na GPU. Pro práci se vstupy a výstupy využívá AIRLab knihovny SimpleITK[22]. Autoři uvádí záměr vytvořit knihovnu umožňující rychlé prototypování registračních algoritmů. [23]

Knihovna klade poměrně velký důraz na regularizaci algoritmu pro registraci. Nabízí čtyři různé regularizátory - difúzní regularizátor, regularizátor využívající celkové variace anizotropní i izotropní a sparsity regularizátor, penalizující nenulové koeficienty. [23]

Poslední příspěvek do této knihovny na GitHubu[24] je z 8. října 2020, připomínky (issues) na GitHubu zůstávají bez reakce. Z toho lze usuzovat, že knihovna již pravděpodobně není dále rozšiřována a vyvíjena.

2 Navržený algoritmus

2.1 Popis algoritmu

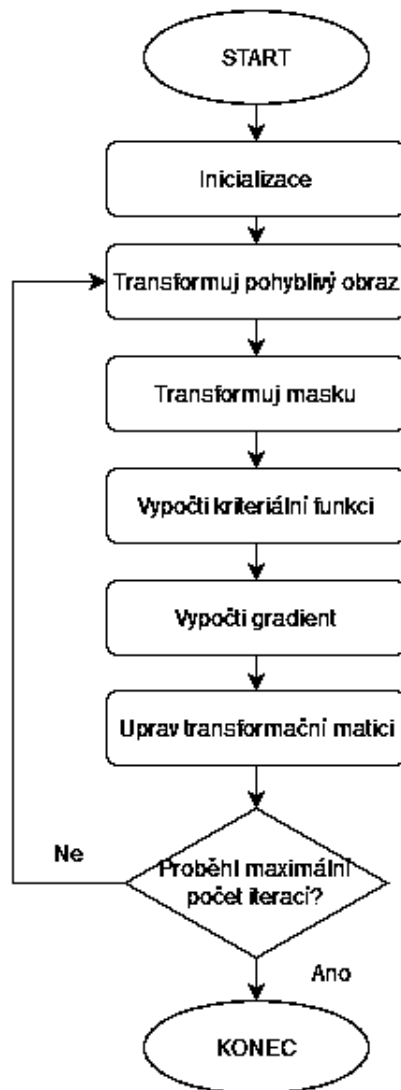
Algoritmus iterativně transformuje pohyblivý obraz a masku pomocí aktuální transformační matice. Poté vypočte z referenčního obrazu, pohyblivého obrazu a masky hodnotu kritériální funkce. Následně vypočte gradient kritériální funkce pomocí automatické diferenciaci. Dále upraví podle vypočteného gradientu transformační parametry a vypočte novou transformační matici. Pokud proběhl požadovaný počet iterací algoritmus končí, pokud ne, proběhne další iterace. Blokované schéma algoritmu je na obrázku 2.1.

2.2 Pyramidový přístup

Algoritmus byl vylepšen s využitím pyramidového přístupu. Před vykonáním algoritmu je třeba zvolit koeficienty, podle kterých budou obrazy podvzorkovány. Na začátku algoritmu jsou referenční obraz, pohyblivý obraz i maska podvzorkovány nejhrubším koeficientem. Poté proběhne předem daný počet iterací algoritmu s podvzorkovanými obrazy. Následně jsou obrazy a maska podvzorkovány jemnějším koeficientem a opět proběhne daný počet iterací algoritmu. Obdobě se opakuje podvzorkování všemi zvolenými koeficienty, nakonec proběhne algoritmus pro nepodvzorkované obrazy. Pyramidový přístup omezuje pravděpodobnost uvíznutí v lokálním minimu a snižuje potřebný výkon pro registraci. Schéma modifikace s využitím pyramidového přístupu je znázorněno na obrázku 2.2

2.3 Změna kroku učení pomocí scheduleru

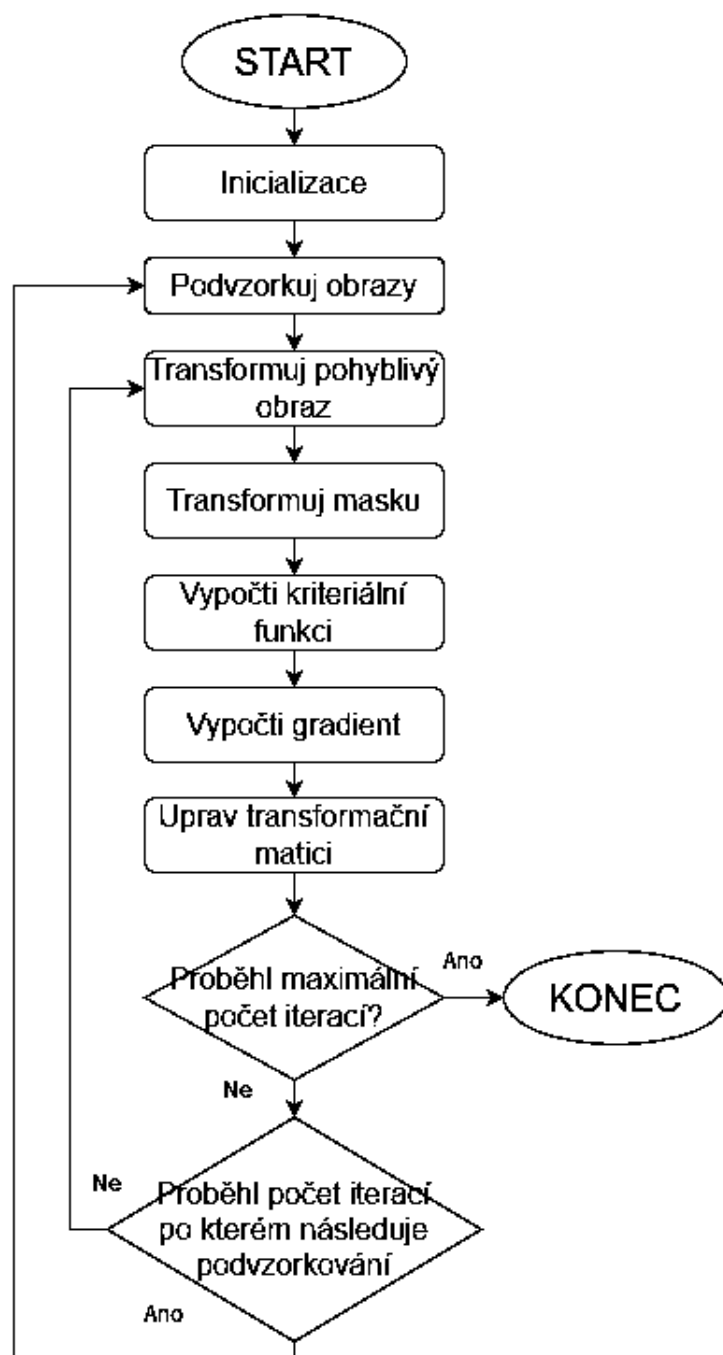
Na začátku registrace je výhodnější, když algoritmus modifikuje transformační parametry ve větších krocích. Je tím dosaženo rychlejšího přiblížení k optimálnímu řešení a omezení uvíznutí v lokálním minimu. V závěru registrace naopak bývá výhodnější postupovat po malých krocích, aby se algoritmus nevzdaloval od optima a našel jej s co nejvyšší přesností. Proto můžeme v průběhu registrace měnit krok učení. Scheduler použitý v této práci mění krok učení tak, že jej vždy po předem daném počtu iterací vynásobí koeficientem γ . Schedulerem lze rozšířit jak základní verzi algoritmu, tak jej zkombinovat s pyramidovým přístupem.



Obr. 2.1: Blokové schéma navrženého algoritmu v základní verzi

2.4 Implementace navrženého algoritmu

Navržený algoritmus využívá knihoven Scikit image[25], Numpy, [26] a PyTorch [27]. Nejprve jsou vytvořeny proměnné pro translaci ve směrech x a y , zvětšení ve směrech x a y , zkosení ve směrech x , y a rotaci. Zvětšení jsou inicializovány hodnotou jedna, ostatní parametry hodnotou nula. Dále je v každé iteraci ze zmíněných proměnných vytvořena transformační matice. Pomocí funkce z knihovny PyTorch je registrovaný obraz transformován. Následně je vypočten gradient a s využitím optimizéru Adam jsou vypočteny nové hodnoty posunů a rotace. Poté se cyklus opakuje až do dosažení požadovaného počtu iterací.



Obr. 2.2: Blokové schéma navrženého algoritmu v modifikaci s využitím pyramidového přístupu

3 Testovací databáze

3.1 Rentgenové snímky hrudníku

Testovací sada obsahuje 100 8bitových šedotónových rentgenových snímků hrudníku o výšce 1280 pixelů a šířce 1024 pixelů. Snímky byly získány z databáze[28] a upraveny, aby měly všechny stejný rozměr. Příklad vstupu z testovací databáze si můžete prohlédnout na obrázku 3.1. Snímky byly transformovány počítačem, aby byly vytvořeny páry obrazů k registraci. Díky tomu známe přesně hledané transformační matice. Hlavní testovací sada obsahovala afinně transformované obrazy s posuny v x a y v rozsahu $\pm 5\%$ šířky a výšky obrazu, otočením $\pm 12^\circ$, zvětšením 90% až 110% a koeficienty zkosení v rozmezí -0,1 až 0,1. Transformační parametry byly generovány rovnoměrným rozložením. Dále byly vytvořeny databáze pro dílčí transformace (například pouze posun v X).



Obr. 3.1: Příklad obrázku z testovací databáze

3.2 FIRE: Fundus Image Registration Dataset

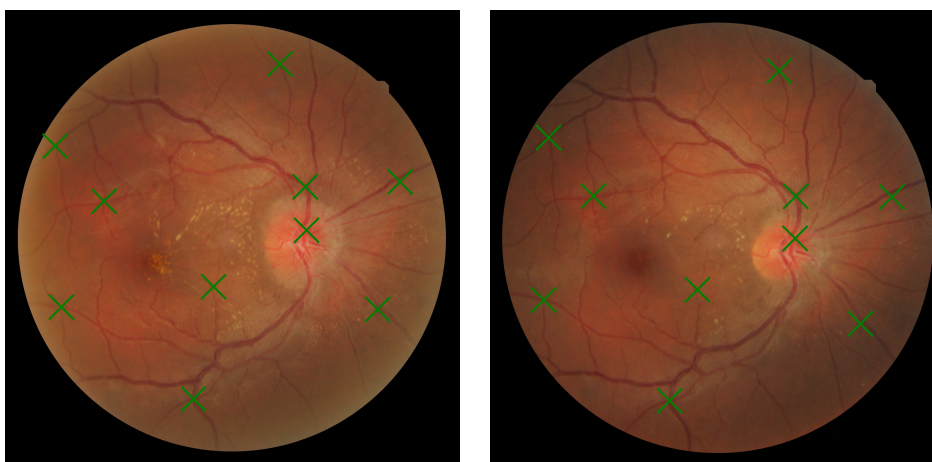
Tato testovací databáze je kompletně převzata [10]. Databáze obsahuje 134 párů snímků k registraci, z důvodů popsanych níže však bylo k testování zde navrženého algoritmu použito jen 85 párů. V databázi jsou zaznamenané pozice kontrolních bodů obou snímků k registraci, díky které lze objektivně měřit přesnost registrace. Na webu databáze jsou zaneseny výsledky několika registračních algoritmů na této databázi, se kterými lze srovnat algoritmus navržený v této práci. Databáze je rozdělena do tří částí A, P a S.

3.2.1 Parametry snímků

Snímky byly pořízeny kamerou Nidek AFC-210. Zorný úhel je 45° . Rozlišení snímků je 2912×2912 . Snímky byly získány na jednotce léčby hypertenze třetího oddělení interní medicíny nemocnice Papageorgiou v Řecku. Jedná se o snímky sítnic 39 různých pacientů ve věkovém rozmezí 19-67 let, pořízené při pravidelných kontrolách mezi lety 2006 a 2015. [10]

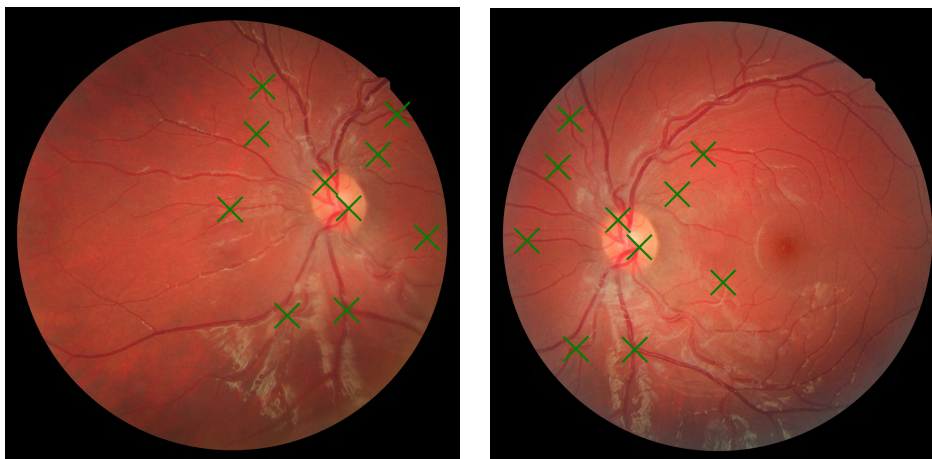
3.2.2 Popis částí databáze

Část A obsahuje 14 párů snímků s anatomickými změnami. Snímky byly pořízeny při dvou návštěvách lékaře s časovým odstupem a ve snímcích lze pozorovat změny vlivem progresu nebo remise retinopatie. Tato část slouží k testování registrace pro studie probíhající v delším čase. Příklad páru snímků k registraci z části databáze A je na obrázku 3.2.



Obr. 3.2: Příklad snímků z části databáze A s vyznačenými významnými body [10]

Část P obsahuje 49 párů snímků s malým překryvem. Tato část databáze nebyla využita proto, že při velkých posunech hraje významnou roli zakřivení sítnice a deformaci nelze vhodně aproximovat afinní transformací, která je předmětem této práce. [9, 11] Účelem této části databáze je registrace pro mozaikování, tedy vytváření snímků zobrazující větší plochu, než jsme schopni sejmout fundus kamerou v jednom snímku. Ukázka páru snímků z části databáze P lze vidět na obrázku 3.3. V obrázku jsou vyznačeny body, jejichž souřadnice jsou součástí datasetu a které slouží k vyhodnocení přesnosti registrace.



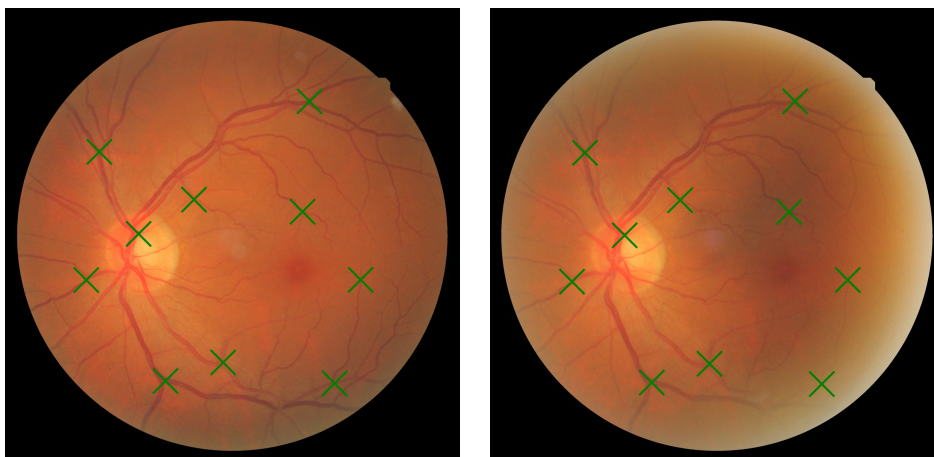
Obr. 3.3: Příklad snímků z části databáze P s vyznačenými významnými body [10]

Část S obsahuje 71 párů snímků s velkým překryvem, bez anatomických změn. Tato část je nejjednodušší na registraci. Záměrem autora je testování registrace pro vytvoření snímků se super rozlišením. Pár snímků z databáze je znázorněn na obrázku 3.4.

3.2.3 Doporučená metrika

Součástí databáze jsou i souřadnice významných bodů v obou snímcích z páru k registraci. Tyto body jsou označeny zelenými křížky na obrázcích 3.2, 3.3 a 3.4. Autoři databáze navrhnou i metrickou pro porovnání algoritmů na této databázi. [10] Metrikou je plocha pod křivkou závislosti přesnosti registrace na hranici od které považujeme registraci za úspěšnou.

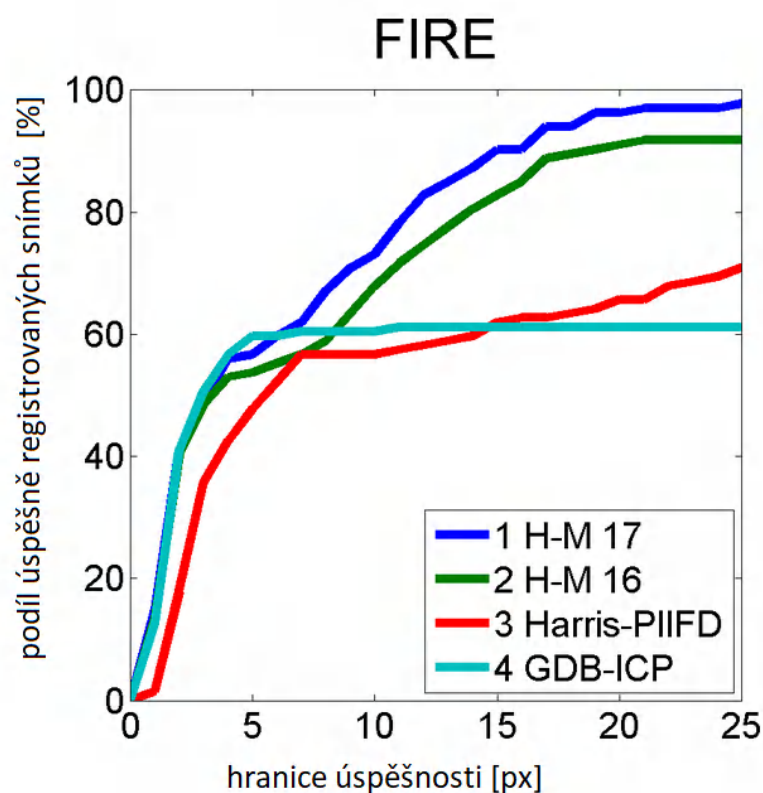
Chyba registrace dvojice snímků je dána jako průměrná eukleidovská vzdálenost v pixelech mezi pozicemi na které byly registrovány kontrolní body a místy na které odpovídající kontrolní body měly být registrovány. Registrace je pak považována za úspěšnou pokud je chyba registrace maximálně rovna hranici úspěšnosti. Do grafu



Obr. 3.4: Příklad snímků z části databáze S s vyznačenými významnými body [10]

vynášíme podíl úspěšně registrovaných snímků v závislosti na nastavení hranice, které testujeme v rozmezí 0 - 25 pixelů. Podíl plochy pod křivkou vzhledem k celé ploše grafu je pak výsledná metrika AUC.

[10]



Obr. 3.5: Křivky úspěšnosti v závislosti na hranici úspěšnosti. [10]

Na webových stránkách datasetu [29] jsou pak uvedeny dosažené hodnoty AUC a křivky přesnosti v závislosti na hranici úspěšnosti pro čtyři registrační algoritmy. Jsou to algoritmy H-M 16 [9], popsány v podkapitole 1.5.1, REMPE [11], popsány v podkapitole 1.5.2, Harris-PIIFD [12], popsány v podkapitole 1.5.3 a GDP-ICP [13], popsány v podkapitole 1.5.4. Dosažené hodnoty a křivky jsou uvedeny jak pro celý dataset, tak pro jeho jednotlivé části. Na grafu 3.5 jsou zobrazeny křivky úspěšnosti zmíněných algoritmů na celém datasetu FIRE.

4 Testování hyperparametrů na rentgenových snímcích

4.1 Metrika testování

Testování proběhlo s využitím metriky Target registration error. Tato metrika využívá definovaných bodů v referenčním a posuvném obraze a chybu registrace pak stanovuje jako průměrnou eukleidovskou vzdálenost, mezi místem na které měl být daný bod být registrován a místem na které registrován byl.

Protože jsou data generována synteticky, není třeba detekovat významné body v referenčním a posuvném obraze. Jako významné body jsou určeny body na souřadnicích $[0,5;0,5]$, $[0,5;-0,5]$, $[-0,5;0,5]$ a $[-0,5;-0,5]$ v referenčním obraze. Využíváme zde souřadnic normalizovaných na rozsah $\langle -1,1 \rangle$ v obou dimenzích.

Protože je známa jak deformační matice, kterou byl vytvořen pohyblivý obraz, tak matice získaná z registrace, lze pak body jednoduše transformovat postupně pomocí obou transformačních matic. Při dokonalé registraci by se body promítly zpět samy na sebe. Chyba je vypočtena jako vzdálenost průměrná eukleidovská vzdálenost původního bodu a bodu získaného jeho postupnou transformací pomocí obou matic.

4.2 Testování hyperparametrů základní verze algoritmu

V základní verzi algoritmu bylo potřeba otestovat dva hyperparametry, kterými jsou krok učení a počet iterací. Počet iterací je třeba volit dostatečně vysoký, aby se algoritmus stihl dostatečně přesně přiblížit správnému řešení a zároveň ne zbytečně velký, aby byla omezena doba běhu programu a spotřeba výpočetního výkonu. U kroku učení je třeba, aby byl dostatečně velký k tomu, aby se algoritmus blížil požadovanému řešení rychle a neuvízl v lokálním minimu. Příliš velký krok učení však omezuje přesnost s jakou jsme schopni dosáhnout výsledku. Tyto hyperparametry se navzájem ovlivňují. Optimální krok učení pro jeden počet iterací nemusí být optimální pro druhý a naopak.

4.2.1 Testování kroku učení v základní verzi algoritmu

Algoritmus byl otestován s kroky učení 0,001, 0,003, 0,005 a 0,007. Testování probíhalo pro 120 iterací. Výsledky testování jsou zaznamenány v tabulce 4.1. Jako optimální hodnota kroku učení bylo zvoleno 0,003.

Tab. 4.1: Testování různých kroků učení

krok učení	TRE	kriteriální funkce
0,001	0,0391	8,00E-4
0,003	0,0203	2,92E-5
0,005	0.0210	1,41E-5
0,007	0.0245	1,30E-5

4.2.2 Testování počtu iterací v základní verzi algoritmu

Vhodnou hodnotu počtu iterací lze odhadnout z grafu hodnoty kriteriální funkce po jednotlivých iteracích algoritmu. Pokud hodnota kriteriální funkce významně neklesá, pravděpodobně již není třeba dále zvyšovat počet iterací.

V tabulce 4.2 je zanesen testovaný počet iterací, chyba TRE, kriteriální funkce a doba běhu programu na základní testovací databáze se 100 obrazy. Na grafu A.1 je patrný průběh kriteriální funkce pro 300 iterací algoritmu.

Jako optimální hodnota počtu iterací bylo stanoveno 120. Při zvoleném počtu iterací je doba běhu algoritmu asi jedna minuta. Inspirací pro tuto volbu byl i článek [4], ve kterém nejrychlejší algoritmy dosahovaly na databázi s obdobným počtem obrazů času běhu rovněž přibližně jedné minuty.

Tab. 4.2: Testování optimálního počtu iterací

počet iterací	TRE	kriteriální funkce	doba běhu programu
50	0,0286	8,17E-4	26,94
100	0,0211	6,38E-5	52,40
120	0,0203	2,92E-5	62,84
140	0,0199	1,94E-5	72,66
160	0,0196	1,34E-5	83,08
180	0,0194	1,00E-5	93,35
200	0,0193	8,67E-6	103,60
300	0,0189	5,03E-6	155,08

4.3 Testování hyperparametrů algoritmu modifikovaného s využitím pyramidového přístupu

U modifikace s využitím pyramidového přístupu je mimo dříve testovaných hyperparametrů třeba otestovat i počet úrovní podvzorkovací pyramidy.

4.3.1 Testování počtu úrovní pyramidy

Podvzorkovací pyramida byla vždy konstruována tak, aby předchozí patro bylo podvzorkováno dvakrát více než předchozí a poslední patro nebylo podvzorkováno vůbec. Vhodný počet úrovní pyramidy byl testován pro 30 iterací na každé úrovni podvzorkování. Výsledky testování jsou zobrazeny v tabulce 4.3. Nejlepšího výsledku bylo dosaženo pro 6 úrovní pyramidy.

Tab. 4.3: Testování různých úrovní podvzorkování

počet úrovní pyramidy	TRE	kritériální funkce	doba běhu programu
1	0,0378	3,02E-3	16,49
2	0,0183	4,81E-4	20,40
3	0,0089	9,67E-5	20,98
4	0,0062	3,84E-5	21,10
5	0,0048	1,93E-5	21,36
6	0,0040	1,25E-5	21,61
7	0,0051	1,33E-5	22,77

4.3.2 Testování počtu iterací

Při použití pyramidového přístupu je optimální počet iterací jiný, než když není použit. Proto bylo třeba optimální počet iterací hledat znovu. Počet iterací byl optimalizován pro 6 úrovní pyramidy a krok učení 0,003. Bylo zjištěno, že optimální počet iterací je 30. Výsledky testování jsou v tabulce 4.4. Pro vyšší počty iterací docházelo k poklesu kritériální funkce, ale chyba TRE se naopak zvyšovala. Uvedený počet iterací probíhá v každé úrovni pyramidy, celkem tedy proběhne iterací šestnásobek, byť většina z nich je s výrazně podvzorkovanými obrazy.

Tab. 4.4: Testování různých počtů iterací

počet iterací	TRE	kriteriální funkce	doba běhu programu
10	0,0130	4,45E-3	07,32
20	0,0045	7,07E-4	14,09
30	0,0040	6,22E-5	21,21
40	0,0051	1,25E-5	28,30

4.3.3 Testování kroku učení

Protože použitím pyramidového přístupu došlo k výrazné změně chování algoritmu, bylo třeba ověřit, zda je stále vhodné využití korku učení 0,003.

Tab. 4.5: Testování různých kroků učení

krok učení	TRE	kriteriální funkce
0,001	0,0125	2,00E-4
0,001	0,0046	3,95E-5
0,003	0,0040	1,25E-5
0,005	0,0086	2,07E-5
0,008	0,0127	6,59E-5
0,01	0,0145	1,12E-4

Tab. 4.6: Testování parametru gamma

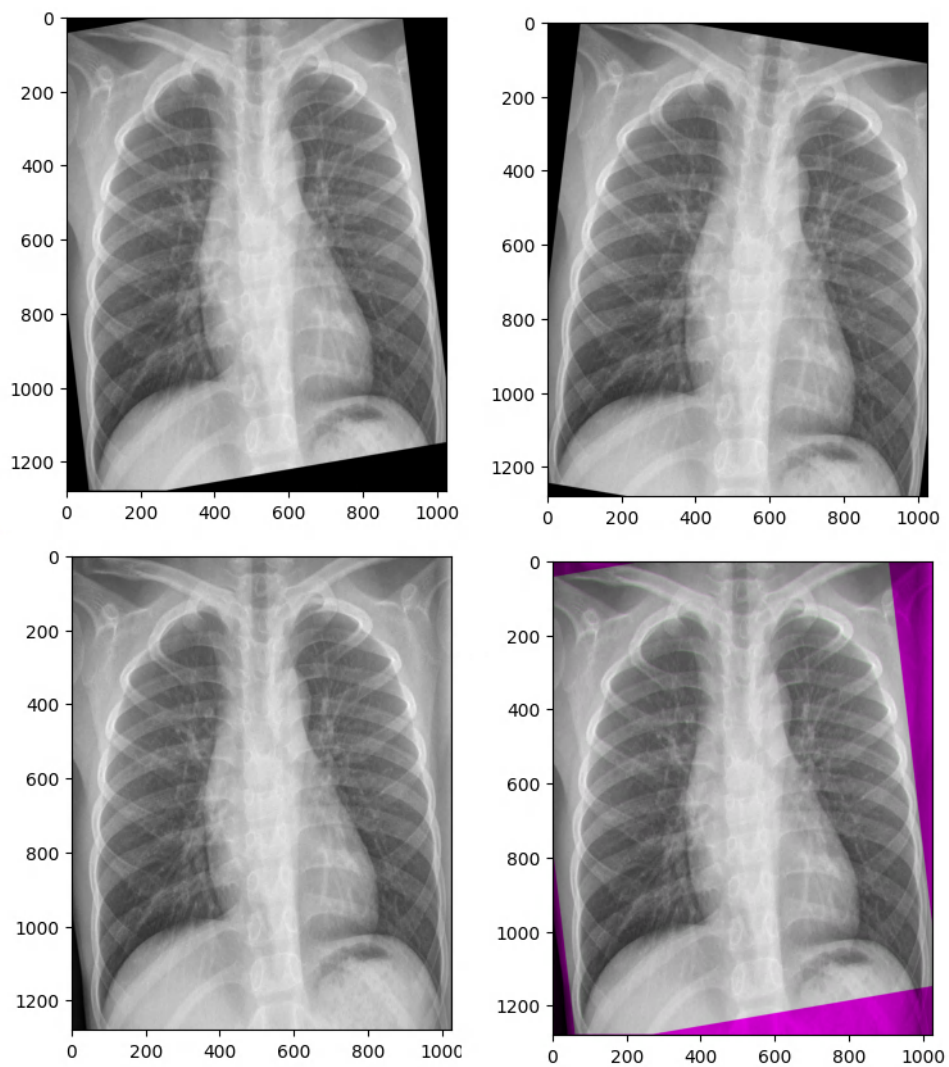
gamma	TRE	kriteriální funkce
0,2	0,0085	1,05E-5
0,3	0,0041	6,84E-6
0,4	0,0038	3,95E-6
0,5	0,0040	1,25E-6

4.4 Testování hyperparametrů algoritmu modifikovaného s využitím pyramidového přístupu a scheduleru

S použitím scheduleru je třeba otestovat parametr γ , který udává jak se mění krok učení. V tabulce 4.6 jsou výsledky testování tohoto parametru. Testování probíhalo pro krok učení 0,005, se šesti úrovněmi pyramidy a maximálním počtem iterací 200 v každé úrovni pyramidy. Optimální hodnota parametru byla stanovena na 0,4. Doba běhu programu byla 204 sekund, nezávisle na velikosti parametru γ .

4.5 Subjektivní zhodnocení výsledku

Na obrázku 4.1 jsou zobrazeny výsledky úspěšné registrace pomocí pyramidového přístupu bez scheduleru s optimalizovanými hyperparametry. Pyramidový přístup bez scheduleru se jeví jako dobrý kompromis mezi přesností jednoduchostí a rychlostí. Na překryvu referenčního obrazu v modrém a červeném kanále a registrovaného obrazu v zeleném kanále vidíme, že v tomto případě je registrace prakticky dokonale úspěšná.

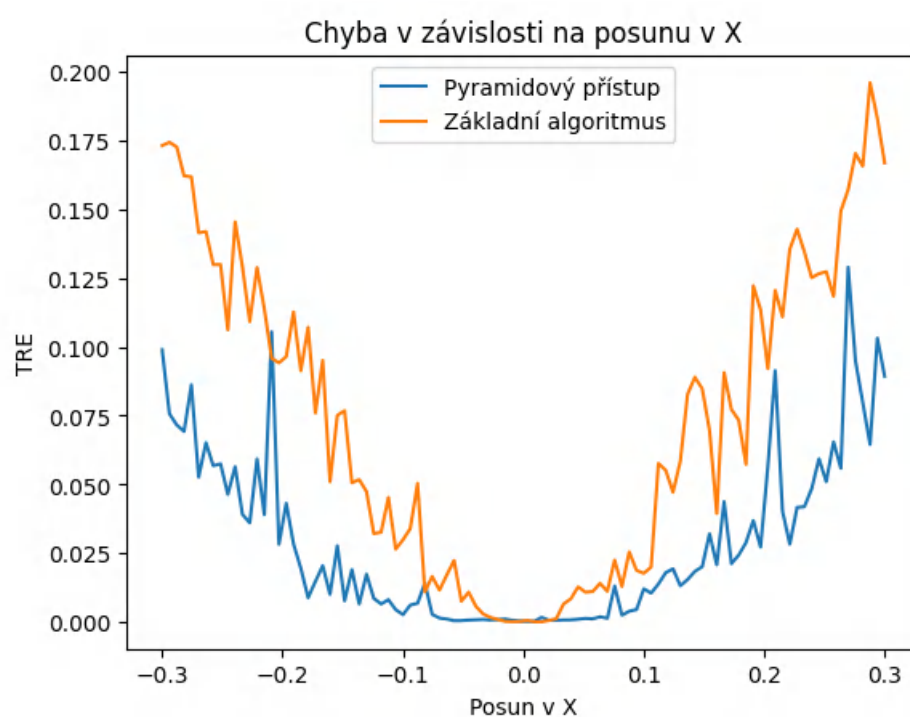


Obr. 4.1: Vlevo nahoře je registrovaný obraz, vpravo nahoře je deformovaný obraz, vlevo dole je referenční obraz, vpravo dole je vykreslen referenční obraz v modrém a červeném kanálu a registrovaný obraz v zeleném kanálu

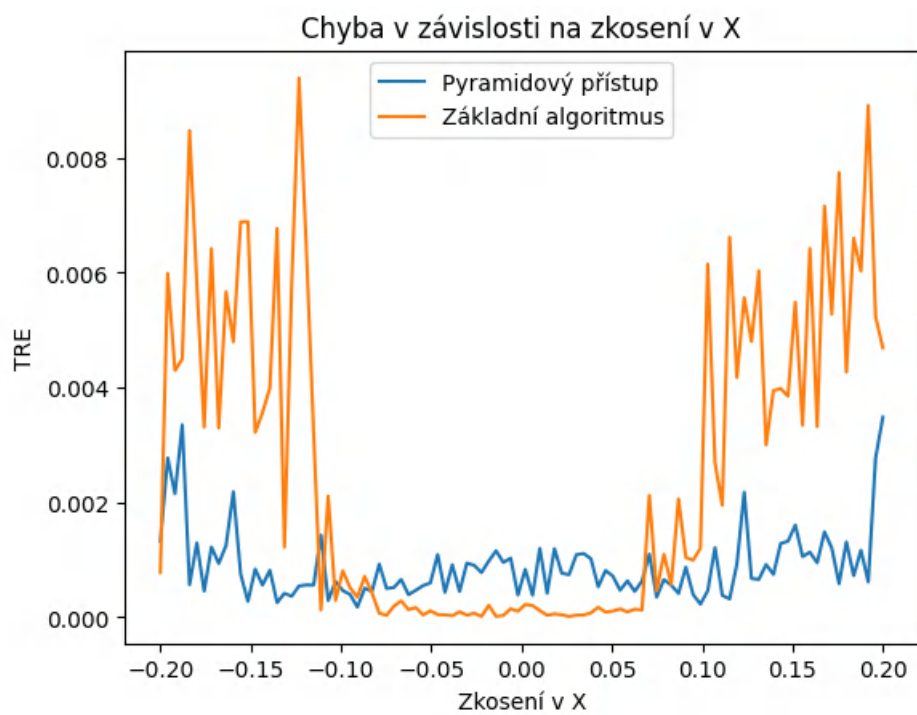
5 Testování algoritmů pro jednotlivé dílčí transformace na rentgenových snímcích

Algoritmus byl otestován na rentgenových snímcích upravených vždy jen jednou dílčí transformací. Testování umožnilo zjistit které typy deformací zvládá algoritmus hůře a které lépe a v jakém rozsahu deformací poskytuje dobrou přesnost registrace.

Testován byl v základní verzi a ve verzi s pyramidovým přístupem. Použití pyramidového přístupu se ukázalo jako výhodné u všech typů deformací s výjimkou zkosení. Ukázka testování pro posun v x a zkosení v x je v grafech 5.1 a A.6. Další grafy z testování jsou v přílohách A.2 až A.5.



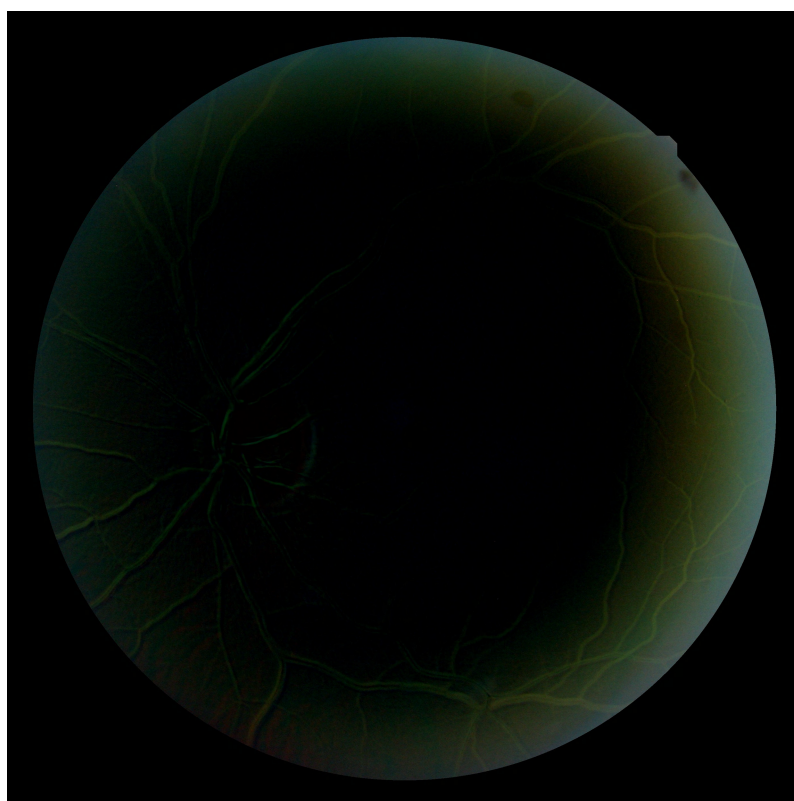
Obr. 5.1: Testování posunu v X



Obr. 5.2: Testování zkosení

6 Testování algoritmu na databázi FIRE

Pokud algoritmus dostane nepředzpracované snímky z databáze FIRE, nedosahuje dobrých výsledků. Dále je popsána příčina problému a řešení. Na obrázku 3.4 vidíme, že pravý snímek je špatně nasvětlen. Na první pohled se zdá že rozdíly v intenzitě mezi dvojicí obrázku, způsobuje spíše osvětlení, které v našem případě představuje šum, než cévy, které jsou důležité pro registraci. Na obrázku 6.1 je proto vykreslený rozdíl snímků z obrázku 3.4. Je zde patrné, že velkou většinu rozdílu intenzit opravdu způsobuje rozdílné osvětlení. Pokud tedy takovéto snímky předložíme vytvořenému algoritmu, bude se snažit primárně kompenzovat osvětlení a na rozmístění cév nebude brát velký zřetel. Proto byl algoritmus testován nejen na datasetu FIRE tak jak byl vytvořen, ale i na předzpracovaných snímcích, ve kterých jsou zvýrazněny cévy. Na takových snímcích funguje algoritmus podstatně lépe.



Obr. 6.1: Rozdíl snímků z obrázku 3.4

6.1 Předzpracování

Předzpracování bylo naprogramováno v programovacím jazyce MATLAB. Nejprve bylo získáno pozadí mediánovou filtrací s oknem 80 px. Následně byl od pozadí

odečten původní obraz. Obraz od pozadí odečítáme proto, že cévy jsou tmavší než pozadí, takže při tomto pořadí odečítání budou mít pixely cév kladné hodnoty a zápornými hodnotami se není třeba zabývat. Výsledný rozdíl je dále mediánově filtrován a naprahován. Tím zůstanou ve výsledném obraze bílé cévy a velká většina ostatních objektů je potlačena. Kód vykonávající předzpracování je přiložen ve výpisu 6.1. Příklad předzpracovaného snímku je v příloze B.1.

Výpis 6.1: Program pro předzpracování

```
close all; close all; clc
input_dir = "images"; % složka s obrazy k předzpracování
output_dir = "preprocessed_images"; % složka na výsledky

5 files = dir(input_dir);
for f = 3:length(files)
    im = imread(string(files(f).folder) + "/" + ...
                string(files(f).name));
    im_green = im(:,:,2);
10 med_diff = medfilt2(im_green, [80,80]) - im_green;
    med_diff = medfilt2(med_diff, [10,10]);
    med_diff(med_diff < mean(mean(med_diff))*5) = 0;
    imwrite(med_diff*8, output_dir + "/" + string(files(f).name))
end
```

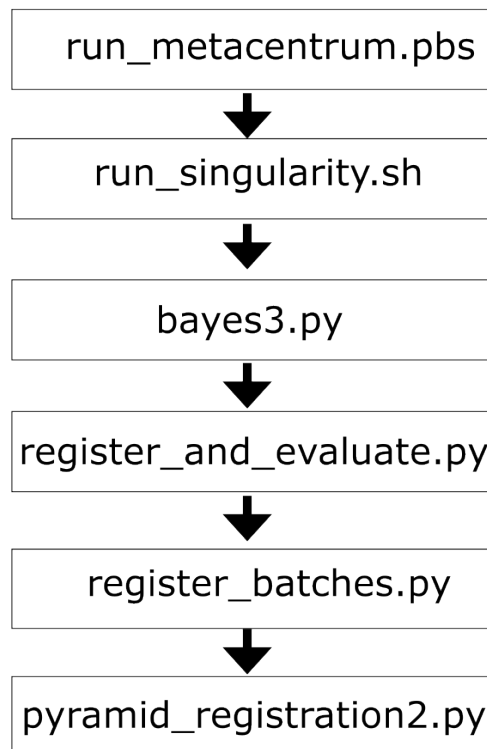
6.2 Optimalizace hyperparametrů

Algoritmus byl testován na částech A a S databáze FIRE. Část databáze P nebyla použita, protože při velkých posunech na sítnici není afinní transformace, využívaná vytvořeným algoritmem, dostatečná k aproximaci deformací.

Na každé části databáze byl algoritmus testován pro 30 a 50 iterací v každé úrovni pyramidu s dvěma různými kriteriálními funkcemi - střední kvadratickou chybou a kovariancí. Pro každou kombinaci počtu iterací a kriteriální funkce byla určena optimální hodnota kroku učení pomocí Bayesovy optimalizace. Ve všech případech byly použity čtyři úrovně pyramidu. Verze algoritmu bez pyramidového přístupu ani pyramidového přístupu se schedulerem nebyla na snímcích sítnice testována. Algoritmus bez pyramidového přístupu měl malou přesnost. Algoritmus se schedulerem má delší dobu běhu a potřebu optimalizace dalšího hyperparametru. Zlepšení přesnosti, které za tyto nevýhody nabízí není příliš velké.

6.2.1 Provedení optimalizace hyperparametrů

Z důvodu vysoké výpočetní náročnosti byla optimalizace hyperparametrů snímků sítnice provedena na gridové infrastruktuře Metacentra. Registrace probíhala pomocí šesti skriptů a funkcí, které se vzájemně volaly. Volání jsou znázorněny ve schématu 6.2. Skript nebo funkce ze které vychází šipka volá skript nebo funkci do které šipka vchází.



Obr. 6.2: Schéma skriptů a funkcí

Skripty *run_metacentrum.pbs* a *run_singularity.sh* byly poskytnuty vedoucím práce a byly upraveny pro konkrétní potřebu. Slouží ke spuštění dalších skriptů na vhodném uzlu Metacentra ve vhodném Singularity kontejneru. Skript *bayes3.py* provádí bayesovskou optimalizaci. Na určené místo uloží nejvhodnější hodnotu kroku učení, nejlepší dosaženou hodnotu AUC a tensor transformačních matic získaný při registraci. Skript volá opakovaně funkci *register_and_evaluate.py*.

Funkce *register_and_evaluate.py* dostane zadané hyperparametry registrace, umístění snímků k registraci a souborů s významnými body. Provede registraci, voláním

funkce *register_batches.py*, vyhodnotí AUC a vrátí ji.

Funkce *register_batches.py* dostane na vstupu umístění snímků k registraci a hyperparametry registrace. Vezme tolik snímků, jaká je určena velikost dávky, připraví z nich dva tensorů v potřebném formátu a předá je funkci *pyramid_registration2.py* k registraci spolu s hyperparametry.

Funkce *pyramid_registration2.py* provádí samotnou registraci. Funkce nedostává umístění obrázků, ale přímo tensorů. Proto dostává obrázky postupně, aby tensorů nezabíraly příliš mnoho paměti RAM, případně paměti grafické karty.

6.2.2 Výsledky optimalizace hyperparametrů

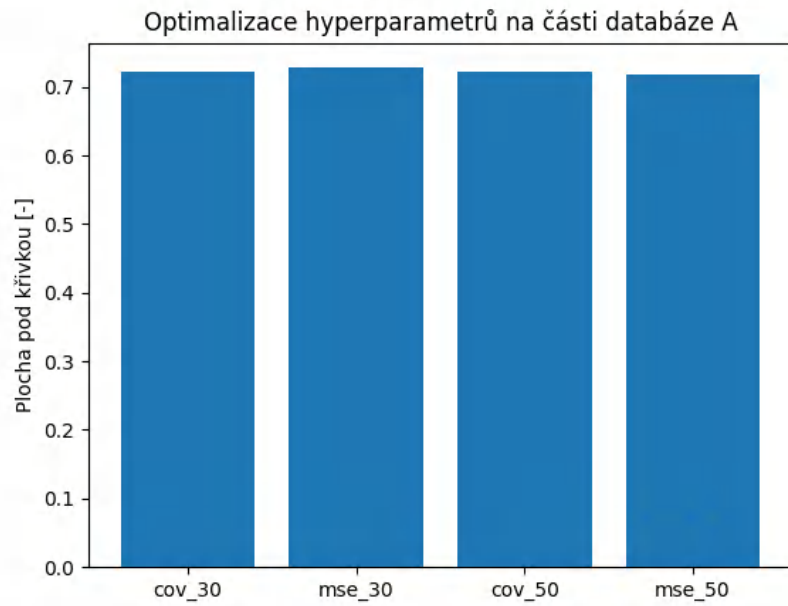
Lze očekávat že optimální parametry pro registraci snímků sítnice budou do jisté míry podobné parametrům pro rentgenové snímky. Snímky sítnice jsou výrazně větší, proto je i registrace a hledání hyperparametrů výpočetně náročnější. Ze zmíněných důvodů nebyly hyperparametry optimalizovány v tak širokém rozmezí. Pro každou testovanou kombinaci kritériální funkce a počtu iterací byl hledán optimální krok učení pomocí Bayesovy optimalizace. Bayesova optimalizace probíhala ve čtyřech iteracích se třemi body v každé iteraci.

Výsledek optimalizace na části databáze A je v grafu 6.3. U sloupců označených *cov* byla jako kritériální funkce použita kovariance, u sloupců označených *mse*, byla použita střední kvadratická chyba. Číslo za podtržítkem značí počet iterací v každé z úrovní pyramid. Nejlepších výsledků dosáhla kombinace střední kvadratické chyby spolu s třiceti iteracemi, plocha pod křivkou byla 0,728 (udávána v rozmezí 0-1).

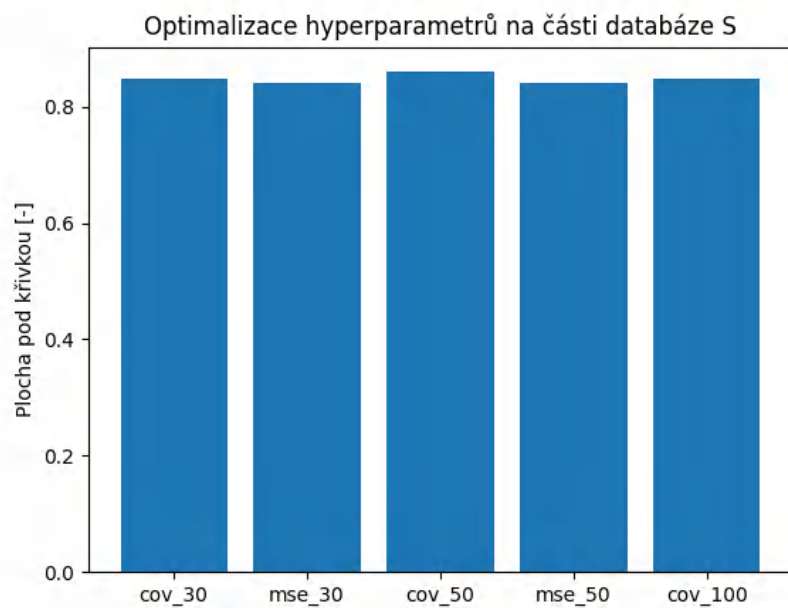
Výsledek optimalizace na části databáze S je v grafu 6.4. Označení sloupců je stejné jako v předchozím grafu. Nejlepšího výsledku dosáhla kombinace kovariance jako kritériální funkce a padesáti iterací. Dodatečně byla testována kovariance se 100 iteracemi, ale nepřinesla zlepšení.

6.2.3 Příklad registrace

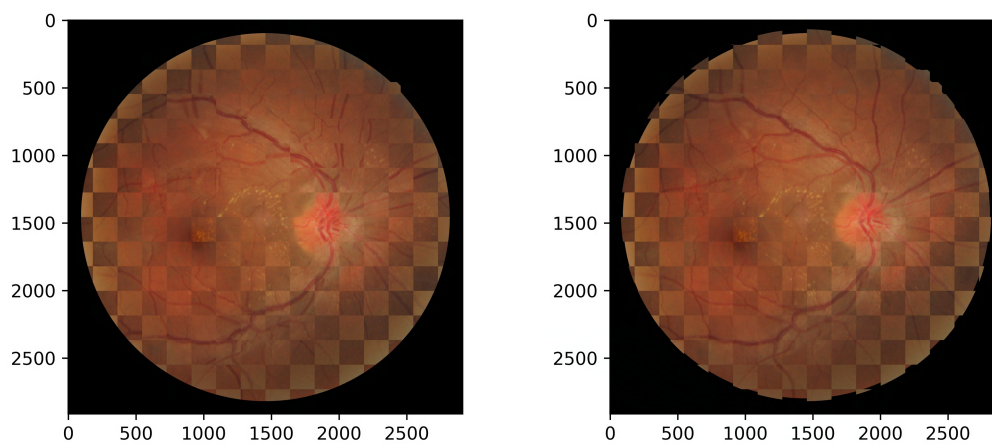
V grafu 6.5 je ukázka registrovaných snímků sítnice z části datasetu A s využitím předzpracování. Na obrázku 6.5a jsou v šachovnici vykresleny obrázky které vstupují do registrace, na obrázku 6.5b je pak referenční a registrovaný obraz. Subjektivně se registrace jeví jako poměrně dobře provedená, průměrná vzdálenost registrovaných kontrolních bodů od jejich skutečných pozic byla 2,53 pixelu.



Obr. 6.3: Výsledky optimalizace hyperparametrů na části databáze A s předzpracováním



Obr. 6.4: Výsledky optimalizace hyperparametrů na části databáze S s předzpracováním



(a) Neregistrované obrazy vykreslené v šachovnici do stejného obrázku (b) Referenční a registrovaný obraz vykreslené v šachovnici

Obr. 6.5: Ukázka registrace s nevhodnějšími parametry a využitím předzpracování na obrázku A01 z databáze FIRE [10]

6.3 Porovnání s jinými algoritmy

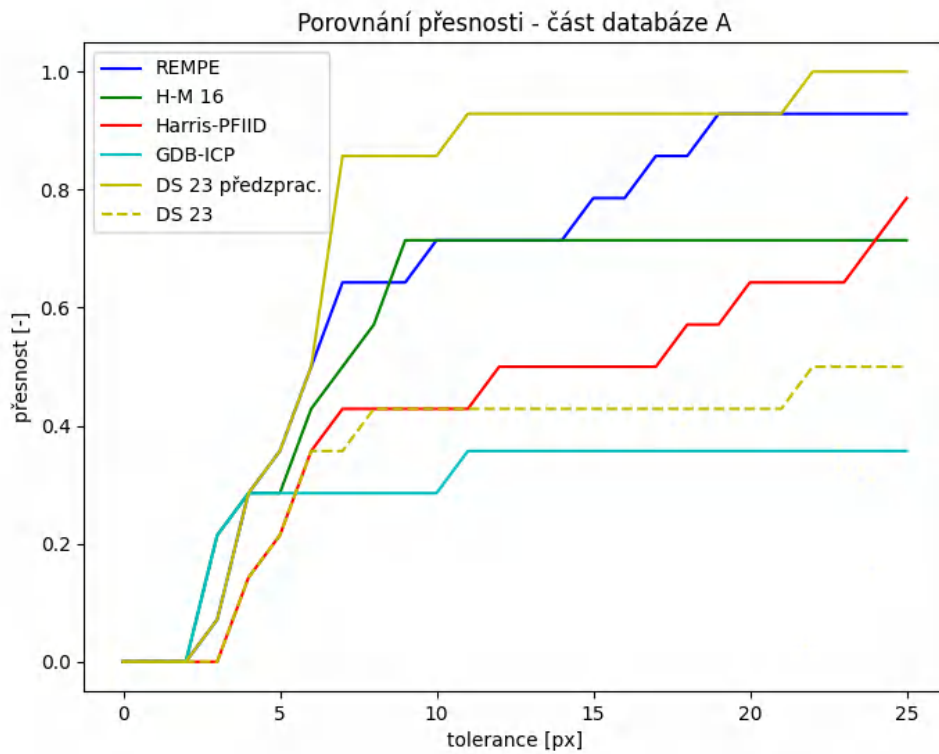
6.3.1 Část databáze A

Na obrázku 6.6 je vidět srovnání navrženého algoritmu s ostatními algoritmy, jejichž výsledky jsou zveřejněny na webu databáze FIRE. [29] Plochy pod křivkami z obrázku 6.6 jsou v tabulce 6.1.

Modře je znázorněn algoritmus REMPE [11], popsáný v podkapitole 1.5.2. Zeleně je vyznačen algoritmus H-M 16 [9], který je předchůdcem algoritmu REMPE a je popsán v podkapitole 1.5.1. Červeně je vyznačen algoritmus Harris-PFIID [12], popsáný v podkapitole 1.5.3. Tyrkysovou barvu je vykreslena křivka přesnosti algoritmu GDB-ICP [13], který jako jediný z algoritmů dostupných pro srovnání na datasetu FIRE není určen čistě k registraci snímků sítnice, ale registruje široké spektrum obrazů. Tento algoritmus je popsán v podkapitole 1.5.4.

Algoritmus navržený v této práci je označen žlutou barvou a zkratkou *DS 23*. Byl otestován jak na datech předzpracovaných způsobem popsáným v kapitole 6.1 tak na nepředzpracovaných datech. Výsledky na předzpracovaných datech jsou označeny plnou čarou, výsledky na nepředzpracovaných datech čarou přerušovanou.

Způsob jakým byly získány grafy přesnosti je popsán v podkapitole 3.2.3. Je to standardní způsob vyhodnocení registrace na databázi FIRE [10]. Na ose x je zaznamenána tolerance v pixelech, na ose y je pak zaznamenáno u jaké části dvojic



Obr. 6.6: Porovnání algoritmu s dalšími registračními algoritmy na části databáze A.

Tab. 6.1: Plochy pod křivkami na části A databáze FIRE

algoritmus	plocha pod křivkou
REMPE	0,660
H-M 16	0,577
Harris-PFIID	0,443
GDB-ICP	0,303
DS 23 předzprac.	0,728
DS 23	0,351

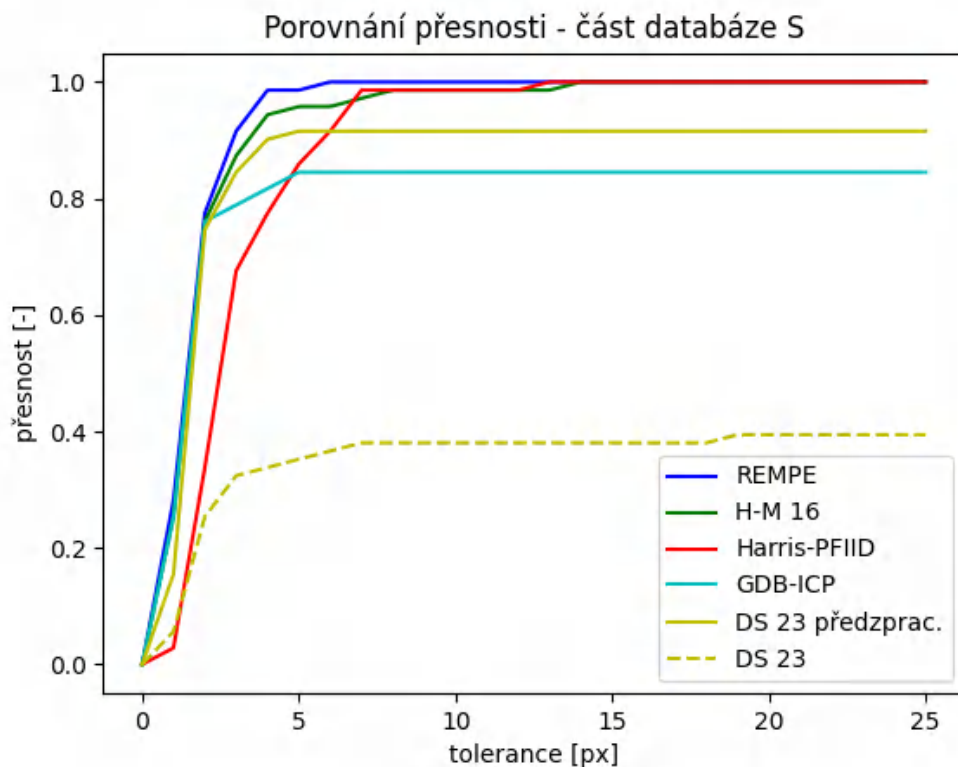
obrazů dosáhl algoritmus stejné nebo lepší metriky TRE než je tolerance.

Je patrné, že předzpracování přináší velkou výhodu. Zatímco bez předzpracování dosáhl algoritmus druhého nejhoršího výsledku. S využitím předzpracování dosáhl výsledku nejlepšího a byl jediným algoritmem, kterému se podařilo úspěšně regis-

trovat všech 14 párů snímků z této části databáze s tolerancí nanejvýš 25 pixelů. Vzhledem k tomu, že v části databáze A se vyskytují měnící se retinopatie, bylo očekáváno, že předzpracování odečtením pozadí přinese významnou výhodu. I přesto jsou výsledky navrženého algoritmu s použitím předzpracování překvapivě dobré s ohledem na jeho podstatně jednodušší stavbu, než mají ostatní porovnávané algoritmy.

Dále lze v grafu pozorovat, že všechny tři algoritmy specializované na snímky sítnice dosahují lepších výsledků, než algoritmus GDB-ICP, který má všeobecné využití.

6.3.2 Část databáze S



Obr. 6.7: Porovnání algoritmu s dalšími registračními algoritmy na části databáze S.

Barevné označení křivek je totožné jako u výsledků na části databáze A a je popsáno v podkapitole 6.3.1.

Část databáze S, která obsahuje snímky s malým posunem sejmuté s malým časovým odstupem. Tyto snímky jsou jednodušší na registraci a algoritmy na ní

Tab. 6.2: Plochy pod křivkami na části S databáze FIRE

algoritmus	plocha pod křivkou
REMPE	0,958
H-M 16	0,945
Harris-PFIID	0,900
GDB-ICP	0,814
DS 23 předzprac.	0,860
DS 23	0,349

obecně dosahují lepších výsledků. Navržený algoritmus bez předzpracování zde selhává a je s velkým odstupem nejhorší. Křivky přesností algoritmů v závislosti na toleranci na části databáze S jsou zobrazeny na obrázku 6.7, plochy pod křivkami jsou pak v tabulce 6.2.

Při použití předzpracování se dostává mezi Algoritmus GDB-ICP s všeobecným využitím a trojici algoritmů specializovaných na snímky sítnice, které již od tolerance 15 pixelů dosahují stoprocentní účinnosti. Je patrné, že u snímků nepostižených měnicími se retinopatiemi nepřináší předzpracování tak velkou konkurenční výhodu.

7 Diskuze

Navržený algoritmus využívá registrace založené na ploše. Implementovaná funkce podporuje dvě různé kriteriální funkce - kovarianci a střední kvadratickou chybu. Rozdíly v kvalitě registrace při použití těchto dvou kriteriálních funkcí jsou malé. Algoritmus využívá pyramidový přístup, který přinesl významné zlepšení zejména ve snížení výpočetních nároků, ale i v přesnosti registrace. Výhodou je využití automatické diferenciací pomocí knihovny PyTorch [27], díky ní může algoritmus počítat gradient efektivně na grafických kartách.

Algoritmus funguje dobře na jednoduchých syntetických datech. Při použití ne předzpracovaných reálných snímcích sítnice však podává poměrně špatné výsledky. Přesnost se podařilo velmi výrazně zlepšit použitím předzpracování snímků sítnice, tak aby vynikly cévy a bylo potlačeno vše ostatní. Bylo očekáváno, že předzpracování přinese zlepšení, ale míra zlepšení byla překvapivá.

Na nezašuměných datech pomohlo předzpracování zvýšit přesnost z hodnoty prakticky nepoužitelné pro reálné aplikace na hodnotu která se blíží registračním algoritmům vyvinutým specificky pro registraci snímků sítnice. Na části databáze postižené měnicími se retinopatiemi dosáhl algoritmus při použití předzpracovaných dat dokonce lepšího výsledku než všechny algoritmy, které jsou na webu databáze prezentovány. Srovnání algoritmu registrujícího předzpracovaná data s algoritmy registrujícími nepředzpracovaná data je však třeba brát s velkou rezervou protože vstupy algoritmů nejsou stejné.

Výsledky testování ukazují na velký význam předzpracování při registraci obrazu. Vzhledem k tomu, že tento poměrně jednoduchý algoritmus dokázal při použití předzpracovaných dat dosáhnout lepší registrace snímků sítnice s měnicími se retinopatiemi, než sofistikované algoritmy silně specializované na snímky sítnice, stálo by za probádání, zda by i ostatní algoritmy netěžily významně z podobného předzpracování. Výhoda toho, že mají algoritmy REMPE [11], H-M 16 [9] a Harris-PIIFD [12] poměrně úzce vymezená data, která registrují je, že by mohlo být možné navrhnout předzpracování, které je dostatečně obecné, aby dokázalo zpracovat všechny jejich možné vstupy.

Implementaci algoritmu by bylo dále možné vylepšovat zejména přidáním dalších kriteriálních funkcí. Obzvláště vhodná by byla vzájemná informace. U ni je šance že by se mohla lépe vypořádat s nerovnoměrným osvětlením i bez předzpracování. Také by bylo zajímavé testovat algoritmus na dalších typech dat.

Závěr

V teoretické části byla nejprve stručně popsána registrace obrazů, s důrazem na popis různých kriteriálních funkcí. Dále byly popsány algoritmy pro registraci medicínských dat. Zvláštní pozornost byla věnována algoritmům u kterých jsou zveřejněny výsledky na databázi snímků sítnice FIRE [10] a se kterými byl následně porovnáván algoritmus navržený v této práci. Dále byly popsány knihovny pro registraci obrazů, využívající automatickou diferenciaci pro výpočet gradientu.

V praktické části byl navržen algoritmus pro registraci obrazů s využitím automatické diferenciaci. Navržený algoritmus byl naprogramován v programovacím jazyce Python, s využitím knihovny PyTorch [27]. Byly vytvořeny modifikace navrženého algoritmu s využitím pyramidového přístupu a s použitím scheduleru. Testováním na rentgenových snímcích bylo zjištěno, že použití pyramidového přístupu i scheduler mohou zlepšit přesnost navrženého algoritmu. Scheduler však zvyšuje přesnost poměrně nevýrazně za cenu významného zvýšení výpočetních nároků a zvýšení komplexnosti algoritmu, proto bylo dále od jeho používání upuštěno.

Algoritmus byl dále testován na částech databáze snímků sítnice FIRE [10] a to ve verzi s pyramidovým přístupem. Na částech databáze FIRE byl také srovnán s dalšími algoritmy. Když dostával algoritmus nepředzpracované snímky sítnice dosahoval poměrně špatných výsledků. Podařilo se navrhnout jednoduchou metodu předzpracování, která výsledky výrazně zlepšila. Metoda je popsána v podkapitole 6.1 Porovnávání algoritmu, který dostává předzpracované snímky s algoritmy, které dostávají nepředzpracované snímky však může být problematické.

Literatura

- [1] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003.
- [2] David Mattes, David R Haynor, Hubert Vesselle, Thomas K Lewellyn, and William Eubank. Nonrigid multimodality image registration. In *Proceedings of SPIE*, volume 4322, pages 1609–1620. SPIE, 2001.
- [3] Jiří Jan. *Medical image processing, reconstruction and analysis : cocepts and methods*. CRC Press, London, second edition edition, 2020.
- [4] Abhishek Nan, Matthew Tennant, Uriel Rubin, and Nilanjan Ray. Drmime: Differentiable mutual information and matrix exponential for multi-resolution image registration. 2020.
- [5] Jorge R. Vergara and Pablo A. Estévez. A review of feature selection methods based on mutual information. *Neural computing applications*, 24(1):175–186, 2014.
- [6] Songbai Ji, Ziji Wu, Alex Hartov, David W. Roberts, and Keith D. Paulsen. Mutual-information-based image to patient re-registration using intraoperative ultrasound in image-guided neurosurgery. *Medical physics (Lancaster)*, 35(10):4612–4624, 2008.
- [7] Pytorch implementation of a differentiable mutual information measure. URL: https://github.com/KrakenLeaf/pytorch_differentiable_mutual_info.
- [8] Ibrahim Yusuf, George Igwegbe, and Oluwafemi Azeez. Differentiable histogram with hard-binning. 2020.
- [9] Carlos Hernandez-Matas, Xenophon Zabulis, and Antonis A. Argyros. Retinal image registration through simultaneous camera pose and eye shape estimation. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, volume 2016-, pages 3247–3251, United States, 2016. IEEE.
- [10] Carlos Hernandez-Matas, Xenophon Zabulis, Areti Triantafyllou, Panagiota Anyfanti, Stella Douma, and Antonis A Argyros. Fire: Fundus image registration dataset. *Modeling and Artificial Intelligence in Ophthalmology*, 1(4):16–28, 2017.

- [11] Carlos Hernandez-Matas, Xenophon Zabulis, and Antonis A. Argyros. Rempe: Registration of retinal images through eye modelling and pose estimation. *IEEE journal of biomedical and health informatics*, 24(12):3362–3373, 2020.
- [12] Jian Chen, Jie Tian, Noah Lee, Jian Zheng, R. Theodore Smith, and Andrew F. Laine. A partial intensity invariant feature descriptor for multimodal retinal image registration. *IEEE transactions on biomedical engineering*, 57(7):1707–1718, 2010.
- [13] Gehua Yang, C.V. Stewart, M. Sofka, and Chia-Ling Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1973–1989, 2007.
- [14] Jiří Borovec, Jan Kybic, Ignacio Arganda-Carreras, Dmitry Sorokin, Gloria Bueno, Alexander Khvostikov, Spyridon Bakas, Eric Chang, Stefan Heldmann, Kimmo Kartasalo, Leena Latonen, Johannes Lotz, Michelle Noga, Sarthak Pati, Kumaradevan Punithakumar, Pekka Ruusuvaori, Andrzej Skalski, Nazanin Tahmasebi, Masi Valkonen, and Arrate Muñoz-Barrutia. Anhir: Automatic non-rigid histological image registration challenge. *IEEE Transactions on Medical Imaging*, PP:1–1, 04 2020. doi:10.1109/TMI.2020.2986331.
- [15] Bob D. de Vos, Floris F. Berendsen, Max A. Viergever, Marius Staring, and Ivana Išgum. End-to-end unsupervised deformable image registration with a convolutional neural network. In M. Jorge Cardoso, Tal Arbel, Gustavo Carneiro, Tanveer Syeda-Mahmood, João Manuel R.S. Tavares, Mehdi Moradi, Andrew Bradley, Hayit Greenspan, João Paulo Papa, Anant Madabhushi, Jacinto C. Nascimento, Jaime S. Cardoso, Vasileios Belagiannis, and Zhi Lu, editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 204–212, Cham, 2017. Springer International Publishing.
- [16] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, diogo149, Brian McFee, Hendrik Weideman, takacsg84, peterderivaz, Jon, instagibbs, Dr. Kashif Rasul, CongLiu, Britefury, and Jonas Degraeve. Lasagne: First release., August 2015. doi:10.5281/zenodo.27878.
- [17] Matthew McCormick, Xiaoxiao Liu, Luis Ibanez, Julien Jomier, and Charles Marion. Itk: enabling reproducible research and open science. *Frontiers in Neuroinformatics*, 8, 2014. URL: <https://www.frontiersin.org/articles/10.3389/fninf.2014.00013>, doi:10.3389/fninf.2014.00013.

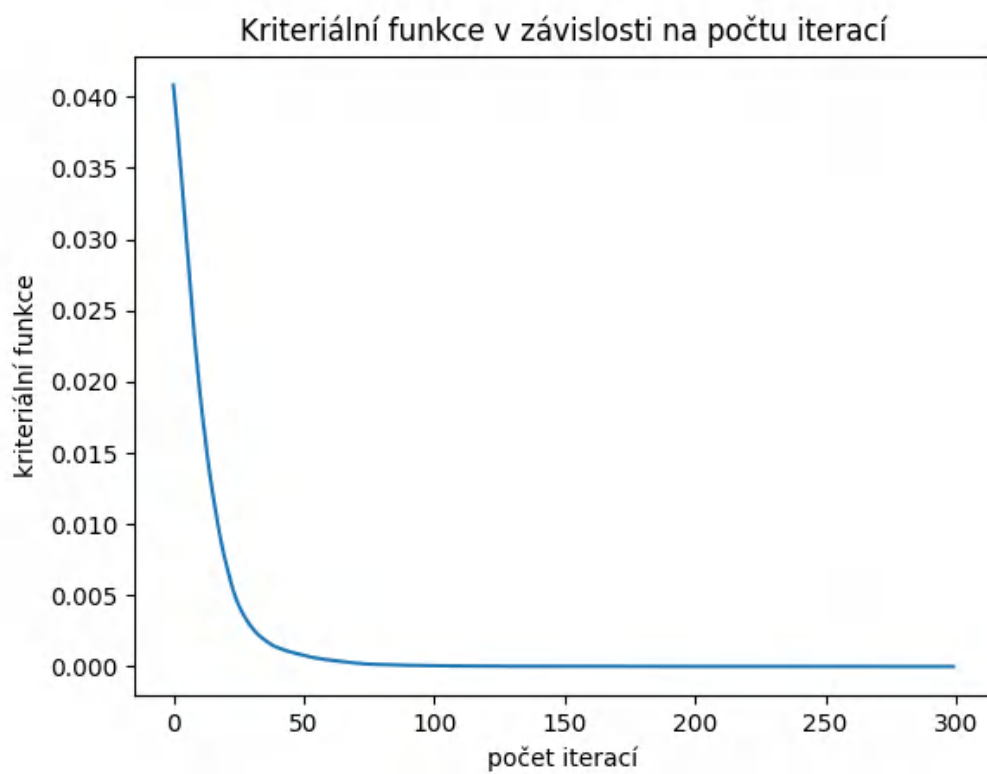
- [18] Stefan Klein, Marius Staring, Keelin Murphy, Max A. Viergever, and Josien P. W. Pluim. elastix: A toolbox for intensity-based medical image registration. *IEEE Trans. Med. Imaging*, 29(1):196–205, 2010. URL: <http://dblp.uni-trier.de/db/journals/tmi/tmi29.html#KleinSMVP10>.
- [19] Kasper Marstal, Floris Berendsen, Marius Staring, and Stefan Klein. Simple-*elastix*: A user-friendly, multi-lingual library for medical image registration. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 574–582. IEEE, 2016.
- [20] J. Shi D. Ponsa F. Moreno-Noguer E. Riba, D. Mishkin and G. Bradski. A survey on kornia: an open source differentiable computer vision library for pytorch. 2020.
- [21] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. 2019.
- [22] Simple itk, 2020. URL: <https://simpleitk.org/>.
- [23] Robin Sandkühler, Christoph Jud, Simon Andermatt, and Philippe C Cattin. Airlab: Autograd image registration laboratory. 2018.
- [24] Airlab. URL: <https://github.com/airlab-unibas/airlab>.
- [25] Stéfan Van Der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. Scikit-image: Image processing in python. *PeerJ (San Francisco, CA)*, 2014(1):e453–e453, 2014.
- [26] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nature (London)*, 585(7825):357–362, 2020.
- [27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. 2019.

- [28] Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.e9, 2018.
- [29] Carlos Hernández Matas. Fire: Fundus image registration dataset. URL: <https://projects.ics.forth.gr/cvrl/fire/>.

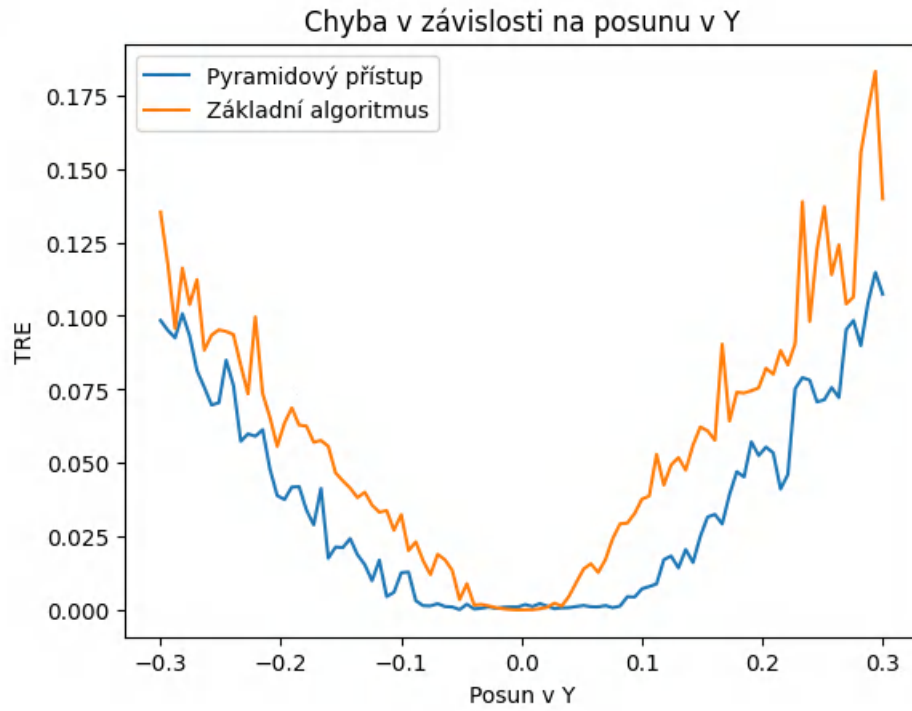
Seznam symbolů a zkratek

CT	počítačová tomografie
MAE	střední absolutní chyba
MI	vzájemná informace
MSE	střední kvadratická chyba
MRI	magnetická rezonance
TRE	chyba registrace daná jako průměrná eukleidovská vzdálenost registrovaných kontrolních bodů od bodů na které se kontrolní body měly registrovat

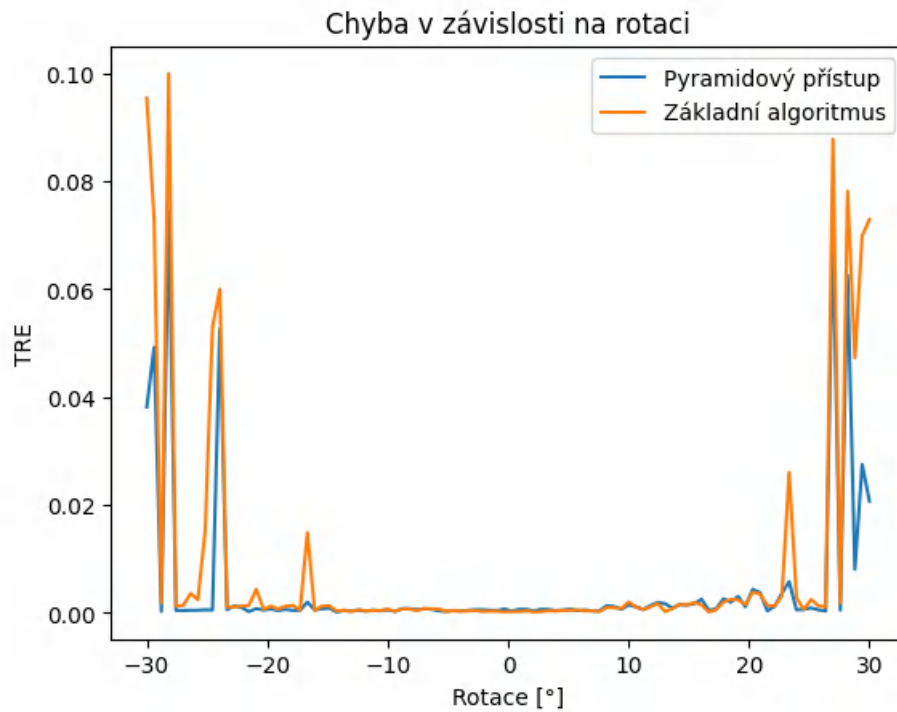
A Grafy testování algoritmu a jeho modifikací



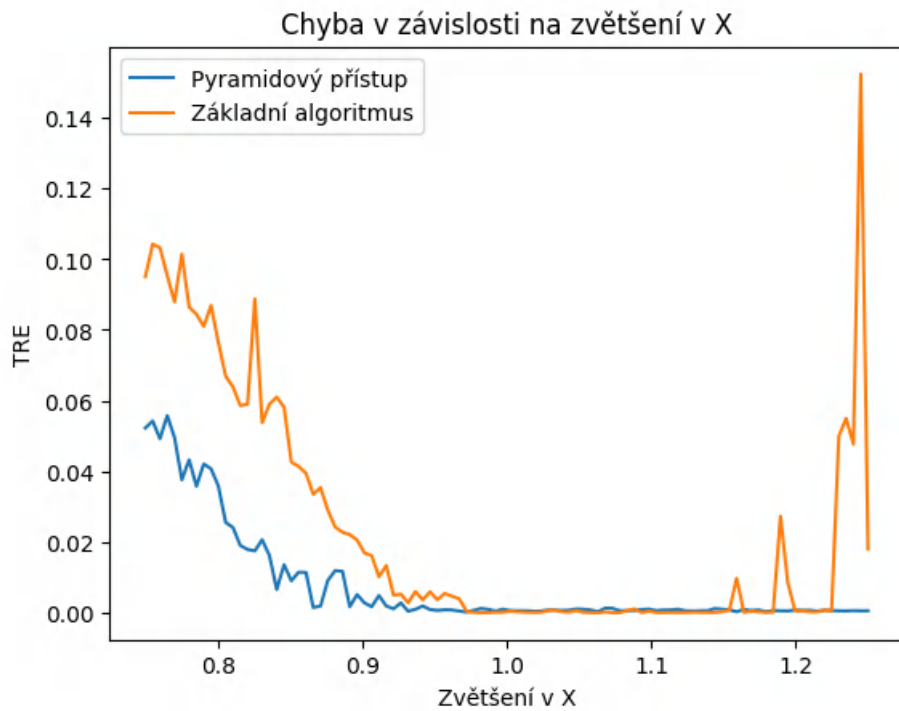
Obr. A.1: Hodnota kriteriální funkce v závislosti na proběhlém počtu iterací pro základní verzi algoritmu



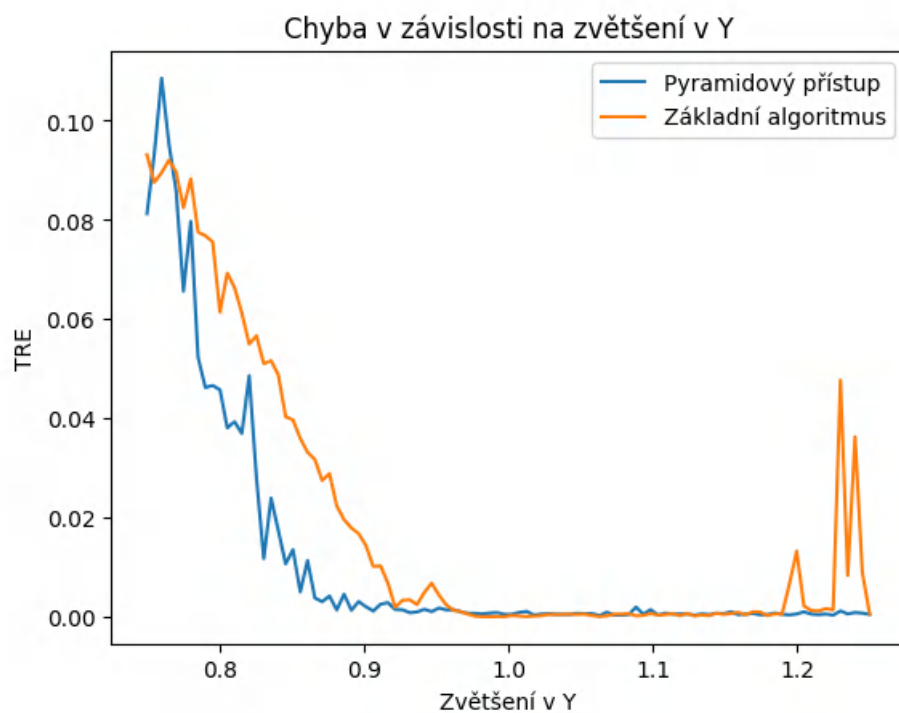
Obr. A.2: Testování posunu v Y



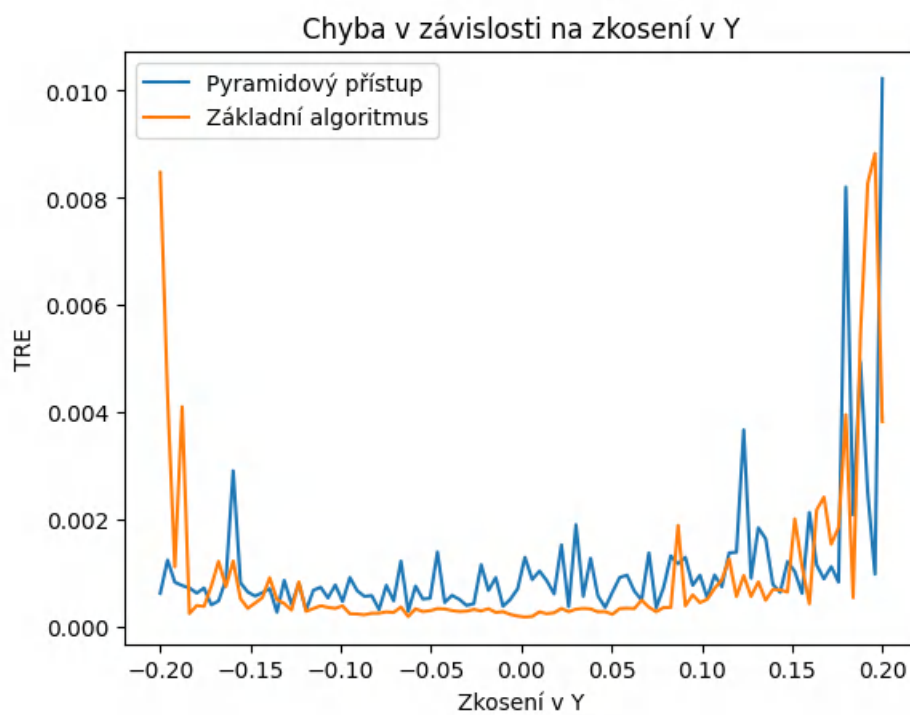
Obr. A.3: Testování rotace



Obr. A.4: Testování zvětšení v X

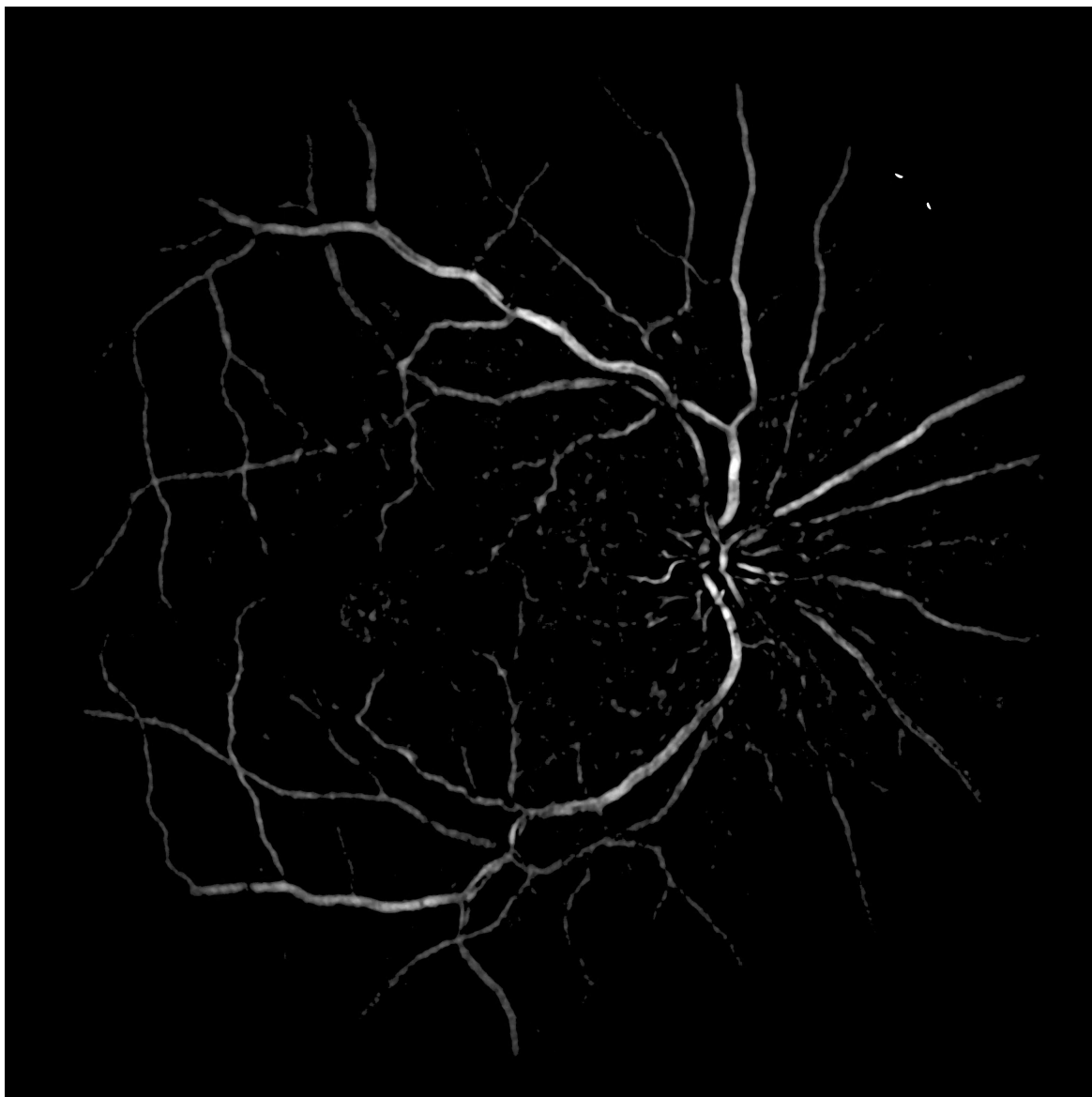


Obr. A.5: Testování zvětšení v Y



Obr. A.6: Testování zkosení v Y

B Ukázka předzpracovaného snímku sítnice



Obr. B.1: Ukázka předzpracovaného snímku. Jedná se o snímek A01_1 z databáze FIRE. [10]