

2019

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

MAGISTRSKO DELO

MAGISTRSKO DELO
**GRAFIČNO RAZVOJNO OKOLJE ZA
PORAVNAVO MEDICINSKIH SLIK Z B-ZLEPKI**

MATJAŽ ŠUBER

MATJAŽ ŠUBER

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Magistrsko delo

**Grafično razvojno okolje za
poravnavo medicinskih slik z B-zlepki**

(Graphical development environment for
B-spline medical image registration)

Ime in priimek: Matjaž Šuber

Študijski program: Računalništvo in informatika, 2. stopnja

Mentor: doc. dr. Peter Rogelj

Koper, maj 2019

Ključna dokumentacijska informacija

Ime in PRIIMEK: Matjaž ŠUBER

Naslov magistrskega dela: Grafično razvojno okolje za poravnavo medicinskih slik z B-zlepki

Kraj: Koper

Leto: 2019

Število listov: 47

Število slik: 17

Število tabel: 4

Število referenc: 20

Mentor: doc. dr. Peter Rogelj

Ključne besede: Poravnava medicinskih slik, netoga geometrijska transformacija, B-zlepki, grafično razvojno okolje

UDK: 004.93:616-073(043.2)

Izvleček:

Poravnava medicinskih slik je zelo pomembna komponenta številnih medicinskih aplikacij, saj omogoča združevanje in izločanje medicinsko pomembne informacijske vsebine slik. V splošnem določa postopek iskanja optimalne geometrijske transformacije, ki preslika prostor ene slike v prostor druge slike, tako, da doseže optimalno prostorsko skladnost anatomskih struktur, ki jih sliki predstavljata. V praksi poznamo dva tipa geometrijskih transformacij. Toge transformacije modelirajo rotacije in premike slikovnih elementov, netoge pa omogočajo modeliranje deformacij objektov na sliki. V magistrski nalogi smo raziskali in implementirali postopek netoge poravnave medicinskih slik z B-zlepki. V namen izdelave učinkovitega postopka poravnave smo v nalogi definirali in implementirali grafično razvojno okolje. Grafično razvojno okolje omogoča hitrejši razvoj, testiranje in analizo različnih postopkov poravnave z minimalnimi začetnimi nastavitvami. Implementirano je v programskem jeziku Matlab, ki je že uveljavljen na tem raziskovalnem področju. Vsebuje grafični uporabniški vmesnik, dobro definirano podatkovno strukturo in veliko število nizko nivojskih funkcij, ki so bile pridobljene v okviru raziskovalnega dela.

Key words documentation

Name and SURNAME: Matjaž ŠUBER

Title of the thesis: Graphical development environment for B-spline medical image registration

Place: Koper

Year: 2019

Number of pages: 47

Number of figures: 17

Number of tables: 4

Number of references: 20

Mentor: Assist. Prof. Peter Rogelj, PhD

Keywords: Medical image registration, non rigid geometric transformation, B-spline, graphical development environment

UDK: 004.93:616-073(043.2)

Abstract: Medical image registration is a very important component of many clinical applications. It can be used for the detection and diagnosis of diseases, for planning the therapy, for guidance of interventions and for the followup and monitoring of patients. In general, the process of image registration involves finding the optimal geometric transformation which maximizes the correspondences across images. Depending on the problem, geometric transformation can model only rotations and scaling of images (rigid transformation) or can include deformations (non rigid transformation). In this work we studied and implemented a B-spline nonrigid medical image registration procedure. In order to provide a high quality implementation we defined and implemented a graphical development environment. The graphical development environment improves the implementation, testing and analysis of different medical image registration procedures. It is built on the top of Matlab programming language that is already well established in this research field. It includes a graphical user interface, a well defined data structure and a large set of lower level processing functions, that were implemented in past research activities.

Zahvala

Zahvaljujem se mentorju doc. dr. Petru Roglju za strokovno pomoč in spodbudo pri izdelavi magistrske naloge. Zahvalil bi se Maji Kralj, mami Bogdani in očetu Andreju, ker so me podpirali na moji študijski poti in mi zmeraj stali ob strani.

Kazalo vsebine

1	Uvod	1
1.1	Lastnosti 3-D medicinskih slik	2
1.2	Postopek za poravnavo medicinskih slik	2
1.2.1	Poravnava	3
1.2.2	Delitev poravnave	4
1.2.3	Mere podobnosti	6
1.2.4	Optimizacija	7
1.3	Izhodišče naloge	9
1.3.1	REG podatkovna struktura	10
1.3.2	REG Toolbox - programska orodjarna	11
1.3.3	B-Spline Toolbox - programska knjižnjica	11
2	Grafično razvojno okolje	13
2.1	Analiza in definiranje zahtev	13
2.2	Arhitektura grafičnega razvojnega okolja	14
2.2.1	Grafični uporabniški vmesnik	14
2.2.2	GUI-API programski vmesnik	17
2.3	Primer uporabe	18
3	Poravnava slik z B-zlepki	22
3.1	Toga geometrijska transformacija	22
3.1.1	Optimizacija toge transformacije	23
3.1.2	Implementacija in integracija z grafičnim razvojnim okoljem	23
3.2	Netoga geometrijska transformacija	24
3.2.1	Optimizacija netoge transformacije	26
3.2.2	Implementacija in integracija z grafičnim razvojnim okoljem	27
3.3	Arhitektura postopka poravnave z B-zlepki	29
4	Rezultati	31
4.1	Rezultati postopka poravnave slik z B-zlepki	31
4.1.1	Rezultati toge geometrijske poravnave	31

4.1.2	Rezultati netoge geometrijske poravnave z B-zlepki	32
5	Zaključek	34
6	Literatura	36

Kazalo tabel

1	Primerjava min., maks. in srednje vrednosti <i>toge geo. trans.</i>	32
2	Prikaz <i>MAD</i> mere podobnosti pred in po togi poravnavi.	32
3	Primerjava min., maks. in srednje vrednosti mreže kontrolnih točk. . .	33
4	Prikaz <i>MI</i> mere podobnosti pred in po netogi poravnavi.	33

Kazalo slik

1	Primer anatomske slikovne modalitete CT, slika levo in funkcijske slikovne modalitete PET, slika desno, slike glave [6].	3
2	Ilustracija poravnave slik.	4
3	Interpolacija delnih volumnov za 2-D [18].	8
4	Gradniki grafičnega razvojnega okolja.	15
5	Grafični uporabniški vmesnik (področja).	16
6	Izsek nastavitvev znotraj konfiguracijske datoteke <i>configureGui.m</i> , za področje <i>Programska orodjarna</i>	17
7	<i>GUI-API</i> modalno okno za izbiro postopka poravnave slik.	18
8	Prikaz MRI slik pred poravnavo (1 rezina).	19
9	Prikaz MRI slik po poravnavi (1 rezina).	19
10	Vsebina datotečnega imenika <i>external</i>	19
11	Prikaz absolutne razlike MRI slik pred poravnavo (1 rezina)	20
12	Prikaz absolutne razlike MRI slik po poravnavi (1 rezina)	20
13	Prikaz mreže kontrolnih točk pred poravnavo (1 rezina)	20
14	Prikaz mreže kontrolnih točk po poravnavi (1 rezina)	20
15	Gradniki postopka poravnave medicinskih slik z B-zlepki.	30
16	Prikaz optimalne mreže kontrolnih točk po poravnavi (1 rezina)	33
17	Prikaz izračunane mreže kontrolnih točk po poravnavi (1 rezina)	33

Seznam kratic

<i>geo.</i>	geometrijska
<i>gui</i>	grafični uporabniški vmesnik (ang. <i>Graphical User Interface</i>)
<i>itd.</i>	in tako dalje
<i>mad</i>	povprečna absolutna razlika (ang. <i>Mean absolute difference</i>)
<i>mi</i>	medsebojna informacija (ang. <i>Mutual Information</i>)
<i>roi</i>	območje zanimanja na sliki (ang. <i>Region Of Interest</i>)
<i>ssd</i>	vsota kvadratov razlik (ang. <i>Sum of squared differences</i>)
<i>trans.</i>	transformacija

1 Uvod

Življenska doba prebivalstva se povečuje in s tem tudi skrb za zdravje. Vedno več finančnih sredstev se vlaga v boljše diagnostične postopke in učinkovitejše zdravljenje. Pri diagnostiki, odkrivanju bolezni, načrtovanju terapij, vodenju medicinskih posegov in spremljanju zdravljenja bolnikov so vse bolj prisotne medicinske slike. Znotraj raziskav je zahtevanih veliko medicinskih slik, ki so lahko zajete v različnem času, z istimi ali različnimi slikovnimi postopki. Vsaka uspešna generacija slikovnih postopkov proizvaja večje količine podatkov. Trend slikovnih postopkov nakazuje naraščanje količine podatkov v medicini, zato je vsaka izboljšava pri obdelavi medicinskih slik močno zaželjena. Informacijska tehnologija v ta namen nudi števila orodja, ki omogočajo vizualizacijo in primerjavo slik. Kritična faza v tem procesu je poravnava ali registracija slik.

Poravnava ali registracija medicinskih slik je postopek, ki temelji na ugotavljanju medsebojne geometrijske odvisnosti dveh ali več slik, z namenom pridobiti medicinsko pomembne informacije o stanju in dogajanju v telesu. Dodatne informacije, pridobljene na ta način, lahko lajšajo in izboljšajo medicinsko diagnostiko in postopke zdravljenja ter prispevajo k boljšemu razumevanju poteka bolezni, delovanja organov in anatomskih ali funkcionalnih sprememb [13].

Postopek poravnave slik lahko uporabljamo v različne namene. Poravnavamo lahko različne vrste slik, ki so zajete v različnem času, z istim (eno-modalna poravnava) ali z različnimi (več-modalna poravnava) slikovnimi postopki. V odvisnosti od dejanskih lastnosti slikane anatomije, lahko geometrijsko soodvisnost med slikami išemo bolj ali manj detajlno. V tem smislu poznamo togo in netogo (deformabilno) poravnavo. Pri togi poravnavi nas zanima le medsebojni premik in rotacija objektov na sliki, pri netogi pa tudi vse deformacije.

Rezultat poravnave slik je geometrijska transformacija T ene ali več slik, s katero želimo doseči, da se iste anatomske strukture na vseh slikah nahajajo na istem mestu, kljub morebitnim razlikam v legi pacienta, postopku zajemanja slik, delovanju organov, patologijam in podobno, kot je prikazano v sliki 2. Poravnavane slike lahko zato enostavno primerjamo, združujemo njihovo informacijsko vsebino ter jih analiziramo, prav tako pa lahko analiziramo tudi geometrijsko soodvisnost med slikami S . Vse to

prispeva k natančnosti, podrobnosti in preglednosti raziskav ter omogoča vpogled v nove razsežnosti dogajanja v telesu.

1.1 Lastnosti 3-D medicinskih slik

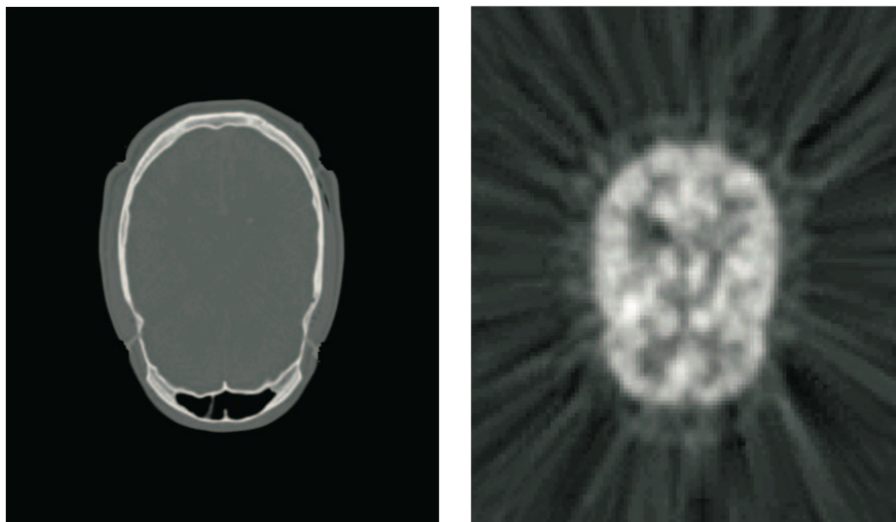
Slikovne tehnike so postale temeljno orodje v medicini. Vsaka izmed njih prikazuje slikani objekt na različne načine. Kljub temu, pa med njimi obstajajo tudi nekatere podobnosti. Medicinske slike so bolj ali manj pošumljene svetlostne slike na temnem ozadju, ki predstavlja okoliški zrak. Na vseh medicinskih slikah je opazen zunanji obris med bolnikom in okolico.

V matematičnem zapisu je svetlostna ali sivinska slika funkcija $f(x, y)$, kjer sta x in y prostorski koordinati v ravnini slike, amplituda f v točki (x, y) pa je svetlost oziroma sivinska vrednost v tej točki [6]. Pri digitalnih slikah so x , y in f diskretne vrednosti. V nadaljevanju naloge se bomo ukvarjali izključno z digitalnimi slikami. V splošnem se v medicini ukvarjamo s trodimenzionalnimi (3-D) slikami. Trodimenzionalne (3-D) slike so funkcije treh prostorskih koordinat $f(x, y, z)$ in so sestavljene iz končnega števila volumskih elementov, imenovanih **vokslov**, ki se nahajajo v točkah (x, y, z) slikovnega polja. Zaloga vrednosti f je razpon dopustnih svetlostnih vrednosti na sliki. Na primer, slike, ki so kodirane v 8-bitnem zapisu, dopuščajo svetlostne vrednosti v razponu od 0 (črna) do $2^8 - 1$ (bela).

Medicinske slike dobimo z uporabo različnih slikovnih postopkov/tehnik/senzorjev, ki zaznavajo določene lastnosti, ki jih želimo predstaviti v slikovni obliki. Slikovne tehnike v medicini delimo v dve večji kategoriji: anatomske in funkcionalne tehnike. Anatomske slikovne tehnike predstavljajo anatomsko informacijo o bolniku, kot na primer položaj in geometrijsko razsežnost organov in tkiv. Primeri anatomskih modalitet so magnetno-resonančno slikanje (MRI), računalniška tomografija (CT), rentgen in ultrazvok (UZ). Funkcionalne modalitete prikazujejo funkcionalno informacijo organov, kot na primer aktivnost možganov pri določeni nalogi. Primeri funkcionalnih slikovnih tehnik so funkcionalna magnetna resonanca (fMRI), pozitronska izsevana tomografija (PET) in enofotonska izsevana tomografija (SPECT). Na sliki 1 sta prikazana primera anatomske in funkcijske slikovne tehnike.

1.2 Postopek za poravnavo medicinskih slik

V tem poglavju so opisani osnovni pojmi, ki tvorijo koncept postopkov za poravnavo medicinskih slik in se bodo pojavljali v nadaljevanju naloge.



Slika 1: Primer anatomske slikovne modalitete CT, slika levo in funkcijske slikovne modalitete PET, slika desno, slike glave [6].

1.2.1 Poravnava

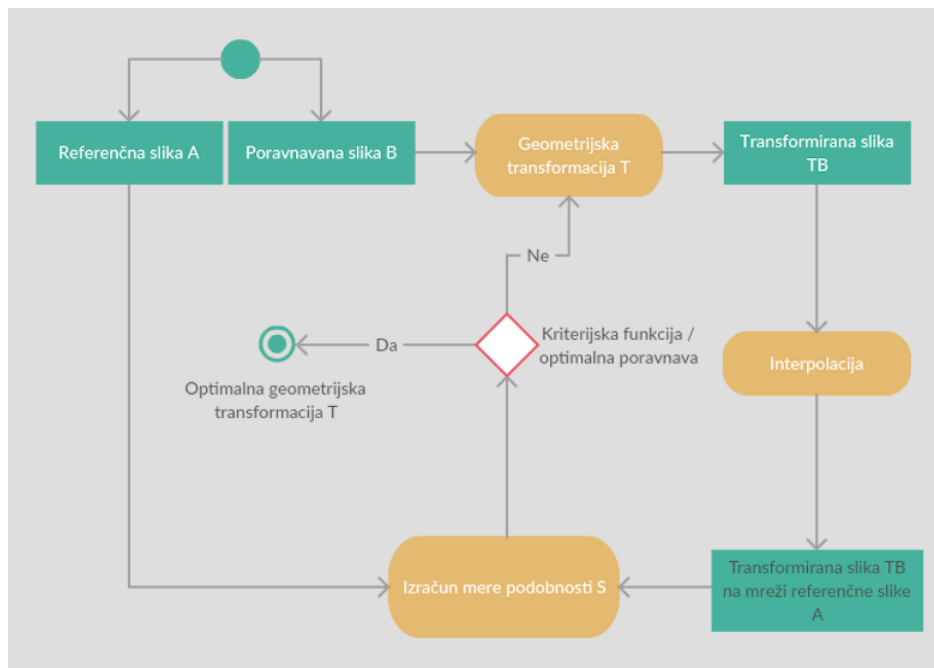
Naj bosta A in B medicinski sliki, ki jih želimo med seboj poravnati. Iskanje geometrijske transformacije T , ki preslika eno sliko v najboljšo lego glede na drugo sliko, se imenuje poravnava. Poravnavo lahko definiramo tudi kot iskanje take geometrijske preslikave T , ki prostor ene slike (poravnavana slika) preslika v prostor druge slike (referenčna slika), tako da točke, ki pripadajo istim anatomskim strukturam ležijo na istih koordinatah. Osnovni koncept poravnave je prikazan na sliki 2.

Sliko si lahko predstavljamo kot preslikavo točk, na slikovnem področju Ω , v svetlostne vrednosti [6], to je:

$$\begin{aligned} A &: x_A \in \Omega_A \rightarrow A(x_A) \\ B &: x_B \in \Omega_B \rightarrow B(x_B) \end{aligned}$$

Medicinske slike imajo praviloma različna slikovna polja, zato sta Ω_A in Ω_B različna. Naj bo O objekt, ki je upodobljen na sliki A in na sliki B . Tako je lega anatomske točke $x \in O$ definirana z x_A na sliki A in z x_B na sliki B . S poravnavo slik iščemo tako geometrijsko transformacijo T , ki preslika x_B na x_A na področju prekrivanja oziroma na preseku referenčne in poravnavane slike. Bolj natančno, transformacija T preslika Ω_B v Ω_A znotraj področja prekrivanja $\Omega_{A,B}^T$, ki je definirano:

$$\Omega_{A,B}^T = \{x_A \in \Omega_A : T^{-1}(x_A) \in \Omega_B\} \quad [6]$$



Slika 2: Ilustracija poravnave slik.

Zgornja enačba določa odvisnost področja prekrivanja na originalnih slikah A in B od transformacije T .

Algoritmi za poravnavo medicinskih slik iščejo tisto geometrijsko transformacijo T , ki izpolni oziroma optimizira kriterij o ujemanju prostorov. Kriterij o ujemanju prostorov je mera podobnosti, ki opisuje prostorsko ujemanje anatomskih struktur na slikah. Cilj postopka poravnave je najti takšno geometrijsko transformacijo T , ki optimizira dani kriterij. Postopke za poravnavo medicinskih slik lahko razdelimo glede na prostorsko domeno, naravo značilnic na katerih temelji poravnava, naravo geometrijskih preslikav, naravo interakcij z uporabnikom, glede na postopke optimizacije, vrsto (modaliteto) slik in glede na osebek in objekt poravnave. Podrobna obravnava in razvrstitev postopkov poravnave medicinskih slik je opisana v [7], glavne delitve pa bomo povzeli v nadaljevanju.

1.2.2 Delitev poravnav

Poravnave lahko delimo glede na **razsežnosti slik**, ki jih poravnavamo. Ločimo dvodimenzionalno (2-D) in trodimenzionalno (3-D) poravnavo, ki sta najpogostejši. Možna pa je tudi poravnava dvodimenzionalnih slik na trodimenzionalne slike in obratno, v primerih, ko je 3-D informacija projicirana na 2-D sliko (na primer rentgensko slika-

nje) [6]. V nalogi smo obravnavali le z 3-D poravnavo.

Glede na **naravo geometrijske transformacije** ločimo *togo* in *netogo (deformabilno)* poravnavo. Pri togi poravnavi, je transformacija sestavljena iz premika in zasuka objektov na sliki. Za togo transformacijo je značilno, da ohranja razdaljo med poljubnima dvema točkama na sliki. Tovrstno transformacijo lahko za dvodimenzionalne slike opišemo s tremi parametri (premik v x in y smeri ter zasuk), za trodimenzionalne slike pa s šestimi parametri (trije premiki, trije zasuki). Netoge transformacije imenujemo tudi nelinearne in jih včasih modeliramo z uporabo fizikalnih lastnosti, kot so gibanje tekočin, elastičnost, viskoznost in podobno. V nalogi se bomo ukvarjali z togo in netogo (deformabilno) geometrijsko transformacijo.

Naslednja delitev postopkov poravnave medicinskih slik je glede na **modalnost (vrsto)** uporabljenih slik. Če poravnavamo slike zajete z istim slikovnim postopkom (tehniko), tako poravnavo imenujemo enomodalna poravnava. Večmodalno poravnava pa je poravnava, kjer so uporabljene slike zajete z različnimi slikovnimi postopki. Slike različnih modalnosti izhajajo iz meritev različnih lastnosti tkiv ali fizioloških pojavov v telesu. Poravnava takšnih slik je še posebej zahtevna, saj relacije med merjenimi lastnostmi v splošnem niso znane. Vsaka modalnost ima lahko drugačen medicinski pomen. To informacijo lahko združimo s postopki večmodalne poravnave. Primer večmodalne poravnave je poravnava CT (ang. *Computer Tomography*) in PET (ang. *Positron Emission Tomography*). CT slike dobro opisujejo anatomsko zgradbo telesa, PET slike pa prikazujejo fiziološko aktivnost telesnih struktur. S poravnavo obeh vrst slik pridobimo znanje o aktivnosti anatomskih struktur. V nalogi se bomo osredotočili na poravnavo enomodalnih slik.

Postopke poravnave lahko delimo tudi glede na uporabljene **značilnice** pri ocenjevanju poravnave slik. Poravnava slik ocenjujemo s pomočjo kriterijske funkcije, ki "meri" podobnost med danima slikama. Merjenje podobnosti je kompleksen postopek odvisen od postopka optimizacije, mere podobnosti in *značilnic*. Poznamo *svetlostne, geometrijske in teksturne značilnice*.

Svetlostne značilnice so najpogosteje zastopane v problemih poravnave slik. V nasprotju z geometrijskimi značilnicami, ne potrebujejo predhodne segmentacije in so zato lahko povsem avtomatizirane. Poravnava z uporabo svetlostnih značilnic je realizirana kot optimizacijski proces, ki na osnovi ocenjevanja skladnosti svetlostnih nivojev med poravnavano in referenčno sliko poišče optimalno preslikavo poravnave slike. Slaba stran uporabe svetlostnih značilnic je pomanjkanje strukturne informacije na slikah, ki je za netogo poravnavo lahko zelo pomembna. Neprimernost uporabe svetlostnih značilnic je izrazita pri poravnavi slik slabe kvalitete, močno pošumljenih slik, kot opisano v [6].

Pomanjkljivosti svetlostnih značilnic se izognemo z uporabo geometrijskih značilnic, ki po vsebini predstavljajo strukturno oziroma anatomsko informacijo, ki je na slikah očitna in jo je smiselno uporabiti v postopku poravnave. Geometrijske značilnice temeljijo na anatomsko pomembnih točkah, krivuljah, površinah oziroma področjih, ki so lahko določene ročno ali avtomatsko. V tem primeru je naloga poravnave najti preslikavo, ki ustreza najboljšemu medsebojnemu prileganju geometrijskih značilnic na obeh slikah. Kvaliteta poravnave je tako odvisna od točnosti in števila uporabljenih značilnic, zato je določanje le-teh bistven korak pri poravnavi. Avtomatsko izločanje geometrijskih značilnic medicinskih slik je težavno zaradi kompleksne geometrije, ročno označevanje geometrijskih značilnic pa izredno zamudno.

Dobra alternativa svetlostnim in geometrijskim značilnicam so teksturne značilnice. Uporaba teksturnih značilnic za namen poravnave slik sodi med novejšje pristope in je podrobneje opisana v [6]. V nalogi bomo za merjene podobnosti uporabili svetlostne značilnice, ker so najbolj zastopane in ne potrebujejo predhodne segmentacije.

1.2.3 Mere podobnosti

Ocenjevanje podobnosti med referenčno in poravnavano sliko je zelo pomemben korak vsakega postopka poravnave. Izbor ustrezne mere podobnosti ni trivialen problem. Ustreznost izbire je odvisna od transformacijskega modela, modalnosti in kvalitete slik, velikosti področja prekrivanja, velikosti slik in drugo. V nadaljevanju bomo na kratko predstavili najbolj uporabljene enomodalne in večmodalne mere podobnosti.

Enomodalne mere podobnosti so uporabljene za merjenje podobnosti med slikama iste modalitete. Te mere ocenijo predvsem linearno ali konstantno ujemanje svetlostnih vrednosti ujemaajočih se slikovnih elementov na obeh slikah, kot opisano v [6]. Zelo pogosti enomodalni meri sta povprečna absolutna razlika MAD (ang. *Mean absolute difference*) in vsota kvadratov razlik SSD (ang. *Sum of squared differences*). Naj bo N število slikovnih elementov na področju $\Omega_{A,B}^T$, potem velja:

$$MAD = \frac{1}{N} \sum_{x_A \in \Omega_{A,B}^T} |A(x_A) - B^T(x_A)| \quad [6],$$

$$SSD = \frac{1}{N} \sum_{x_A \in \Omega_{A,B}^T} (A(x_A) - B^T(x_A))^2 \quad [6],$$

Obe meri sta normirani, kar pomeni da sta neobčutljivi na število slikovnih elementov na področju prekrivanja $\Omega_{A,B}^T$. Optimizacijski algoritmi pri teh dveh merah iščejo minimum, kar pomeni da sta si sliki čedalje bolj podobni, če se razlika med njima manjša oziroma je enaka 0.

Večmodalne mere podobnosti morajo biti za razliko od enomodalnih neobčutljive na razlike v svetlosti in kontrastu slik. Dopuščajo povsem poljubne medsebojne relacije med svetlostnimi nivoji slik. Za razliko od slik istih modalitet, slike različnih modalitet prikazujejo skupno in hkrati tudi komplementarno informacijo o istem prikazanem objektu, vendar v različnih svetlostnih nivojih. Najbolj pogosto uporabljena večmodalna mera podobnosti je medsebojna informacija MI (ang. *Mutual Information*). Izhaja iz teorije informacij, na področje obdelave medicinskih slik pa sta jo vpeljala Viola in Wells leta 1995 [19]. To je statistična mera podobnosti in ocenjuje statistično odvisnost med svetlostnimi nivoji obeh slik. Na ta način poda oceno, koliko ena slika pove o drugi sliki in zavzame največjo vrednost, ko sta sliki geometrijsko poravnani. Medsebojno informacijo izračunamo na sledeči način:

$$MI(A, B) = H(A) + H(B) - H(A, B) [19]$$

kjer sta $H(A)$ in $H(B)$ Shannonovi entropiji značilnic posamezne slike A in B , $H(A, B)$ pa določa vezano entropijo značilnic na obeh slikah hkrati. Entropijo $H(\cdot)$ izračunamo kot:

$$H(\cdot) = - \sum_i p(i) \times \log_2 p(i)$$

kjer je p verjetnostna porazdelitev značilnic na sliki. V nalogi bomo za merjene podobnosti medicinskih slik uporabili medsebojno informacijo, saj se je v številnih študijah izkazala za zanesljivo mero.

1.2.4 Optimizacija

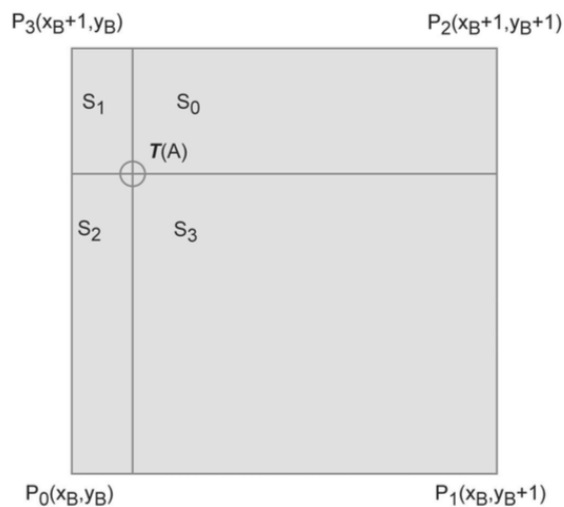
Za večino metod poravnave velja, da metoda optimizacije išče optimum kriterijske funkcije, ki meri podobnost med slikama v vsakem koraku transformacije. Za vsako transformacijo T , transformiramo sliko B in jo nato prevzorčimo v prostor referenčne slike A . Prevzorčenje transformirane slike TB zahteva interpolacijo svetlostnih vrednosti v novo nastalih legah točk na sliki A . Interpolacijske metode imajo praviloma velik vpliv na rezultat poravnave, saj vsaka interpolacijska metoda vnese določene nepravilnosti, hkrati pa povzroči glajenje slike do določene stopnje, kot opisano v [6]. V praksi se najpogosteje uporabljajo naslednje interpolacijske metode:

1. *Linearna interpolacija* (ang. *linear interpolation*): je hitra metoda, primerna za poravnavo, ki zahteva veliko korakov optimizacije. Določanje neznanih svetlostnih vrednosti transformirane slike TA , temelji na interpolaciji znanih svetlo-

stnih vrednosti slikovnih elementov slike A . V splošnem lahko linearno interpolacijo zapišemo z izrazom $C = A * (1 - \alpha) + B * \alpha$, kjer C določa svetlostno vrednost transformirane točke, A in B svetlostne vrednosti znanih točk, α pa interpolirano vrednost na linearni premici med točkama A in B .

2. *Interpolacija delnih volumnov (ang. partial volume interpolation - PVI)*: interpolacija delnih volumnov v primeru 2-D slik je prikazana na sliki 3. $T(A)$ predstavlja transformirano koordinato točke na sliki A , vrednosti P_0 do P_3 pa vrednosti štirih slikovnih elementov med katerimi leži točka $T(A)$. S_0 do S_3 so površine, na katere transformirana točka $T(A)$ razdeli površino med štirimi sosednjimi vokslami mirujoče slike A . Vsakemu paru točk $(P_i, I(A))$ v histogramu prištejemo vrednost nasprotne površine. $I(A)$ določa svetlost tiste točke, ki jo trenutno transformiramo. V primeru 3-D slik, namesto s površinami operiramo z volumni.

$$(P_i, I(A))_+ = S_i, \forall_i [18]$$



Slika 3: Interpolacija delnih volumnov za 2-D [18].

Največji problem optimizacijskih metod je, da končna transformacija ne ustreza pravilni poravnavi med slikama, saj se optimizacija lahko ustavi pri enem od številnih lokalnih ekstremov kriterijske funkcije. Po drugi strani pa ni zagotovila, da pravilna rešitev leži v globalnem optimumu funkcije, kot je to opisano v [6]. Postopki poravnave imajo lahko v nekaterih primerih veliko število parametrov geometrijske transformacije, tako da je parametričen prostor za optimizacijo precej razsežen. To pomeni, da je iskanje rešitve takega problema časovno in računsko zelo zamuden postopek. Po drugi

strani pa naj bi bila poravnava slik v medicini, v idealnem primeru, izvedena v realnem času. To zahteva od algoritmov, ki jih razvijamo, da so računsko učinkoviti, stabilni in robustni.

V medicinski praksi se verjetno večina poravnav še vedno izvaja s pomočjo togih geometrijskih transformacij, vendar se potreba po netogi poravnavi stopnjuje. Netogo poravnavo lahko uporabimo za poravnavo slik, ki so deformirane zaradi napak pri zajemu, časovnega spreminjanja anatomije kot posledice staranja ali delovanja zdravil in drugo. Ene izmed najbolj uveljavljenih metod za netogo poravnavo medicinskih slik temeljijo na matematični krivulji imenovani zlepek. Pojem zlepek izvira iz dolgih gibljivih lesenih ali kovinskih trakov, ki so bili uporabljeni za modeliranje površin ladij ali letal. Metode poravnave, ki temeljijo na zlepkih, predpostavljajo, da lahko na slikah določimo točke imenovane kontrolne točke. Zlepki poskrbijo, da se kontrolna točka na eni sliki preslika v kontrolno točko na drugi sliki. Med kontrolnimi točkami pa zlepki poskrbijo za gladko interpolacijo novih položajev. Najbolj znana vrsta zlepkov, ki je uporabljena v postopkih poravnave medicinskih slik so **B-zlepki**.

1.3 Izhodišče naloge

Obstajajo številni uveljavljeni postopki za netogo poravnavo medicinskih slik. Velika večina takih postopkov je vključena v kompleksna programska orodja, ki po večini zahtevajo veliko začetnih adaptacij in učenja. Zelo težko je najti razvojna okolja, ki bi na enostaven način omogočala analizo, uporabo in razširitev znanih postopkov v namen pohitritve njihovega delovanja. To je tudi glavni povod magistrskega dela. V nalogi želimo implementirati razvojno okolje, ki bo na enostaven način omogočalo analizo, integracijo, razširitev in razvoj številnih postopkov poravnave medicinskih slik. Kot dokaz uporabnosti in pravilnosti delovanja razvojnega okolja, bomo v njem implementirali znani postopek za netogo poravnavo medicinskih slik z B-zlepki.

Izhodišče naloge določajo naslednje komponente. Prva komponenta je dobro definirana podatkovna struktura, ki določa osnovni nabor podatkov, ki so potrebni pri vsaki poravnavi medicinskih slik. Druga komponenta je skupek nizko nivojskih programskih funkcij, ki so bile implementirane v predhodnih raziskavah doc. dr. Peter Rogelj. Tretja komponenta je nabor programskih funkcij, ki vsebuje osnovno implementacijo postopka za netogo poravnavo medicinskih slik z B-zlepki, avtorja Dirk-Jan Kroon.

1.3.1 REG podatkovna struktura

V namen manipulacije različnih postopkov poravnave medicinskih slik na enovit in strukturiran način je bila v predhodnih raziskavah doc. dr. Peter Rogelj določena podatkovna struktura, imenovana *REG podatkovna struktura*. Ta je organizirana v obliki polja, kjer elementi določajo medicinsko sliko in pripadajoče metapodatke. Te lahko po potrebi dodajamo, dokler ne zadostimo zahtevam postopka poravnave. V magistrski nalogi smo opisano podatkovno strukturo vključili v grafično razvojno okolje, kot edini način manipulacije z medicinskimi slikami in rezultati poravnave.

Spodaj so opisani elementi podatkovne strukture:

RefIdx: Indeks slike v polju, ki določa referenčno sliko.

MovIdx: Indeks slike v polju, ki določa poravnavano sliko.

Img[]: Polje, ki shranjuje medicinske slike in pripadajoče podatke.

Img[i]: Struktura, ki določa medicinsko sliko na i -tem mestu v polju in pripadajoče podatke.

Img[i].name: Ime medicinske slike.

Img[i].path: Absolutna pot do medicinske slike.

Img[i].voxelSize: Velikost slikovnih elementov/vokslov medicinske slike.

Img[i].data_orig: Podatki slike, ki niso podvrženi obdelavi.

Img[i].data: Podatki slike, ki se uporabljajo v postopku poravnave slik in so pretvorjeni v *uint8* format (pretvorba določa kompromis med zahtevano slikovno resolucijo in signifikanco rezultatov poravnave slik);

Img[i].roi: Koordinate, ki določajo območje zanimanja na sliki.

Img[i].O: Izhodišče koordinatnega sistema medicinske slike v milimetrih.

Img[i].T: 3-D globalna geometrijska transformacija, definirana kot 4x4 matriki.

Img[i].D: Polje deformacij, ki omogoča opis netoge (elastične) geometrijske transformacije. Določeno je kot 3-D polje premikov vseh slikovnih elementov/vokslov.

1.3.2 REG Toolbox - programska orodjarna

Postopki poravnave medicinskih slik sestojijo iz številnih komponent. Implementacija le-teh zahteva veliko znanja in truda. V izogib razvoju skupnih gradnikov različnih postopkov poravnave slik, smo v okviru magistrskega dela zbrali kopico v najprej pripravljenih nizko nivojskih funkcij, ki so bile razvite v predhodnih raziskavah doc. dr. Peter Rogelj. Omenjene funkcije so združene v programsko orodjarno imenovano *REG Toolbox*. Ta je dostopna na spletnem naslovu <https://github.com/progelj/REG-toolbox> pod GNU GPL licenco. V magistrski nalogi smo za namen implementacije postopka poravnave z B-zlepki iz REG Toolbox programske orodjarne uporabili naslednje funkcije:

- *checkData.*: Matlab skripta, ki preveri pravilnost podatkov v REG podatkovni strukturi.
- *im2uint8.m*: Matlab skripta, ki pretvori vrednosti slikovnih elementov originalne slike v *uint8* format.
- *loadDicomVolume.m*: Matlab skripta, ki omogoča nalaganje medicinskih slik iz DICOM datotečnega formata.
- *resampleMov2Ref.m*: Matlab skripta, ki prevzorči prostor poravnavane slike na prostor referenčne slike.
- *pvi.m*: Matlab skripta, ki izračuna interpolacijo delnih volumnov.
- *rigid.m*: Matlab skripta, ki izračuna togo geometrijsko transformacijo T, na podlagi parametrov premika in rotacij.

1.3.3 B-Spline Toolbox - programska knjižnjica

B-spline Grid, Image and Point based Registration je programska knjižnjica implementirana v programskem okolju Matlab, avtorja Dirk-Jan Kroon. Dostopna je na spletnem naslovu: <https://www.mathworks.com/matlabcentral/fileexchange/20057-b-spline-grid-image-and-point-based-registration>.

Iz omenjene programske knjižnjice smo prevzeli različico implementacije postopka za poravnavo medicinskih slik z B-zlepki, ki temelji na raziskavi objavljeni v [15]. Namen magistrske naloge je raziskati in uporabiti omenjeno implementacijo v grafičnem

razvojnem okolju. Da bomo zadostili zahtevam grafičnega razvojnega okolja bomo obstoječo implementacijo spremenili tako, da bo temeljila na REG podatkovni strukturi ter jo bo tako mogoče uporabiti znotraj nastalega grafičnega uporabniškega vmesnika. Določene gradnike obstoječe implementacije netovega postopka bomo zamenjali z nizko nivojskimi funkcijami iz REG programske orodjarne. Ker so funkcije iz programske orodjarne implementirane v programskem jeziku C++ [4] si obetamo tudi izboljšave časovne kompleksnosti obstoječega algoritma.

2 Grafično razvojno okolje

V namen izdelave visoko zanesljivih in učinkovitih postopkov poravnave medicinskih slik, raziskovalci potrebujejo razvojna orodja, ki poenostavijo implementacijo, testiranje in analizo rezultatov poravnave. Taka orodja morajo omogočati delo s 3-D medicinskimi slikami, kot na primer prikaz posameznih rezin, prikaz statistike slik z vezanim histogramom, prikaz absolutne razlike slik, itd. Vsebovati morajo orodja, ki omogočajo vizualizacijo in interpretacijo rezultatov poravnave.

Implementacija in testiranje različnih postopkov poravnave medicinskih slik je trenutno možna s pomočjo programov, kot so 3DSlicer [5] in ImageJ [1], programskih knjižnic ITK [2] in VTK [16] ali programskih okolij kot je Matlab [17] in Python [14]. Glavni problem večine omenjenih možnosti je daljša uporabniška adaptacija in učenje njihove uporabe.

V nadaljevanju naloge bomo analizirali in opisali implementacijo grafičnega razvojnega okolja za Matlab, ki poenostavi, pohitri in pripomore k razvoju učinkovitih postopkov poravnave medicinskih slik. Glavna prednost, ki jo prinaša grafično razvojno okolje je celovita rešitev za implementacijo različnih postopkov poravnave medicinskih slik in preprosta začetna namestitvev ter uporaba.

2.1 Analiza in definiranje zahtev

Grafično razvojno okolje je namenjeno študentom in raziskovalcem, ki si želijo preprostega in zanesljivega programskega okolja za analizo, razvoj in testiranje različnih postopkov poravnave medicinskih slik. V ta namen mora grafično razvojno okolje zadostiti nekaterim funkcionalnim zahtevam. Grafično razvojno okolje mora biti implementirano v programskem jeziku Matlab, saj je že dobro uveljavljen na tem raziskovalnem področju. Omogočati mora centraliziran, strukturiran in dobro definiran način shranjevanja medicinskih slik, pripadajočih metapodatkov ter rezultatov poravnave. Vsebovati mora nabor orodij in programskih funkcij, ki omogočajo hitrejši razvoj postopkov poravnave. Med slednje sodijo funkcije, ki omogočajo branje medicinskih slik različnih formatov, kot so DICOM [11], BrainWeb [3] in RIRE (ang. Retrospective Image Registration Evaluation project) [20]. Nabor predpripravljenih funkcij mora vse-

bovati funkcije za izračun različnih mer podobnosti, različnih načinov interpolacije ter vsesplošnega preverjanja pravilnosti podatkov v podatkovni strukturi. Za lažje delo z medicinskimi slikami mora vsebovati grafični uporabniški vmesnik. Grafično razvojno okolje mora biti zasnovano tako, da na enostaven način omogoča integracijo novih postopkov poravnave. Slednji morajo biti dostopni preko grafičnega uporabniškega vmesnika, kot tudi preko Matlab-ove ukazne vrstice. Zaradi zahteve po enostavni namestitvi mora grafično razvojno okolje delovati na podlagi namestitvene datoteke, ki omogoča popolno kontrolo nad delovanjem vseh komponent razvojnega okolja.

2.2 Arhitektura grafičnega razvojnega okolja

Grafično razvojno okolje bo implementirano v programskem jeziku Matlab. Za shranjevanje medicinskih slik in rezultatov poravnave bo v grafično razvojno okolje integrirana REG podatkovna struktura, ki je bila opisana v poglavju 1.3.1. V splošnem bo grafično razvojno okolje sestavljeno iz naslednjih gradnikov.

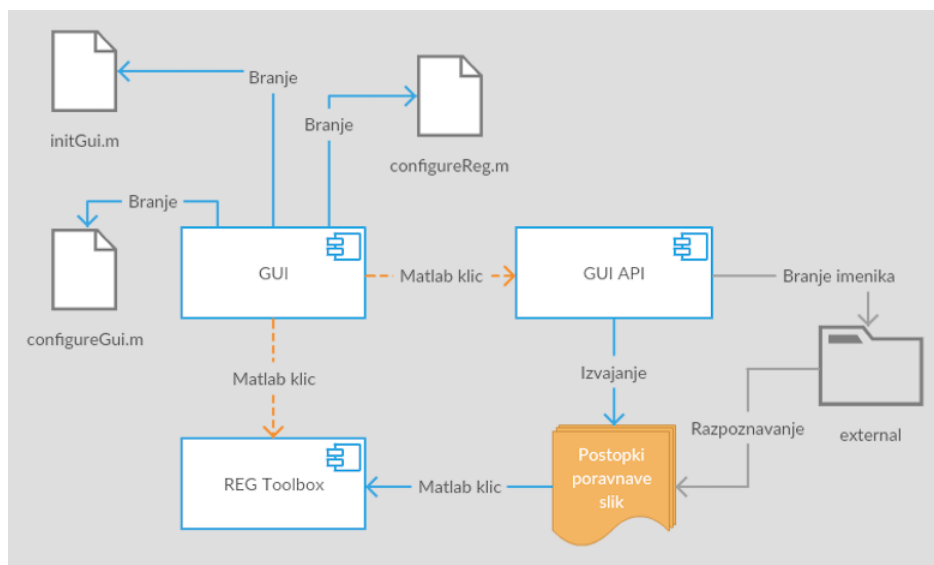
Prvi gradnik bo vseboval programsko orodjarno imenovano *REG-Toolbox*, ki je bila pridobljena v okviru predhodnih raziskav, kot opisano v poglavju 1.3.2. Uporaba omejenih orodjarne bo možna preko Matlab-ove ukazne vrstice in grafičnega uporabniškega vmesnika.

Drugi gradnik grafičnega razvojnega okolja bo grafični uporabniški vmesnik z oznako *GUI*. Implementiran bo s pomočjo Matlab-ovega dodatka, imenovanega Matlab GUI-IDE [9]. Ta na enostaven način omogoča razvoj uporabniških vmesnikov v Matlab okolju. Uporabniški vmesnik grafičnega razvojnega okolja bo v splošnem razdeljen na štiri področja. Prvo področje bo določalo programski meni, drugo področje bo določalo programsko orodjarno, tretje področje bo namenjeno vizualizaciji referenčne slike in četrto področje pa vizualizaciji poravnavane slike. Grafični uporabniški vmesnik bo možno prilagajati s pomočjo konfiguracijske datoteke.

Tretji gradnik grafičnega razvojnega okolja bo programski vmesnik imenovan *GUI API*, ki bo omogočal dodajanje in izvajanje različnih postopkov poravnave medicinskih slik preko grafičnega uporabniškega vmesnika. V nadaljevanju bomo podrobneje opisali razvoj zgoraj naštetih gradnikov grafičnega razvojnega okolja.

2.2.1 Grafični uporabniški vmesnik

Eden izmed glavnih gradnikov grafičnega razvojnega okolja je **grafični uporabniški vmesnik**, ki poenostavi delo z medicinskimi slikami in različnimi postopki poravnave.



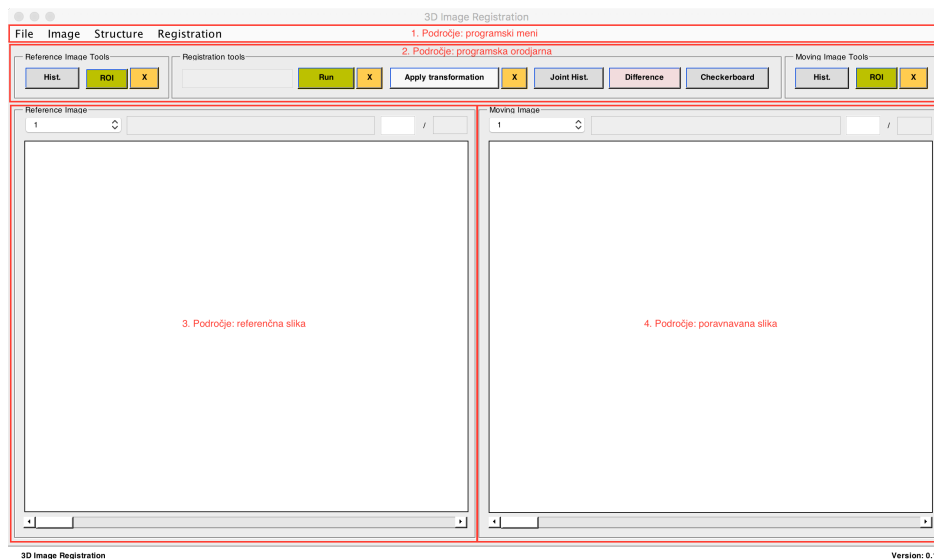
Slika 4: Gradniki grafičnega razvojnega okolja.

Implementiran je s pomočjo Matlab-ovega programskega dodatka, imenovanega Matlab GUIDE [9]. Ta na enostaven način omogoča razvoj uporabniških vmesnikov v Matlab okolju.

Inicializacija in nastavitve delovanja grafičnega uporabniškega vmesnika je določena s pomočjo naslednjih konfiguracijskih datotek.

- *initGui.m*: Matlab skripta, ki skrbi za inicializacijo in zagon grafičnega uporabniškega vmesnika.
- *configureGui.m*: Matlab skripta, ki vsebuje nastavitve vseh funkcij, ki jih lahko izvajamo v grafičnem uporabniškem vmesniku. Nastavitve posamezne funkcionalnosti je določena z relativno potjo do Matlab skripte, ki vsebuje implementacijo funkcionalnosti. Takšen arhitekturni vzorec uporabnikom dopušča možnost, da povsem prilagodijo funkcionalnosti vmesnika.
- *configureReg.m*: Matlab skripta, ki skrbi za pravilno inicializacijo REG podatkovne strukture.

Osnovni uporabniški vmesnik je določen z regijami ali s področji. Regija ali področje predstavlja logični prostor, ki združuje sorodne funkcionalnosti. Področja grafičnega uporabniškega vmesnika so *Programski meni*, *Programska orodjarna*, *Referenčna slika* in *Poravnavana slika*, kot prikazano na sliki 5.



Slika 5: Grafični uporabniški vmesnik (področja).

Programski meni uporabniku omogoča nalaganje 3-D medicinskih slik v različnih formatih, nalaganje in shranjevanje podatkovne strukture in izbiro postopkov poravnave, ki jih je razpoznal *GUI-API* programski vmesnik. Nalaganje 3-D medicinskih slik je možno preko menija *Image*. Uporabnik lahko naloži slike, ki so v DICOM [11], BrainWeb [3] in RIRE (ang. Retrospective Image Registration Evaluation project) [20] datotečnem formatu. Upravljanje s podatkovno strukturo opisano v poglavju 1.3.1 je možno iz menija *Structure*. Grafični uporabniški vmesnik omogoča nalaganje podatkovne strukture iz datotečnega sistema ali neposredno iz Matlab izvajalnega okolja. Izbira postopka poravnave je omogočena preko menija *Registration*. Pojavi se modalno okno, ki prikazuje vse razpoznane postopke, kot opisano v poglavju 2.2.2.

Programska orodjarna je sestavljena iz treh skupin. Prva skupina, imenovana *Reference Image Tools* vsebuje orodja, ki neposredno vplivajo na referenčno sliko. Orodje *Hist.* prikaže histogram in orodje *ROI* omogoča opredelitev območja zanimanja na referenčni sliki. Druga skupina, imenovana *Registration Tools* je namenjena postopkom poravnave slik in analizi rezultatov. Vsebuje orodje *Run*, ki zažene izbrani postopek poravnave. Orodje *Apply transformation* aplicira geometrijsko transformacijo na poravnano sliko. Podatke o transformaciji prebere iz REG podatkovne strukture. Orodje *Joint Hist.* prikaže vezan histogram referenčne in poravnane slike. Orodje *Difference* prikaže absolutno razliko med referenčno in poravnano sliko, orodje *Checkerboard* pa v obliki šahovnice prikaže razlike med slikama. Tretja skupina, imenovana *Moving Image Tools* vsebuje orodja, ki neposredno vplivajo na poravnano sliko. V tej skupini se nahajata orodji *Hist* in *ROI*. Za področje *Programske orodjarne* je značilno, da je vsaka funkcionalnost natanko določena s pomočjo konfiguracijske datoteke, imeno-

vane *configureGui.m*. V dani datoteki je za vsako funkcionalnost določena relativna pot do Matlab datoteke, ki bo izvedla izbrano akcijo. Takšen arhitekturni vzorec dopušča možnost, da povsem prilagodimo funkcionalnosti danega področja.

```
%% gui/internal/tools

GUI.internal.tools = {};
GUI.internal.tools.path = strcat(GUI.internal.path, 'tools/');

GUI.internal.tools.selector = {};
GUI.internal.tools.selector.name = 'selectorHandler.m';

GUI.internal.tools.joint_histogram = {};
GUI.internal.tools.joint_histogram.name = 'jointHistogram.m';

GUI.internal.tools.transform = {};
GUI.internal.tools.transform.name = 'transform.m';

GUI.internal.tools.difference = {};
GUI.internal.tools.difference.name = 'differenceHandler.m';

GUI.internal.tools.checkerboard = {};
GUI.internal.tools.checkerboard.name = 'checkerboardHandler.m';
```

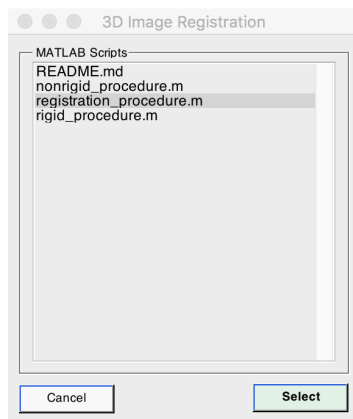
Slika 6: Izsek nastavitve znotraj konfiguracijske datoteke *configureGui.m*, za področje *Programska orodjarna*.

Področji *Referenčna slika* in *Poravnavana slika* omogočata prikazovanje posameznih rezin 3-D medicinskih slik. Uporabnik lahko bodisi vpiše željeno rezino ali se s pomočjo drsnika premakne do nje.

2.2.2 GUI-API programski vmesnik

Zelo pomembna funkcionalna zahteva grafičnega razvojnega okolja je enostavno dodajanje in uporaba novih postopkov poravnave medicinskih slik. Za doseg opisane funkcionalnosti, smo implementirali programski vmesnik imenovan *GUI-API*. Implementiran je kot bralec datotek v točno določenem datotečnem imeniku. Privzeta nastavitev določa imenik *external*, ki se nahaja znotraj projekta grafičnega razvojnega okolja. Seveda pa lahko dani imenik zamenjamo s poljubnim, tako da v konfiguracijski datoteki *configureGui.m* ustrezno popravimo relativno pot, ki se nahaja pod oznako *GUI.internal.path*. V splošnem *GUI-API* programski vmesnik razpozna datoteke v izbranem imeniku na podlagi naslednjih pravil:

- datoteka mora imeti *Matlab* končnico (*.m),
- v imeniku obstaja natanko ena datoteka za postopek poravnave (v kolikor implementacija postopka zahteva več datotek, morajo biti te shranjene v pod-imeniku).



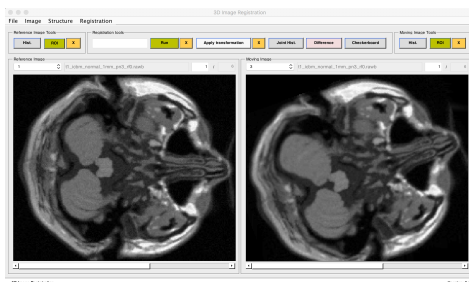
Slika 7: *GUI-API* modalno okno za izbiro postopka poravnave slik.

Pravilno strukturiran datotečni imenik *external* je nujno potreben za uspešno integracijo in uporabo različnih postopkov poravnave medicinskih slik v grafičnem razvojnem okolju. Po pravilno izvedeni integraciji, *GUI-API* programski vmesnik razpozna postopek. Postopek je nato na voljo za uporabo v grafičnem uporabniškem vmesniku, kot prikazano na sliki 7.

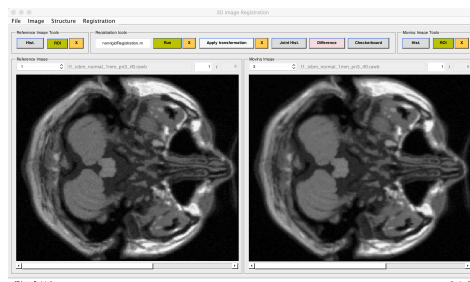
2.3 Primer uporabe

Primer uporabe grafičnega razvojnega okolja bomo predstavili s pomočjo razvoja postopka poravnave medicinskih slik z B-zlepki. Pri tem se bomo usmerili predvsem na enostavnost in hitrost začetne postavitve takega okolja. Preverili bomo možnost uvoza, vizualizacije in primerjave 3-D medicinskih slik v grafičnem uporabniškem vmesniku. Opisali bomo način implementacije in pravilne integracije postopka poravnave medicinskih slik v grafičnem razvojnem okolju. Usmerili se bomo na pravilnost uporabe REG podatkovne strukture za shranjevanje rezultatov poravnave.

Razvoj netovega postopka poravnave slik smo pričeli tako, da smo v Matlab programu odprli projekt v katerem se nahaja izvorna koda grafičnega razvojnega okolja. Preko Matlab ukazne vrstice smo zagnali ukaz *runGui*. Prikazal se je grafični uporabniški vmesnik, kot vidimo na sliki 5. Medicinske slike smo pridobili iz BrainWeb spletne podatkovne baze, ki je prosto dostopna na spletnem mestu <http://brainweb.bic.mni.mcgill.ca/brainweb/>. Za referenčno sliko smo uporabili MRI sliko možganov. Poravnava sliko smo pridobili iz referenčne slike, kot opisano v poglavju 4. Pridobljeni slike smo preko grafičnega uporabniškega vmesnika naložili v REG podatkovno strukturo. Slika 8 prikazuje referenčno in poravnava sliko pred izvedbo postopka poravnave. Brez dodatnega kodiranja smo se lahko premikali po rezinah



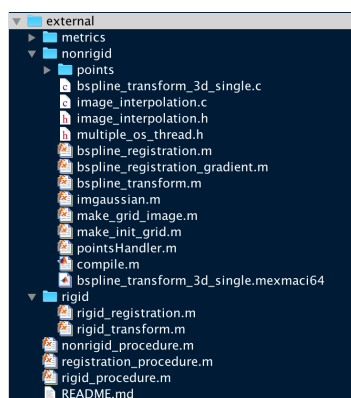
Slika 8: Prikaz MRI slik pred poravnavo (1 rezina).



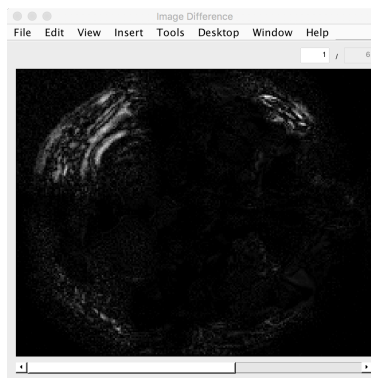
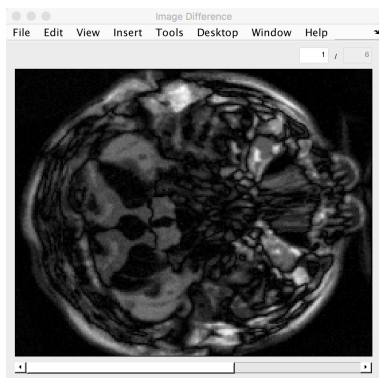
Slika 9: Prikaz MRI slik po poravnavi (1 rezina).

referenčne in poravnavane slike in s pomočjo orodij opisanih v poglavju 2.2.1 analizirali njihove relacije. Slika 11 prikazuje absolutno razliko MRI slik pred poravnavo. Slika 13 prikazuje začetno mrežo kontrolnih točk pred poravnavo. Iz opisanega lahko zaključimo, da je inicializacija grafičnega uporabniškega vmesnika uspešno izvedena. Delo s 3-D medicinskimi slikami je v grafičnem razvojnem okolju zelo enostavno.

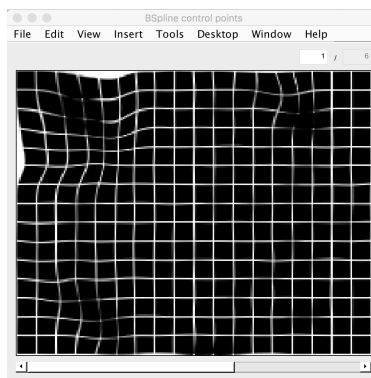
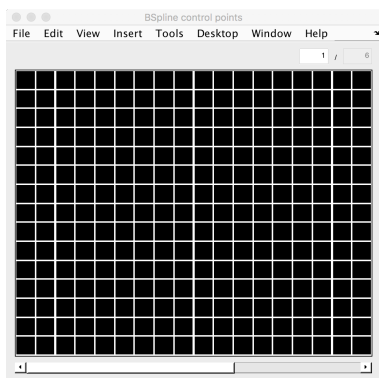
V namen uspešne integracije in uporabe postopka poravnave slik z B-zlepki v grafičnem razvojnem okolju, smo projekt postopka poravnave organizirali na naslednji način. Datoteka *registration_procedure.m*, ki določa vstopno točko v postopek poravnave se nahaja v korenu imenika *external*. Zaradi pravil GUI-API programskega vmesnika, so vse funkcije, ki omogočajo togo geometrijsko poravnavo shranjene v podimeniku *rigid*, funkcije ki omogočajo netogo (elastično) geometrijsko poravnavo pa v podimeniku *nonrigid*. Slika 10 prikazuje vsebino imenika *external*.

Slika 10: Vsebina datotečnega imenika *external*.

Pravilno strukturiran projekt postopka poravnave je nujno potreben za uspešno integracijo v grafično razvojno okolje. Za testiranje izbire postopka poravnave medicinskih slik z B-zlepki preko grafičnega uporabniškega vmesnika smo izbrali akcijo



Slika 11: Prikaz absolutne razlike MRI slik pred poravnavo (1 rezina) Slika 12: Prikaz absolutne razlike MRI slik po poravnavi (1 rezina)



Slika 13: Prikaz mreže kontrolnih točk pred poravnavo (1 rezina) Slika 14: Prikaz mreže kontrolnih točk po poravnavi (1 rezina)

Registration/Select procedure.... Pojavilo se je modalno okno, ki nam je s pomočjo *GUI-API* programskega vmesnika prikazalo vse razpoznane postopke, med drugim tudi opisani postopek. Po uspešno izbranem postopku smo le-tega zagnali preko akcije *Run*, ki se nahaja v programski orodjarni, kot opisano v poglavju 2.2.1. Grafični uporabniški vmesnik je zagnal postopek in omogočil njegovo razhroščevanje. Iz opisanega lahko zaključimo, da grafično razvojno okolje omogoča dodajanje, izvajanje in testiranje različnih postopkov poravnave medicinskih slik na enostaven in pregleden način.

Po končanem postopku poravnave smo preko akcije *Apply transformation*, aplicirali rezultate poravnave na poravnavano sliko, kot prikazano na sliki 9.

Na sliki 12 je prikazana absolutna razlika slik v prvi rezini po opravljeni poravnavi. Vidimo, da je bila deformacija lobanje uspešno odpravljena. Kljub temu so še vedno prisotne določene svetlostne razlike. Iz tega sklepamo, da se je optimizacijski postopek zaustavil v lokalnem minimumu kriterijske funkcije netoge poravnave z B-zlepki. Slika

14 prikazuje mrežo kontrolnih točk po izvedeni poravnavi. Ugotavljamo, da grafično razvojno okolje omogoča pregled in analizo rezultatov postopkov poravnave medicinskih slik.

Medicinske slike in pridobljene rezultate, ki so naloženi v podatkovni strukturi, smo preko grafičnega uporabniškega vmesnika shranili v datotečni sistem. To smo storili tako, da smo izbrali akcijo *Structure / Save To File....* Pojavilo se je modalno okno, ki nam je omogočilo izbiro lokacije na datotečnem sistemu. Po uspešno izbrani lokaciji smo izbrali akcijo *Save*. Grafični uporabniški vmesnik je podatkovno strukturo shranil na izbrano lokacijo v datotečnem sistemu in nam omogočil kasnejši uvoz.

Iz primera uporabe je razvidno, da je grafično razvojno okolje zadostilo vsem začetnim funkcionalnim zahtevam. Dobro definirana podatkovna struktura je omogočila uporabo postopka poravnave medicinskih slik z B-zlepki v grafičnem razvojnem okolju in prikaz rezultatov v grafičnem uporabniškem vmesniku. Brez dodatnega kodiranja smo se lahko premikali po rezinah referenčne in poravnane slike ter s pomočjo številnih orodij analizirali njihove relacije. Postopek implementacije je bil zaradi uporabe nizko nivojskih funkcij iz programske orodjarne *REG Toolbox* veliko preprostejši in bolj zanesljiv. Implementirati smo morali le višje nivojske komponente postopka poravnave, kot opisano v poglavju 3. Rezultate poravnave smo lahko neposredno aplicirali na poravnano sliko znotraj grafičnega uporabniškega vmesnika, kar je prispevalo k hitrejši in zanesljivejši analizi in razumevanju rezultatov. Medicinske slike in rezultate poravnave smo lahko shranili na datotečni sistem za kasnejšo obdelavo in analizo. Ugotovili smo, da je razvoj postopkov poravnave medicinskih slik v grafičnem razvojnem okolju veliko hitrejši in zanesljivejši, kot sicer.

3 Poravnava slik z B-zlepki

Drugi del magistrske naloge opisuje študijo postopka poravnave medicinskih slik z B-zlepki in njegovo dopolnitev ter integracijo v grafično razvojno okolje. Omenjeni postopek temelji na raziskovalnem delu, objavljenem v [15]. Začetna implementacija je pridobljena iz programske knjižnice, imenovane *B-spline Grid, Image and Point based Registration*, avtorja Dirk-Jan Kroon. Dostopna je na spletnem mestu <https://www.mathworks.com/matlabcentral/fileexchange/20057-b-spline-grid-image-and-point-based-registration>.

Glavni namen geometrijske transformacije v postopku poravnave slik je določiti relacijo med slikovnimi elementi referenčne in poravnane slike. Za par slik A in B je geometrijska transformacija določena s preslikavo T , ki vsako točko x slike A , preslika v ustrezno anatomsko lokacijo $T(x)$ na sliki B . V splošnem so pri poravnavi medicinskih slik najbolj pogoste netoge (elastične) geometrijske transformacije, ki omogočajo modeliranje lokalnih deformacij. V nadaljevanju bomo opisali postopek poravnave medicinskih slik, ki vključuje toge in netoge geometrijske transformacije z B-zlepki. V osnovi lahko opisani postopek poravnave medicinskih slik določimo s spodnjim matematičnim izrazom, kjer T_{global} označuje togo in T_{local} netogo transformacijo z B-zlepki:

$$T(x, y, z) = T_{local}(T_{global}(x, y, z)).$$

3.1 Toga geometrijska transformacija

Toga geometrijska transformacija omogoča modeliranje togih transformacij objektov na sliki. Za togo transformacijo je značilno, da ohranja razdaljo med poljubnima dvema točkama na sliki. Tovrstno transformacijo lahko za dvodimenzionalne slike opišemo s tremi parametri (premik v x in y smeri ter rotacijo), za trodimenzionalne slike pa s šestimi parametri (trije premiki t , trije zasuki R). Obstoječe toge operacije v splošnem razširimo še z možnostjo skaliranja, ki še dodatno uvede novih 6 prostostnih stopenj. V 3-D prostoru lahko togo geometrijsko transformacijo zapišemo z naslednjim mate-

matičnim izrazom:

$$T_{global}(x, y, z) = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \theta_{14} \\ \theta_{24} \\ \theta_{34} \end{pmatrix} [15],$$

kjer koeficienti θ določajo parametre 12 prostostnih stopenj toge geometrijske transformacije.

3.1.1 Optimizacija toge transformacije

Za optimizacijo parametrov toge geometrijske transformacije smo uporabili optimizacijski postopek imenovan *COBYLA*, opisan v [12]. Ta je del prosto dostopne knjižnice NLOpt [8], ki je integrirana v grafičnem razvojnem okolju. V vsakem koraku optimizacije, *kriterijska funkcija* s pomočjo parametrov premika t , rotacije R in opisane toge transformacije izračuna transformirano sliko TB . Podobnost anatomskih struktur med referenčno sliko A in sliko TB ovrednoti z *MAD* mero podobnosti. Postopek optimizacije se zaključi, ko izračunana podobnost slik doseže globalni optimum kriterijske funkcije.

3.1.2 Implementacija in integracija z grafičnim razvojnim okoljem

Postopek toge poravnave je razdeljen na več korakov. V prvem koraku postopek inicializira polje premikov in rotacij toge transformacije. Inicializirane vrednosti posreduje optimizacijskemu algoritmu *COBYLA*. Slednji je namenjen optimiziranju parametrov premika in rotacije slikovnih elementov. Za namen ovrednotenja opisanih parametrov uporablja kriterijsko funkcijo, ki je prikazana v algoritmu 2. Po uspešno opravljeni optimizaciji postopek ponovno izračuna geometrijsko transformacijo T , na podlagi optimalnih parametrov premika in rotacije. Izračunano geometrijsko transformacijo T postopek shrani v REG podatkovno strukturo, kot prikazano v algoritmu 1.

Kriterijska funkcija toge geometrijske poravnave na podlagi vhodnih parametrov premika in rotacije izračuna geometrijsko transformacijo T . Geometrijska transformacija T služi za izračun koordinat vokslov na transformirani mreži poravnane slike TB . Omenjena transformacija se izvrši v nizko nivojski funkciji, imenovani *resample-Mov2Ref*, ki je del REG programske orodjarne. Po uspešni transformaciji poravnane slike TB , kriterijska funkcija izmeri podobnost z referenčno sliko A . Podobnost

Algoritem 1: Postopek toge poravnave (*rigid_procedure*)

Vhod: *REG* podatkovna struktura**Izhod:** *REG* podatkovna struktura

```

1 // Inicializacija polja premikov
2  $t = [0, 0, 0]$ ;
3 // Inicializacija polja rotacij
4  $R = [0, 0, 0]$ ;
5 // Optimizacijski algoritem s kriterijsko funkcijo rigid_registration
6  $(t, R) = \text{optimize}(\text{rigid\_registration}(t, R))$ ;
7 // Na podlagi optimiziranih parametrov premika in rotacij izračunamo
  transformacijsko matriko  $T$ 
8  $T = \text{rigid\_transform}(t, R)$ ;
9 // Shranjevanje rezultatov poravnave v REG podatkovno strukturo
10  $REG[\text{movingImageIdx}].T = T$ ;
11 vrni REG;

```

izračuna na podlagi *MAD* mere podobnosti. Izračunana podobnost je nato posredovana optimizacijskemu algoritmu, ki se odloči o naslednjem koraku optimizacije, kot prikazano v algoritmu 2.

Kot razvidno iz psevdokode postopka poravnave in kriterijske funkcije je integracija z grafičnim razvojnim okolje dosežena s pravilno uporabo *REG* podatkovne strukture. Ta je uporabljena za branje referenčne slike *A*, poravnavane slike *B* in za shranjevanje rezultatov toge transformacije *T*.

3.2 Netoga geometrijska transformacija

Netoga geometrijska transformacija omogoča modeliranje netogih (elastičnih) deformacij objektov na slikah. Za modeliranje uporablja netogi matematični model in B-zlepke. Osnovna ideja netovega modela temelji na deformaciji objekta na sliki z manipulacijo mreže kontrolnih točk. Rezultat deformacije je gladka in nadzorovana oblika 3-D objekta.

Zlepek ali zlepljena krivulja je povezana, odsekoma polinomska krivulja, ki jo sestavljajo zlepki. *Odsek* krivulje imenujemo območje med dvema točkama, imenovanima *vozla*. Odseki se med seboj stikajo izključno v sprednjem in zadnjem koncu in skupaj sestavljajo krivuljo. Ponavadi so sestavljeni iz enega samega zlepka (npr. hermitski zlepki). V tem primeru sta pojma med seboj ekvivalentna. Pri nekaterih pa so združene vrednosti več zaporednih zlepkov (npr. bazni zlepki). Velja, da je odsek zmeraj mogoče

Algoritem 2: Kriterijska funkcija toge poravnave (*rigid_registration*)

Vhod: parametri premikov t , parametri rotacije R **Izhod:** Mera podobnosti O_error

```

1 // Branje referenčne slike iz REG podatkovne strukture
2  $A = REG[referenceImageIdx].data;$ 
3 // Branje poravnavane slike iz REG podatkovne strukture
4  $B = REG[movingImageIdx].data;$ 
5 // Na podlagi parametrov premika in rotacij izračunamo transformacijsko
   matriko  $T$ 
6  $T = rigid\_transform(t, R);$ 
7 // Shranjevanje rezultatov poravnave v REG podatkovno strukturo
8  $REG[movingImageIdx].T = T;$ 
9 // Prezorčimo poravnavano sliko TB na mrežo referenčne slike A
10  $TB = resampleMov2Ref(REG);$ 
11 // Izračunamo  $MAD$  mero podobnosti, med referenčno sliko A in
   transformirano sliko TB
12  $O\_error = MAD(A, TB);$ 
13 vrni  $O\_error;$ 

```

zapisati z enim samim polinomom. *Zlepek* je funkcija, ki je sestavljena iz enega (npr. hermitski zleпки) ali več (npr. bazni zleпки) polinomov. V primeru, da ga sestavlja en sam polinom, je zlepek definiran med dvema zaporednima vozlova in ekvivalenten odseku na tem območju. V drugem primeru, kjer ga sestavlja več polinomov, pa se zlepek razteza preko več vozlov in med vsakim parom le-teh ga definira drug polinom. Bistveno je, da ta funkcija služi kot sestavni del zlepljene krivulje.

Slovensko ime zlepek izvira iz besedne igre, da se zleпки zlepijo med seboj in tako skupaj ustvarijo zlepljeno krivuljo. Angleško ime pa izvira iz besede “spline”, ki je označevala tanke lesene deske, ki so jih ladjedelci ukrivili in ukrivljene zlepli skupaj tako, da so obdržale ukrivljenost. Slednje so nato uporabili pri izgradnji ukrivljenih ladijskih trupov in kasneje tudi pri avtomobilih in letalih ter še pogosteje kot pomoč pri risanju krivulj. Bistven pri tem je koncept, da so uspeli iz majhnih koščkov ustvariti večjo gladko krivuljo.

Lastnosti zlepkov:

- **Povezanost stopnje n :** število zlepkov/odsekov ki tvori krivuljo.
- **Stopnja ali red zlepka:** najvišja stopnja polinoma, ki ga sestavlja.
- **Stopnja ali red krivulje:** najvišja stopnja zlepka, ki ga sestavlja.

Za določiti netogi matematični model, bomo označili domeno slike z $\Omega = (x, y, z) \mid 0 \leq x \leq X, 0 \leq y \leq Y, 0 \leq z \leq Z$. Naj Φ označuje $n_x \times n_y \times n_z$ mrežo kontrolnih točk $\Phi_{i,j,k}$ z enoličnim razmikom med kontrolnimi točkami δ . Potem netogi model lahko zapišemo kot 3-D tenzorski produkt 1-D kubičnih B-zlepkov:

$$T_{local}(x, y, z) = \sum_{l=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 B_l(u)B_m(v)B_n(w)\phi_{i+l,j+m,k+n} [15],$$

kjer $i = \lfloor x/n_x \rfloor - 1$, $j = \lfloor y/n_y \rfloor - 1$, $k = \lfloor z/n_z \rfloor - 1$, $u = x/n_x - \lfloor x/n_x \rfloor$, $v = y/n_y - \lfloor y/n_y \rfloor$, $w = z/n_z - \lfloor z/n_z \rfloor$ in B_l določa l -to osnovno funkcijo B-zlepka:

$$B_0(u) = (1 - u)^3/6$$

$$B_1(u) = (3u^3 - 6u^2 + 4)/6$$

$$B_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6$$

$$B_3(u) = u^3/6$$

V splošnem so B-zlepki lokalno nadzorovani, kar prispeva k pohitritvi računske kompleksnosti, tudi v primeru velikega števila kontrolnih točk. Sprememba kontrolne točke na poziciji (i, j, k) vpliva zgolj na transformacijo slikovnih elementov v njeni neposredni bližini.

Mreža kontrolnih točk določa parametre postopka poravnave slik z B-zlepki in število prostostnih stopenj algoritma. Majhno število kontrolnih točk omogoča poravnavo globalnih, veliko število pa izredno lokalnih netogih deformacij. Posledično velja, da večja kot je ločljivost mreže kontrolnih točk, večja bo časovna kompleksnost algoritma. Na koncu gre za empirično odločitev med fleksibilnostjo poravnave in trajanjem postopka, ki je določena tudi na podlagi pričakovanih deformacij tkiv na slikah.

3.2.1 Optimizacija netoge transformacije

Za optimizacijo mreže kontrolnih točk smo uporabili gradientni optimizacijski postopek imenovan *Limited memory BFGS*, opisan v [10]. Ta je del prosto dostopne knjižnice NLOpt, ki je integrirana v grafičnem razvojnem okolju. V vsakem koraku optimizacije, *kriterijska funkcija* s pomočjo mreže kontrolnih točk in opisane netoge transformacije z B-zlepki izračuna transformirano sliko TB . Podobnost anatomskih struktur med referenčno sliko A in sliko TB ovrednoti z MI mero podobnosti. Smer optimizacije določi z izračunom gradienta mreže kontrolnih točk. Gradient in MI mera podobnosti določata izhodne parametre kriterijske funkcije, s katerimi optimizacijski postopek optimizira mrežo kontrolnih točk, kot prikazano v algoritmu 4.

3.2.2 Implementacija in integracija z grafičnim razvojnim okoljem

Postopek netoge poravnave je razdeljen na več korakov. V prvem koraku postopek inicializira mrežo kontrolnih točk. Začetna mreža kontrolnih točk je kvadratna mreža, ki ne deformira poravnavane slike. Inicializirano mrežo posreduje optimizacijskemu algoritmu *Limited memory BFGS*, ki optimizira mrežo kontrolnih točk, dokler ne doseže globalnega optimuma kriterijske funkcije. Po uspešno opravljeni optimizaciji, postopek s pomočjo optimizirane mreže kontrolnih točk izračuna transformirano sliko TB in polje premikov D vseh slikovnih elementov s pomočjo B-zlepkov. Funkcija *bspline_transform*, ki izračuna transformirano sliko TB in polje premikov D , je del programske knjižnice *B-spline Grid, Image and Point based Registration*, avtorja Dirk-Jan Kroon. Izračunano polje premikov D postopek shrani v REG podatkovno strukturo.

Algoritem 3: Postopek netoge poravnave (*nonrigid_procedure*)

Vhod: REG podatkovna struktura

Izhod: REG podatkovna struktura

```

1 // Branje poravnavane slike iz REG podatkovne strukture
2  $B = REG[movingImageIdx].data;$ 
3 // Inicializacija mreže kontrolnih točk
4  $O\_grid = make\_init\_grid(B);$ 
5 // Optimizacijski algoritem s kriterijsko funkcijo nonrigid_registration
6  $O\_grid = optimize(nonrigid\_registration(O\_grid));$ 
7 // Izračun transformirane slike TB in polja premikov D vseh slikovnih
  elementov s pomočjo B-zlepkov
8  $[TB, D] = bspline\_transform(B, O\_grid);$ 
9 // Shranjevanje polja premikov D v REG podatkovno strukturo
10  $REG[movingImageIdx].D = D;$ 
11 vrni REG;
```

Kriterijska funkcija netoge poravnave na podlagi vhodne mreže kontrolnih točk in poravnavane slike B izračuna transformirano sliko TB in polje premikov D vseh slikovnih elementov s pomočjo B-zlepkov. Po uspešni transformaciji poravnavane slike TB , kriterijska funkcija izmeri podobnost z referenčno sliko A . Podobnost izračuna na podlagi mere MI , kot je to prikazano v algoritmu 4. Implementacija funkcije, ki izračuna dano mero podobnosti, je del REG programske orodjarne. V kolikor optimizacijski algoritem zahteva izračun gradienta mreže kontrolnih točk, se dana operacija

izvede v funkciji *error_gradient*.

Algoritem 4: Kriterijska funkcija netoge poravnave (*nonrigid_registration*)

Vhod: mreža kontrolnih točk *O_grid*

Izhod: Mera podobnosti *O_error*, gradient mreže kontrolnih točk *O_grad*

```

1 // Branje referenčne slike iz REG podatkovne strukture
2 A = REG[referenceImageIdx].data;
3 // Branje poravnavane slike iz REG podatkovne strukture
4 B = REG[movingImageIdx].data;
5 // Izračun transformirane slike TB in polja premikov D vseh slikovnih
  elementov s pomočjo B-zlepkov
6 [TB, D] = bspline_transform(B, O_grid);
7 // Izračun MI mere podobnosti, med referenčno sliko A in transformirano sliko
  TB
8 O_error = MI(A, TB);
9 // V kolikor optimizacijski algoritem zahteva izračun gradienta mreže kontrolnih
  točk
10 če O_grad > 0 potem
11   // Izračunamo gradient mreže kontrolnih točk
12   O_grad=error_gradient(A, B, TB, O_grid);
13 vrni O_error, O_grad;
```

Gradient mreže kontrolnih točk kriterijska funkcija pridobi na podlagi referenčne slike *A*, poravnavane slike *B*, transformirane slike *TB* in pridobljene mreže kontrolnih točk *O_grid*. V vsaki iteraciji optimizacijskega postopka, vsako kontrolno točko ločeno premaknemo v *x, y, z* smeri za določen korak μ in izračunamo netogo transformacijo. Iz dobljene transformirane slike *TB'* in referenčne slike *A* izračunamo podobnost. Dobljeno mero odštejemo z *MI* mero podobnosti izračunano v prvem koraku kriterijske funkcije in delimo s korakom μ . Izračunana vrednost nam določa smer gradienta obravnavane kontrolne točke, kot prikazano v algoritmu 5. V splošnem uporabljamo iterativno tehniko najstrmejšega spusta, ki za določen korak μ teži v smeri gradienta.

Kot razvidno iz psevdokode postopka poravnave in kriterijske funkcije je integracija z grafičnim razvojnim okolje dosežena s pravilno uporabo REG podatkovne strukture. Ta je uporabljena za branje referenčne slike *A*, poravnavane slike *B* in za shranjevanje rezultatov netoge transformacije.

Algoritem 5: Izračun gradienta mreže kontrolnih točk

Vhod: Referenčna slika A , poravnavana slika B , transformirana slika TB ,
mreža kontrolnih točk O_grid , mera podobnosti O_error

Izhod: Gradient mreže kontrolnih točk O_grad

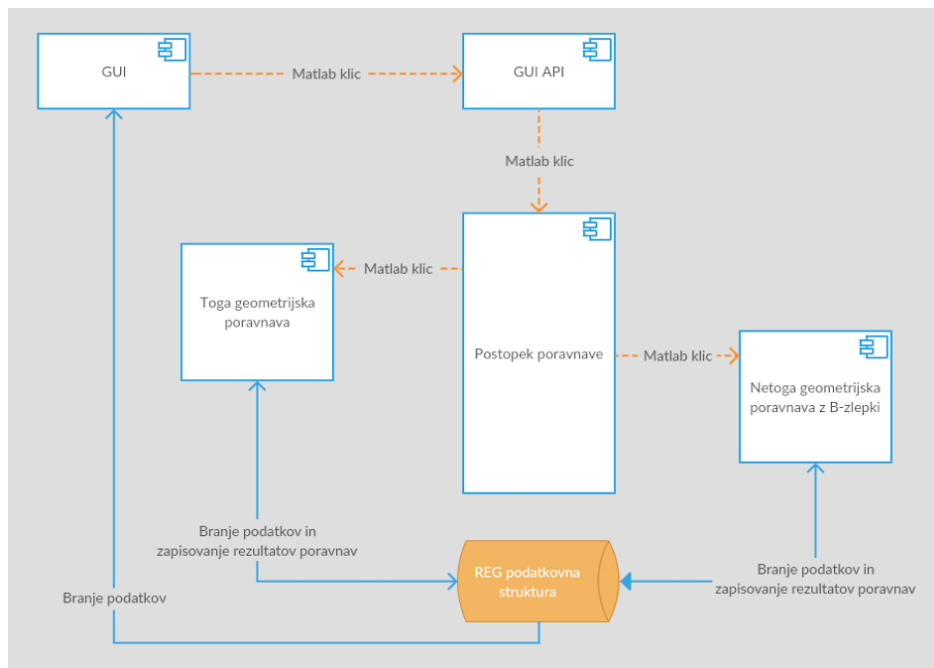
```

1  $O\_grad = zeros(size(O\_grid));$ 
2  $step = 1;$ 
3 za  $zi = 0 : 1 : 3$ 
4     za  $zj = 0 : 1 : 3$ 
5         za  $zk = 0 : 1 : 3$ 
6              $O\_gradpx = O\_grid;$ 
7              $O\_gradpy = O\_grid;$ 
8              $O\_gradpz = O\_grid;$ 
9             za  $i = (1 + zi) : 4 : size(O\_grid, 1)$ 
10                 za  $j = (1 + zj) : 4 : size(O\_grid, 2)$ 
11                     za  $k = (1 + zk) : 4 : size(O\_grid, 3)$ 
12                          $O\_gradpx(i, j, k, 1) = O\_gradpx(i, j, k, 1) + step;$ 
13                          $O\_gradpy(i, j, k, 2) = O\_gradpy(i, j, k, 2) + step;$ 
14                          $O\_gradpz(i, j, k, 3) = O\_gradpz(i, j, k, 3) + step;$ 
15                      $I\_gradpx = bspline\_transform(B, O\_gradpx);$ 
16                      $I\_gradpy = bspline\_transform(B, O\_gradpy);$ 
17                      $I\_gradpz = bspline\_transform(B, O\_gradpz);$ 
18                     za  $i = (1 + zi) : 4 : size(O\_grid, 1)$ 
19                         za  $j = (1 + zj) : 4 : size(O\_grid, 2)$ 
20                             za  $k = (1 + zk) : 4 : size(O\_grid, 3)$ 
21                                  $E\_gradpx = MI(A, I\_gradpx);$ 
22                                  $E\_gradpy = MI(A, I\_gradpy);$ 
23                                  $E\_gradpz = MI(A, I\_gradpz);$ 
24                                  $O\_grad(i, j, k, 1) = (E\_gradpx - O\_error)/step;$ 
25                                  $O\_grad(i, j, k, 2) = (E\_gradpy - O\_error)/step;$ 
26                                  $O\_grad(i, j, k, 3) = (E\_gradpz - O\_error)/step;$ 
27 vrni  $O\_grad;$ 

```

3.3 Arhitektura postopka poravnave z B-zlepki

Arhitektura postopka poravnave medicinskih slik z B-zlepki je prikazana na sliki 15. Grafični uporabniški vmesnik preko GUI API programskega vmesnika zažene *Postopek*



Slika 15: Gradniki postopka poravnave medicinskih slik z B-zlepki.

poravnave. Pseudokoda le-tega je prikazana v algoritmu 6.

Algoritem 6: Postopek poravnave (*registration_procedure*)

- 1 // Izvedi togi postopek poravnave
 - 2 *rigid_procedure*();
 - 3 // Izvedi netogi postopek poravnave z b-zlepki
 - 4 *nonrigid_procedure*();
-

Kot prikazano v pseudokodi postopka poravnave, se najprej zagotovi izvedbo toge poravnave medicinskih slik. Za to je zadolžen *Togi postopek poravnave*, opisan v poglavju 3.1. Po uspešno izvedeni togi poravnavi medicinskih slik sledi *Netogi postopek poravnave z B-zlepki*, ki je opisan v poglavju 3.2. Po uspešni izvedbi postopka poravnave, so rezultati shranjeni v REG podatkovni strukturi. Rezultati so nato prikazani v grafičnem uporabniškem vmesniku. Opisana arhitektura omogoča, da lahko iz grafičnega razvojnega okolja testiramo togi in netogi postopek posamično ali združeno v skupnem postopku poravnave. Vsak postopek je implementiran tako, da podatke bere in zapisuje v REG podatkovno strukturo. Na takšen način smo omogočili enovit programski vmesnik za dostop do medicinskih slik in rezultatov poravnave, bodisi preko grafičnega uporabniškega vmesnika ali Matlab-ove ukazne vrstice.

4 Rezultati

Grafično razvojno okolje smo uspešno uporabili za implementacijo postopka poravnave medicinskih slik z B-zlepki. Osnovna implementacija tega postopka je bila pridobljena iz programske knjižnice, imenovane *B-spline Grid, Image and Point based Registration*, avtorja Dirk-Jan Kroon. Implementacijo smo spremenili tako, da ustreza zahtevam grafičnega razvojnega okolja. Posamezne gradnike smo zamenjali z nizko nivojskimi funkcijami iz REG programske orodjarne, kot opisano v poglavju 3. V nadaljevanju bomo predstavili rezultate delovanja postopka poravnave medicinskih slik z B-zlepki v grafičnem razvojnem okolju.

4.1 Rezultati postopka poravnave slik z B-zlepki

Postopek poravnave medicinskih slik z B-zlepki vključuje togo in netogo poravnavo, zato bomo rezultate obravnavali ločeno. V prvem delu bomo analizirali rezultate toge geometrijske poravnave, v drugem delu pa bomo opisali rezultate netoge poravnave z B-zlepki. Medicinske slike smo pridobili iz BrainWeb spletne podatkovne baze, ki je prosto dostopna na spletnem mestu <http://brainweb.bic.mni.mcgill.ca/brainweb/>. Za referenčno sliko smo uporabili MRI sliko možganov velikosti 181x217x181 vokslov in 1mm debeline. Za namen hitrejšega testiranja smo velikost referenčne slike zmanjšali na prvih 6 rezin, tako da je bila končna velikost uporabljenih slik 181x217x6 vokslov. Poravnava slike smo pridobili iz referenčne slike z apliciranjem toge in netoge (elastične) transformacije. Togo transformacijo smo simulirali z rotacijo po z osi za 20 stopinj. Netogo transformacijo smo pridobili z manipulacijo mreže kontrolnih točk, tako da smo točke na območju $[2 : 7, 2 : 7, 1 : 5, :]$ premaknili za 2mm. Pridobljeni sliko smo preko grafičnega uporabniškega vmesnika naložili v podatkovno strukturo.

4.1.1 Rezultati toge geometrijske poravnave

Rezultate toge geometrijske poravnave bomo opisali s pomočjo primerjanja vrednosti med izračunano in optimalno togo geometrijsko transformacijo. Ker smo poravnava slike pridobili z umetno aplicirano togo poravnavo, lahko z inverzno operacijo

izračunamo optimalno geometrijsko transformacijo T . V tabeli 1 so prikazane minimalne, maksimalne in srednje vrednosti izračunane in optimalne toge geometrijske transformacije. Razvidno je, da se je izračunana toga poravnava približala optimalni. Kljub temu pa so še vedno prisotne določene razlike, ki so lahko posledica interpolacije svetlostnih vrednosti. Poleg tega obstaja verjetnost, da se je optimizacijski postopek zaustavil v lokalnem minimumu kriterijske funkcije. Vse to lahko prispeva, da končna transformacija ne ustreza pravilni poravnavi med slikama.

Geo. trans. T	Min. vred.	Maks. vred.	Srednja vred.
izračunana	-0.3419	1.0000	0.2415
optimalna	-0.3420	1.0000	0.2425

Tabela 1: Primerjava min., maks. in srednje vrednosti *toge geo. trans.*.

Kot dodatek k analizi rezultatov toge poravnave je v tabeli 2 prikazana MAD mera podobnosti pred in po poravnavi. Optimizacijski algoritmi pri MAD meri iščejo minimum kriterijske funkcije, kar pomeni da sta si sliki čedalje bolj podobni, če se razlika med njima manjša oziroma je enaka 0. Iz opisane trditve je razvidno, da se je podobnost med slikama po poravnavi povečala.

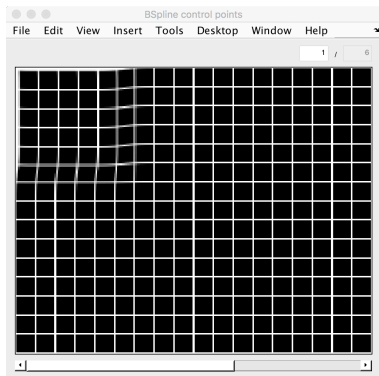
Mera podobnosti	Pred poravnavo	Po poravnavi
MAD	225.5830	66.9548

Tabela 2: Prikaz MAD mere podobnosti pred in po togi poravnavi.

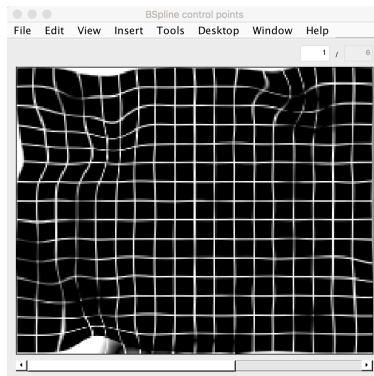
4.1.2 Rezultati netoge geometrijske poravnave z B-zlepki

Kot opisano na začetku poglavja 4 smo poravnavano sliko pridobili iz referenčne slike z apliciranjem toge in netoge transformacije. Netogo transformacijo smo pridobili z manipulacijo mreže kontrolnih točk, tako da smo točke na območju $[2 : 7, 2 : 7, 1 : 5, :]$ premaknili za 2mm. Rezultate netoge geometrijske poravnave z B-zlepki bomo opisali s primerjavo mreže kontrolnih točk. Primerjali bomo mrežo, ki jo je izračunal netogi postopek z B-zlepki z mrežo kontrolnih točk, ki v našem primeru določa optimum. Optimalno mrežo smo pridobili z inverzno operacijo nad umetno generirano netogo transformacijo.

V tabeli 3 so prikazane minimalne, maksimalne in srednje vrednosti izračunane in optimalne mreže kontrolnih točk po poravnavi. Iz tabele 3 in slik 17 in 16 je razvidno, da se mreža kontrolnih točk, ki jo je izračunal netogi postopek poravnave z B-zlepki ne



Slika 16: Prikaz optimalne mreže kontrolnih točk po poravnavi (1 rezina)



Slika 17: Prikaz izračunane mreže kontrolnih točk po poravnavi (1 rezina)

ujema z optimalno mrežo. Vidne razlike v poziciji kontrolnih točk so posledica številnih dejavnikov. Na končen rezultat poravnave vpliva interpolacija svetlostnih vrednosti v vsaki iteraciji optimizacije, predhodni postopek toge poravnave in optimizacijski postopek, ki se lahko zaustavi v lokalnem minimumu kriterijske funkcije.

Mreža kontrolnih točk	Min. vred.	Maks. vred.	Srednja vred.
po poravnavi	-12.9378	240.0063	71.2111
po poravnavi (optimalna)	-12	240	71.2103

Tabela 3: Primerjava min., maks. in srednje vrednosti mreže kontrolnih točk.

Kot dodatek k analizi rezultatov netoge poravnave je v tabeli 4 prikazana *MI* mera podobnosti pred in po poravnavi. *MI* mera ocenjuje statistično odvisnost med svetlostnimi nivoji obeh slik. Na ta način poda oceno, koliko ena slika pove o drugi sliki in zavzame največjo vrednost, ko sta sliki geometrijsko poravnani. Ker je vrednost *MI* mere po poravnavi večja kot pred poravnavo, lahko trdimo, da se je podobnost med slikama povečala.

Mera podobnosti	Pred poravnavo	Po poravnavi
MI	0.5298	1.9639

Tabela 4: Prikaz *MI* mere podobnosti pred in po netogi poravnavi.

5 Zaključek

Poravnava medicinskih slik je zelo pomembna komponenta številnih medicinskih aplikacij, saj omogoča združevanje in izločanje medicinsko pomembne informacijske vsebine slik. Uporablja se za odkrivanje in diagnozo bolezni, za načrtovanje terapij, za vodenje medicinskih posegov in za spremljanje zdravljenja bolnikov. Aplikira se lahko na slike enega objekta, ki so pridobljene z istim (enomodalna poravnava) ali različnim (večmodalna poravnava) slikovnim postopkom. V splošnem poravnava slik določa postopek iskanja optimalne geometrijske transformacije, ki preslika prostor ene slike v prostor druge slike, tako, da doseže optimalno prostorsko skladnost anatomskih struktur, ki jih sliki predstavljata. V praksi poznamo dva tipa geometrijskih transformacij. Toge transformacije modelirajo rotacije in premike slikovnih elementov, netoge (deformabilne) pa omogočajo modeliranje deformacij objektov na sliki.

Obstajajo številni uveljavljeni postopki za netogo poravnavo medicinskih slik. Velika večina takih postopkov je vključena v kompleksna programska orodja, ki po večini zahtevajo veliko začetnih adaptacij in učenja. Zelo težko je najti razvojna okolja, ki bi na enostaven način omogočala analizo, uporabo in razširitev znanih postopkov v namen pohitritve njihovega delovanja. To je bil tudi glavni povod magistrskega dela. V nalogi smo implementirali grafično razvojno okolje, ki na enostaven način omogoča analizo, integracijo, razširitev in razvoj številnih postopkov poravnave medicinskih slik. Kot dokaz uporabnosti in pravilnosti delovanja razvojnega okolja, smo v njem implementirali znani postopek za netogo poravnavo medicinskih slik z B-zlepki.

Izhodišče naloge so bile naslednje komponente. Prva komponenta je dobro definirana podatkovna struktura, ki določa osnovni nabor podatkov, ki so potrebni pri vsaki poravnavi medicinskih slik. Druga komponenta je programska orodjarna, skupek nizko nivojskih programskih funkcij, ki so bile implementirane v predhodnih raziskavah doc. dr. Peter Rogelj. Tretja komponenta je nabor programskih funkcij, ki vsebuje osnovno implementacijo postopka za netogo poravnano medicinskih slik z B-zlepki, avtorja Dirk-Jan Kroon.

Glavni namen naloge je bil implementirati grafično razvojno okolje, ki omogoča enostaven razvoj, integracijo in testiranje različnih postopkov poravnave medicinskih slik. Za dokaz pravilnosti delovanja in uporabnosti razvojnega okolja smo v njem implementirali znani postopek za poravnavo medicinskih slik z B-zlepki. Naloga je bila

razdeljena na štiri dele. V prvem delu smo povzeli lastnosti poravnave slik, uporabljenih slikovnih tehnik in na kratko predstavili glavne komponente, razdelitve in probleme postopkov poravnave v medicini. V drugem delu smo analizirali in opisali implementacijo grafičnega razvojnega okolja, ki omogoča hiter in učinkovit razvoj različnih postopkov poravnave medicinskih slik. V tretjem delu smo opisali in implementirali postopek poravnave medicinskih slik z B-zlepki v grafičnem razvojnem okolju. Postopek smo spremenili tako, da podatke o medicinskih slikah pridobi iz REG podatkovne strukture. V zadnjem delu smo opisali rezultate poravnave medicinskih slik s pomočjo grafičnega razvojnega okolja in postopka poravnave z B-zlepki.

Izvorna koda grafičnega razvojnega okolja in postopka poravnave medicinskih slik z B-zlepki je dostopna na spletnem mestu <https://github.com/matjazsu/image-registration> pod GNU GPL licenco.

6 Literatura

- [1] M. D. Abràmoff, P. J. Magalhães, and S. J. Ram. Image processing with imageJ, 2004. (*Citirano na strani 13.*)
- [2] B. B. Avants, N. J. Tustison, M. Stauffer, G. Song, B. Wu, and J. C. Gee. The Insight ToolKit image registration framework. *Frontiers in Neuroinformatics*, 8, 2014. (*Citirano na strani 13.*)
- [3] C. Cocosco, V. Kollokian, R. K. Kwan, G. B. Pike, and A. C. Evans. BrainWeb : Online Interface to a 3D MRI Simulated Brain Database. *3-rd International Conference on Functional Mapping of the Human Brain*, 5(4):S425, 1997. (*Citirano na straneh 13 in 16.*)
- [4] S. C. Dewhurst and K. Stark. Programming in C++. *ACM SIGPLAN OOPS Messenger*, 2(4):54–56, jan 2007. (*Citirano na strani 12.*)
- [5] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J. C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper, and R. Kikinis. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*, 30(9):1323–1341, 2012. (*Citirano na strani 13.*)
- [6] A. Jarc. Poravnava medicinskih slik na osnovi teksturnih značilnic. 2009. (*Citirano na straneh VIII, 2, 3, 5, 6, 7 in 8.*)
- [7] M. J.B.A. and V. M. A. A survey of medical image registration. *Med Image Anal*, 2(1):1–36, 1998. (*Citirano na strani 4.*)
- [8] U. Kumar, S. Soman, and Jayadeva. Benchmarking NLopt and state-of-the-art algorithms for continuous global optimization via IACOR. *Swarm and Evolutionary Computation*, 27:116–131, 2016. (*Citirano na strani 23.*)
- [9] C. S. Lent. Learning to Program with MATLAB Building GUI Tools. *John Wiley and Sons Inc.*, page 310, 2013. (*Citirano na straneh 14 in 15.*)

- [10] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989. (*Citirano na strani 26.*)
- [11] P. Mildenerger, M. Eichelberg, and E. Martin. Introduction to the DICOM standard, 2002. (*Citirano na straneh 13 in 16.*)
- [12] M. J. D. Powell. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In *Advances in Optimization and Numerical Analysis*, pages 51–67. 1994. (*Citirano na strani 23.*)
- [13] P. Rogelj. Poravnava medicinskih slik. pages 48–53, 2006. (*Citirano na strani 1.*)
- [14] G. V. Rossum, P. S. Foundation, U. Swallow, S. Python, P. Software, and F. License. Python (programming language). *Flying*, pages 1–14, 2011. (*Citirano na strani 13.*)
- [15] D. Rueckert. Nonrigid registration using free-form deformations: Application to breast mr images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, 1999. (*Citirano na straneh 11, 22, 23 in 26.*)
- [16] W. J. Schroeder and K. M. Martin. The visualization toolkit. In *Visualization Handbook*, pages 593–614. 2005. (*Citirano na strani 13.*)
- [17] G. Sharma and J. Martin. MATLAB®: A language for parallel computing. *International Journal of Parallel Programming*, 37(1):3–36, 2009. (*Citirano na strani 13.*)
- [18] D. Skerl. Kvantitativno vrednotenje mer podobnosti za poravnavo medicinskih slik. 2006. (*Citirano na straneh VIII in 8.*)
- [19] P. Viola and W. M. Wells. Alignment by Maximization of Mutual Information. *International Journal of Computer Vision*, 1997. (*Citirano na strani 7.*)
- [20] J. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer, R. M. Kessler, R. J. Maciunas, C. Barillot, D. Lemoine, A. Collignon, F. Maes, P. Suetens, D. Vandermeulen, P. A. van den Elsen, S. Napel, T. S. Sumanaweera, B. Harkness, P. F. Hemler, D. L. Hill, D. J. Hawkes, C. Studholme, J. B. Maintz, M. A. Viergever, G. Malandain, and R. P. Woods. Comparison and evaluation of retrospective intermodality image registration techniques. *International Society for Optics and Photonics*, pages 332–347, 1996. (*Citirano na straneh 13 in 16.*)