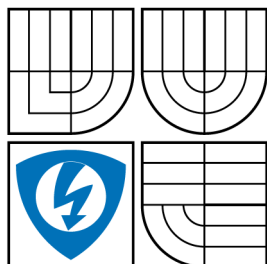


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ



FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

AFINNÍ LÍCOVÁNÍ NATIVNÍCH A POSTKONTRASTNÍCH CT SNÍMKŮ MOZKU

AFFINE REGISTRATION OF NATIVE AND CONTRAST-ENHANCED CT BRAIN
IMAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JIŘÍ CHMELÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR WALEK

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav biomedicínského inženýrství

Diplomová práce

magisterský navazující studijní obor
Biomedicínské a ekologické inženýrství

Student: Bc. Jiří Chmelík

ID: 119454

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Afinní lícování nativních a postkontrastních CT snímků mozku

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši metod a algoritmů využívaných při lícování objemových dat získaných pomocí rentgenové výpočetní tomografie. Zaměřte se především na metodu lícování pomocí optimalizace globální intenzitní podobnostní funkce dvojice obrazů. 2) V programovém prostředí MATLAB realizujte algoritmus, kterým načtete data ve formátu DICOM a provedete globální afinní transformaci objemového obrazu podle pevně zadaných parametrů včetně interpolace dat. 3) Navrhněte a realizujte algoritmus, pomocí kterého z obrazů odstraníte patientský stůl a klínový filtr, které by registraci znemožňovaly. 4) S ohledem na charakter obrazových dat (nativní a postkontrastní snímky mozku) vyberte a naprogramujte kritérium podobnosti dvojice obrazů. 5) Vyberte, popište a realizujte optimalizační algoritmus, který bude vhodný pro řešení daného problému. Programy vytvořené v předešlých bodech propojte v jeden celek a realizujte vlastní slícování obrazových dat. 6) Proveďte diskuzi a zhodnocení dosažených výsledků.

DOPORUČENÁ LITERATURA:

- [1] JAN, Jiří. Medical Image Processing, Reconstruction and Restoration: Concepts and Methods. Boca Raton: CRC Press, 2005, ISBN 0-8247-5849-8
[2] ZITOVÁ, Barbara a FLUSSER, Jan. Image registration methods: a survey. Image and Vision Computing. 2003, roč. 21, č. 11, s. 977-1000.

Termín zadání: 11.2.2013

Termín odevzdání: 24.5.2013

Vedoucí práce: Ing. Petr Walek

Konzultanti diplomové práce:

prof. Ing. Ivo Provazník, Ph.D.

Předseda oborové rady

ABSTRAKT

Tato práce se zabývá řešením problému registrace obrazů mozku pořízených rentgenovou výpočetní tomografií. Úvodem práce je seznámení s metodami geometrických transformací obrazů, zejména pak afinní transformací. Dále se text zabývá problematikou interpolačních metod, výpočtů podobnostních kritérií a následné optimalizace. Veškerá problematika je řešena speciálně pro trojrozměrná data. Druhá část práce je praktickou ukázkou programu v prostředí MatLab® pro registraci pořízených snímků pomocí afinní transformace, jehož součástí je také algoritmus pro odstranění schodového artefaktu, podložky pod hlavu a patientského stolu. Jako optimalizační algoritmus je použita metoda kontrolovaného náhodného prohledávání (CRS). Vzhledem k typu medicínských obrazů poskytovaných ve formátu DICOM je součástí práce také proces jejich načtení a uložení.

KLÍČOVÁ SLOVA

afinní transformace, rentgenová výpočetní tomografie, interpolační metoda, podobnostní kritérium, lícování obrazů, optimalizační metoda

ABSTRACT

This thesis is dealing with problem of brain images registration acquired by computed tomography. At the beginning is explanation of image geometrical transformation methods, notably affine transformation. Following part of text the is dealing with interpolation methods issues, calculation of similarity criterion and subsequent optimization. All issues are handled especially for three-dimensional data. Second part of this work is practical sample of MatLab® program environment for registration of acquired frames by affine transformation. In this program is algorithm for removal of stair-step artefact, under head pillow and patient's desk, too. As an optimization algorithm is used control random search (CRS) method. Due to medical images type, which are distributed in DICOM format, is included process for their load and save at this work.

KEYWORDS

affine transformation, computed tomography, interpolation method, similarity criterion, image registration, optimization method

CHMELÍK, Jiří *Afinní lícování nativních a postkontrastních CT snímků mozku*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2013. 55 s. Vedoucí práce byl Ing. Petr Walek

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Afinní lícování nativních a postkontrastních CT snímků mozku“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Tímto děkuji vedoucímu semestrálního projektu panu Ing. Petru Walkovi za veškerou ochotu a vstřícnost jakož i za odborné rady a nápady, které vedly k umožnění realizace této práce.

Brno

.....

(podpis autora)

OBSAH

1	Lícování obrazů	9
1.1	Geometrické transformace obrazu	9
1.2	Interpolace obrazu	14
1.3	Podobnostní kritéria	18
1.4	Optimalizační metody	21
1.4.1	Řízené náhodné prohledávání	22
1.4.2	Simulované žíhání	23
2	Realizace programu	25
2.1	Formát DICOM	25
2.2	Odstranění nežádoucích struktur	26
2.2.1	Pacientský stůl a podložka	26
2.2.2	Schodový artefakt	29
2.3	Geometrická transformace a interpolace obrazu	29
2.4	Podvzorkování obrazových dat	32
2.5	Algoritmus CRS	33
2.5.1	Heuristická funkce	35
2.5.2	Podmínka zastavení výpočtu	37
2.6	Grafické rozhraní	38
3	Výsledky	41
3.1	Uměle zkreslená data	41
3.1.1	Translace obrazu	41
3.1.2	Rotace obrazu	41
3.1.3	Škálování obrazu	44
3.1.4	Afinní transformace	45
3.2	Reálná data	45
4	Závěr	52
	Literatura	53
	Seznam symbolů, veličin a zkratk	55

SEZNAM OBRÁZKŮ

1.1	Transformace obrazu: původní obraz (a), translace 10 pixelů v ose x (b)	10
1.2	Transformace obrazu: původní obraz (a), rotace 10° roviny z (b) . . .	11
1.3	Transformace obrazu: původní obraz (a), zvětšení 1,3× v ose x (b) . .	12
1.4	Transformace obrazu: původní obraz (a), 10 pixelů zkosení xy (a) . . .	14
1.5	Ideální interpolace: interpolační jádro (a), frekvenční odezva (b) . . .	15
1.6	Interpolace nejbližší soused: interpolační jádro (a), frekvenční odezva (b)	17
1.7	Lineární interpolace: interpolační jádro (a), frekvenční odezva (b) . .	18
2.1	Prahování obrazu: originální obraz (a), naprahovaný obraz s prahem určeným Otsu metodou (b)	27
2.2	Histogramy obrazů: originální obraz (a), naprahovaný obraz (b) . . .	28
2.3	Vytvoření masky: hrubá binární maska (a), Binární maska (b)	29
2.4	Výsledný obraz s odstraněným patientským stolem a podložkou (a) a jeho histogram (b)	30
2.5	Odstranění schodového artefaktu	30
2.6	Ukázka provedené transformace	32
2.7	Ukázka použité vlnkové dekompozice	33
2.8	Blokové schéma programu	34
2.9	Schéma použité reflexe simplexu	36
2.10	Schéma použité diferenciální evoluce	36
2.11	Schéma použité perturbace	37
2.12	Ukázka výpočtu zastavovací podmínky	39
2.13	Grafické uživatelské rozhraní	40
3.1	Průběh podobnostního kritéria (translace x)	42
3.2	Průběh transformačního parametru translace	42
3.3	Translace v ose x	43
3.4	Průběh podobnostního kritéria (rotace xy)	44
3.5	Průběh transformačních parametrů rotace	45
3.6	Rotace v rovině xy	46
3.7	Průběh podobnostního kritéria (škálování x a y)	47
3.8	Průběh transformačních parametrů škálování	47
3.9	Škálování v osách x a y	48
3.10	Průběh transformačních parametrů (afinní)	49
3.11	Afinní transformace	50
3.12	Registrace reálných obrazů – obraz 1	51

ÚVOD

Rentgenová výpočetní tomografie je jednou z moderních metod užívaných v medicíně k získávání řezových snímků nejrůznějších částí lidského těla. Její výhodou je možnost rekonstrukce trojrozměrného nebo dokonce čtyřrozměrného zobrazení orgánů, kostí a dalších tkání. S využitím kontrastních látek je také možné získat nejrůznější snímky se zvýrazněnými zájmovými oblastmi.

V našem případě budou použity nativní (snímky bez kontrastní látky) a postkontrastní (snímky s kontrastní látkou) obrazy mozku, aby bylo možné za využití subtraktivní angiografie získat trojrozměrný obraz mozkových cév. Veškeré složky, které jsou ve scéně nepotřebné, rušivé nebo nadbytečné se dají považovat za šum. Z pohledu na původ tohoto šumu může být rozdělen do několika skupin. První skupinou je *fyzikální šum*, který je tvořen převážně aditivními šumy jako například kvantový šum, rozptýlené záření, kvantovací šum, elektronický šum, apod. Druhou skupinou je *pohybový šum*. Do této skupiny spadají pohybové artefakty, které vznikají například srdečním rytmem, náhodným pohybem pacienta, dýcháním, atd. Třetím typem je *strukturální šum*. Za strukturální šum se považují prakticky veškeré struktury, které jsou mimo oblast zájmu (v tomto případě všechny struktury mimo mozkových cév) [1], [6]. Při zobrazování mozkových cév jsou tedy strukturálním šumem ostatní mozkové tkáně, lebeční kosti, případně okolí pacienta.

Problémem tedy je, že během získávání dat se projeví všechny tyto tři šumové složky. Pohybový šum je způsoben faktem, že nativní a postkontrastní snímky jsou získávány v různém čase. Prakticky je nejdříve provedeno snímkování bez kontrastu a poté je pacientovi aplikována kontrastní látka. Jakmile je látka dostatečně rozložena ve sledovaném orgánu, je provedeno další snímkování s přítomností kontrastní látky. Během tohoto procesu se však pacient může pohnout, dýchá, případně dojde ke změně objemu tkáně v důsledku vpravené látky nebo psychického stavu pacienta. Důsledkem je, že snímky nebudou přesně lícovat a při jejich odečtení nezískáme pouze cévní řečiště s kontrastní látkou, ale také rušivé části ostatních tkání (strukturální šum). Je proto nutné provést předzpracování obrazových dat s využitím zejména oboru registrace obrazů a dosáhnout tak co nejlepšího slícování obou sad obrazů před samotnou fúzí. Dalším problémem je také značná výpočetní náročnost při zpracování takto objemných dat.

1 LÍCOVÁNÍ OBRAZŮ

Registrace je oblast zpracování obrazů, zabývající se hledáním ideálního překryvu obrazů, které byly pořízeny v různých časech, z rozdílných pohledů nebo různými detektory. Registrace geometricky upravuje obrazy tak, aby byly odpovídající si struktury na stejných prostorových souřadnicích. V zásadě může jít o obrazy pořízené z různých pohledů (složení několika obrazů do mozaiky, stereo-vize, ...). Další skupinou mohou být obrazy získané v rozdílném čase, kde jsou porovnány změny, které se v cílovém objemu dat udály během určité časové periody (sledování vývoje krajiny, postupu nemoci nebo jejího léčení, ...). Obrazy z různých senzorů, kdy je potřeba zobrazit více vlastností stejného objektu (anatomie a funkce orgánů v lékařství, radarové a fotografické snímky ve sledování počasí, ...). Poslední skupinou je porovnávání skutečných scén s jejich modely (srovnání orgánu s anatomickým atlasem, automatické hodnocení kvality, ...)[16].

Registrace obrazů může být celkově chápána jako iterativní proces zpracování obrazů, který v cyklu provádí následující kroky. Nejprve je proveden výpočet podobnostního kritéria lícovaných obrazů. Tato část procesu má zjistit míru podobnosti obrazů (tedy na kolik jsou si dané obrazy podobné). Následuje odhad transformačních parametrů na základě daného kritéria a je podle nich provedena geometrická transformace obrazu. Tímto krokem dostaneme určitým způsobem upravený (deformovaný) obraz, který by se měl více podobat obrazu referenčnímu. Dalším krokem je interpolace prostorových souřadnic transformovaného obrazu k odpovídající referenci. Tímto postupem je získán transformovaný obraz v prvním iteračním kroku. Tento obraz však stále nemusí být (a v převážné většině případů není) dostatečně slícován. Je proto nutné použít optimalizačních metod, které budou tento cyklus efektivně opakovat tak, aby po konečném počtu iterací bylo dosaženo kýženého výsledku slícování. V této práci jsou řešeny zejména metody pro optimalizaci globální intenzitní kritériální funkce. Pro optimalizaci však může být použita řada dalších metod jako optimalizace podle korespondenčních bodů nebo optimalizace globální povrchové kritériální funkce.

1.1 Geometrické transformace obrazu

Principem geometrických transformací obrazu je převedení původního obrazu přemapováním jeho prostorových souřadnic na obraz výstupní. Hodnoty, které se nacházejí na daných souřadnicích vstupního obrazu tedy zůstávají stejné i v obrazu výstupním, avšak jejich souřadnice jsou rozdílné. Geometrické transformace mohou být lineární a nelineární, přičemž při lineárních transformacích je zachována přímost

a rovnoběžnost čar, případně rovinnost ploch (v případě trojrozměrného zobrazení) v obraze. Nelineární transformace mohou způsobit porušení rovnoběžnosti nebo zakřivení čar, případně ploch. Dále můžeme rozlišit transformace podle oblasti aplikace na globální a lokální, kde globální transformace jsou prostorově invariantní (tedy pro každou prostorovou souřadnici mají stejnou transformační funkci). Lokální transformace mohou pro různé části obrazu použít transformační funkce s různými parametry. Jak již bylo naznačeno výše, je možné transformovat jak obrazy dvourozměrné (dvě prostorové souřadnice x, y), tak trojrozměrné (navíc souřadnice z). Je patrné, že pro trojrozměrné obrazy bude transformační rovnice rozšířena oproti rovnici pro dvourozměrné obrazy. Pro pozdější použití budou veškeré následující transformace definovány pro 3D scény.

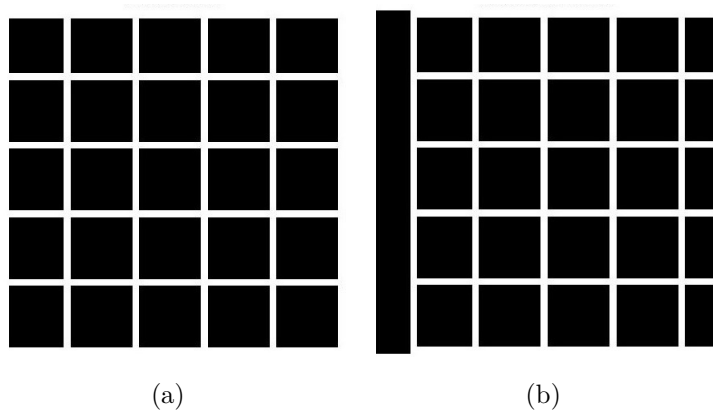
První skupinou jsou takzvané tuhé (*rigid*) transformace, které zachovávají geometrický tvar původního obrazu a pouze posouvají transformovaný obraz vůči původnímu v různých směrech oproti počátku souřadné soustavy. Tyto operace jsou lineární.

Nejjednodušším případem je posunutí/translace (*plain shift*), které je získáno pouhým přičtením konstanty k odpovídajícím původním prostorovým souřadnicím

$$\mathbf{r}' = \mathbf{r} + \Delta\mathbf{r}, \quad (1.1)$$

kde \mathbf{r}' je poziční vektor transformovaných prostorových souřadnic, \mathbf{r} je poziční vektor původních prostorových souřadnic a $\Delta\mathbf{r}$ je vektor rozdílu prostorových souřadnic. Po rozepsání pro trojrozměrný prostor dostaneme

$$\mathbf{r}' = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}. \quad (1.2)$$



Obr. 1.1: Transformace obrazu: původní obraz (a), translace 10 pixelů v ose x (b)

Další tuhou transformací je rotace obrazu (*rotation transform*). V tomto případě dochází k otáčení obrazu okolo určité osy. Ve 3D prostoru tedy kolem jedné z os x, y, z . Velikost otočení je dána úhlem otočení $\theta_x, \theta_y, \theta_z$. Transformace prostorových souřadnic je dána násobkem

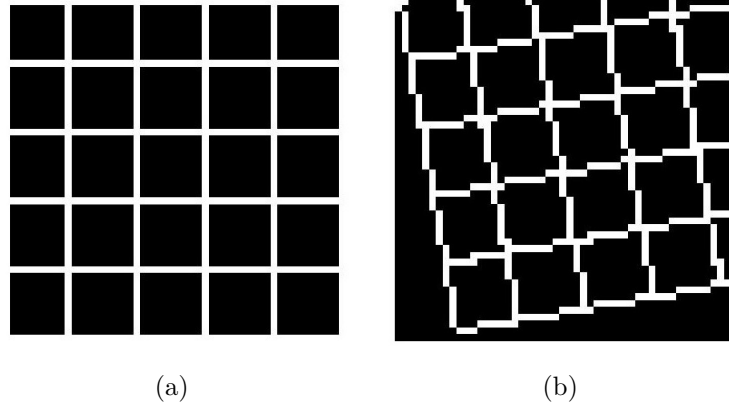
$$\mathbf{r}' = \mathbf{B}\mathbf{r}, \quad (1.3)$$

kde \mathbf{B} je transformační matice, jejíž tvar je definován jako

$$\mathbf{B} = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix}. \quad (1.4)$$

Roznásobením získáme výslednou matici \mathbf{B} , přičemž jednotlivé koeficienty pro zjednodušení zápisu můžeme označit jako b_{ik}

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}. \quad (1.5)$$



Obr. 1.2: Transformace obrazu: původní obraz (a), rotace 10° roviny z (b)

Sloučením obou předchozích transformací získáme obecnou tuhou transformaci (*generic rigid transform*)

$$\mathbf{r}' = \mathbf{B}\mathbf{r} + \Delta\mathbf{r}. \quad (1.6)$$

Abychom získali tento vztah vyjádřený pomocí jediné matice \mathbf{R} , musíme nahradit klasický vektor souřadnic rozšířeným vektorem homogenních souřadnic

$$(x, y, z) \rightarrow (cx, cy, cz, c)^T,$$

kde c je libovolná reálná konstanta, přičemž zvolíme $c = 1$. Po této úpravě získáme vztah

$$\mathbf{r}' = \mathbf{R}\mathbf{r} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & \Delta x \\ b_{21} & b_{22} & b_{23} & \Delta y \\ b_{31} & b_{32} & b_{33} & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (1.7)$$

Je vidět, že pro definování trojrozměrné tuhé transformace je zapotřebí 12 parametru (b_{ik} a $\Delta x, \Delta y, \Delta z$). Z předchozích výpočtů ale vyplývá, že parametry b_{ik} jsou vzájemně závislé přes úhly otočení ($\theta_x, \theta_y, \theta_z$), a proto můžeme říci, že prostá trojrozměrná tuhá transformace je definována pouze šesti parametry.

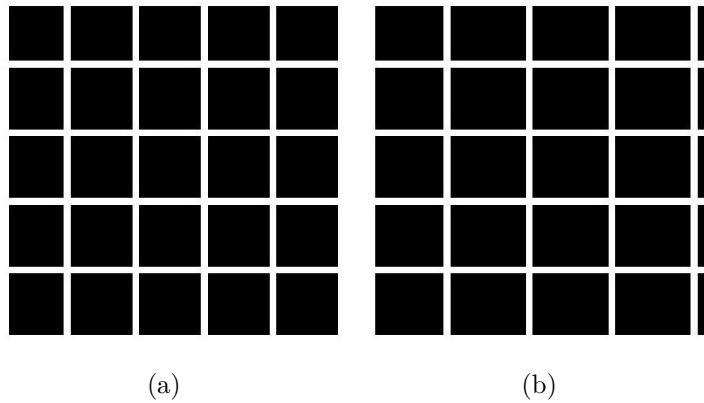
Druhou skupinou jsou pružné (*flexible*) transformace. Můžeme si představit, že obraz je natištěn na pružném podkladu, který můžeme natahovat nebo zužovat. Touto představou si můžeme přiblížit funkci pružných transformací. Pružné transformace mohou být jak lineární, tak nelineární.

Mezi lineární pružné transformace patří změna měřítka/škálování (*plain scaling*),

$$\mathbf{r}' = \mathbf{S}\mathbf{r}, \quad (1.8)$$

kde

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.9)$$



Obr. 1.3: Transformace obrazu: původní obraz (a), zvětšení $1,3\times$ v ose x (b)

Pokud budou všechny hodnoty s_i stejné (tedy změna měřítka bude ve všech směrech stejná - isotropní), bude docházet pouze k jednoduchému zvětšování nebo zmenšování obrazu. V opačném případě bude obraz zkreslen změnou poměrů.

Spojením obecné tuhé transformace a změny měřítka povede k obecné měřítkové transformaci (*generic scaling transformation*)

$$\mathbf{r}' = \mathbf{SRr}. \quad (1.10)$$

Pokud si vyjádříme matice \mathbf{SR} a dosadíme parametry, získáme

$$\mathbf{SR} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & \Delta x \\ b_{21} & b_{22} & b_{32} & \Delta y \\ b_{31} & b_{32} & b_{33} & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.11)$$

Roznásobením těchto matic získáme tvar:

$$\mathbf{SR} = \begin{bmatrix} s_x b_{11} & s_x b_{12} & s_x b_{13} & s_x \Delta x \\ s_y b_{21} & s_y b_{22} & s_y b_{23} & s_y \Delta y \\ s_z b_{31} & s_z b_{32} & s_z b_{33} & s_z \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.12)$$

V případě, že je změna měřítka isotropní, získáme jednodušší podobnostní transformaci (*similarity transformation*).

Další lineární pružnou transformací je zkosení obrazu (*image shearing*), která postupně posouvá jednotlivé sloupce, řádky nebo vrstvy podle osy určité osy v obrazové matici. Je definována jako

$$\mathbf{G} = \mathbf{G}_x \mathbf{G}_y \mathbf{G}_z, \quad (1.13)$$

kde po rozepsání matic získáme

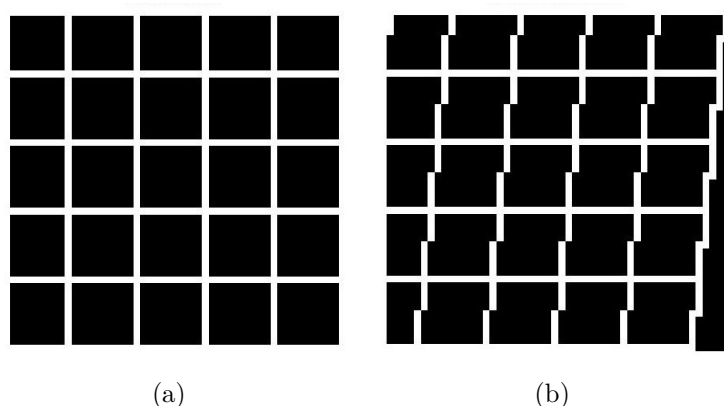
$$\mathbf{G} = \begin{bmatrix} 1 & g_{xy} & g_{xz} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ g_{yx} & 1 & g_{yz} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ g_{zx} & g_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.14)$$

Nejobecnější lineární obrazovou transformací je afinní transformace *affine transformation*, která slučuje všechny výše popsané lineární transformace, tedy posunutí, rotaci, změnu měřítka a zkosení obrazu. Výsledná transformace je rovna

$$\mathbf{r}' = \mathbf{GSRr} = \mathbf{Ar}, \quad (1.15)$$

Dále můžeme transformační matici \mathbf{A} přepsat s obecnými parametry takto:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & t_1 \\ a_{21} & a_{22} & a_{23} & t_2 \\ a_{31} & a_{32} & a_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.16)$$



Obr. 1.4: Transformace obrazu: původní obraz (a), 10 pixelů zkosení xy (a)

Afinní transformace je definována 12 na sobě nezávislými parametry. Rovněž zachovává rovinnost ploch, přímost čar a rovnoběžnost mezi nimi. Tato transformace je často užívána jako maximálně pružná lineární transformace, protože zachovává výhody jednoduchého interpretace právě díky linearitě.

Dalšími pružnými transformacemi jsou nelineární transformace, mezi něž patří například projektivní transformace (*projective transformation*), jež stále zachovává rovinnost ploch a přímost čar, avšak porušuje jejich rovnoběžnost. Ostatní transformace jsou zakřivující (*curved*); tedy deformují plochy i přímky. Mezi tyto transformace patří například polynomiální transformace (*polynomial transforms*), které nový poziční vektor počítají s využitím koeficientů polynomu. Problémem polynomiálních transformací vyšších řádů je jejich tendence k falešným oscilacím, které mohou vést až k nežádoucímu lokálnímu zvlnění a ohybům obrazu. Dalšími transformacemi mohou být například transformační funkce s radiální bází (*radial basis functions*), jako například funkce *thin plate spline*. Problematika rozboru geometrických transformací obrazu byla čerpána z uvedené literatury [2], [3] a [16].

1.2 Interpolace obrazu

Při aplikaci geometrických transformací může dojít k vypočtení nových souřadnic tak, že budou ležet mimo diskretní souřadnou síť (např. posun o neceločíselnou hodnotu, rotace, apod.). V tomto případě je nutné provést interpolaci těchto bodů do souřadné sítě, protože pro zobrazení takového obrazu je nutné mít diskretní obraz v prostoru (pole pixelů - 2D, pole voxelů - 3D). Nesprávnou aproximací však mohou být ve výsledném obrazu způsobena různá zkreslení a artefakty. Kvůli možnosti získání výsledku v diskretních souřadnicích srovnatelným s výsledkem získaným ve spojitém obrazu, je definována *jasovou interpolací obrazu*. Účelem této interpolace je co

nejpřesněji získat jasové hodnoty na určitých prostorových souřadnicích, které jsou však rozdílné od diskrétní souřadné sítě získané vzorkováním.

Pokud budou diskrétní obrazová data na ekvidistantní čtvercové síti, jejíž uzly budou jednotlivé vzorky, a také bude dodržena vzorkovací teorém, je definován obecný tvar interpolační rovnice. Interpolovaná hodnota potom bude

$$f_r(x, y, z) = \sum_i \sum_k \sum_j f(k\Delta x, i\Delta y, j\Delta z) r(x - k\Delta x, y - i\Delta y, z - j\Delta z), \quad (1.17)$$

kde

$$r(x, y, z) = \frac{u_m v_m w_m}{\pi^3} \frac{\sin(u_m x)}{u_m x} \frac{\sin(v_m y)}{v_m y} \frac{\sin(w_m z)}{w_m z},$$

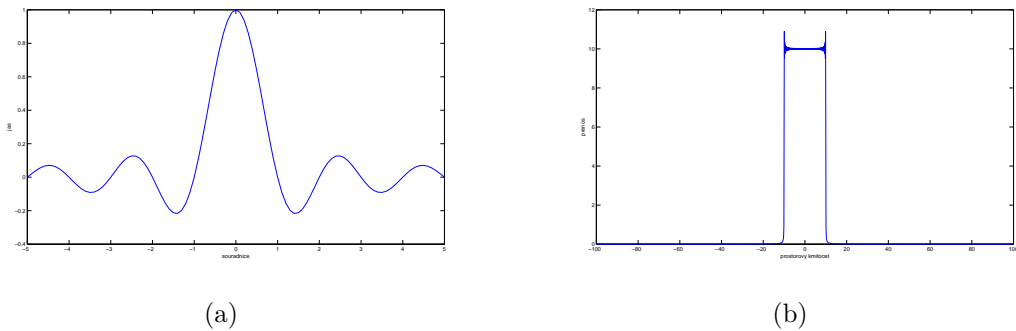
přičemž

$$u_m = \frac{\pi}{\Delta x}, \quad v_m = \frac{\pi}{\Delta y}, \quad w_m = \frac{\pi}{\Delta z}.$$

Funkce $f_r(x, y, z)$ tedy popisuje nově vypočtené hodnoty, kde $f(x, y, z)$ je původní obraz a $r(x, y, z)$ je označena jako interpolační jádro. Je patrné, že interpolační jádro může být rozděleno na funkce $\text{sinc}(c_x x)$, $\text{sinc}(c_y y)$ a $\text{sinc}(c_z z)$. Protože tyto funkce nabývají hodnoty 1 pouze na místě odpovídajícího vzorku a jsou nulové v místech ostatních vzorků, potom rovnice splňuje interpolační podmínku

$$f_r(k\Delta x, i\Delta y, j\Delta z) = f(k\Delta x, i\Delta y, j\Delta z). \quad (1.18)$$

Výsledek rovnice 1.17 také může být chápán jako konvoluce, tedy jako výstup lineárního filtru s impulsní odezvou $r(x, y, z)$, která odpovídá ideální dolní propusti s mezními frekvencemi u_m , v_m a w_m . Modelová ukázka ideálního jednorozměrného interpolačního jádra a jeho frekvenční odezva může být znázorněna následovně (Obr.: 1.5). Pro každý rozměr by tedy byl zapotřebí obdobný filtr.



Obr. 1.5: Ideální interpolace: interpolační jádro (a), frekvenční odezva (b)

Ve skutečnosti je však funkce $\text{sinc}()$ velmi složitá, protože teoreticky pokrývá celou oblast interpolovaného obrazu. Takovýto výpočet by byl velmi složitý a časově náročný. Ani tak by nikdy nebylo dosaženo přesného výpočtu, protože funkce má teoreticky nekonečnou podporu. Musí být proto nalezeno jednodušší vyjádření, které však zachová relativně dobrou přesnost. Obecně jsou na výpočet takového požadavky:

- dobrá přesnost
- pouze malé okolí počítaného bodu vstupuje do výpočtu
- co nejjednodušší interpolační jádro
- spojitě a co nejhladší prostředí v celém obrazu
- propouštět všechny kmitočty splňující Nyquistovu podmínku a co nejvíce potlačit kmitočty, které ji přesahují

Žádná aproximace nemůže optimálně splnit všechny tyto požadavky. Proto jsou voleny kompromisy mezi nimi v závislosti na použití interpolace.

Nejjednodušší aproximační metoda je interpolace metodou *nejbližšího souseda* (NN). V trojrozměrném prostoru je interpolační pole tvořeno krychlovou oblastí mezi osmi sousedními vzorky. Hodnota vypočtená touto metodou může být zapsána takto

$$_{NN}f_r(x, y, z) = f_{\mathbf{r}} = f((k+l)\Delta, (i+m)\Delta, (j+n)\Delta), \quad l, m, n \in \{0, 1\}, \quad (1.19)$$

kde

$$[l, m, n] = \frac{\mathbf{r}_n - \mathbf{r}_0}{\Delta}, \quad \mathbf{r}_0 = [k, i, j] \Delta, \quad \mathbf{r}_n = \arg \min_n |\mathbf{r} - \mathbf{r}_n|, \quad n = 0, 1, \dots, 7.$$

Označení \mathbf{r}_0 je bod nejbližší ke středu souřadného systému a \mathbf{r}_n je nejbližší bod vzorkovací mřížky. Oba dva vektory popisují některý z 8 bodů kolem hledané pozice \mathbf{r} . Výsledná funkce není spojitá (přechod z maxima do nuly). Odezva ve frekvenční oblasti má tvar funkce sinc s mnoha postranními laloky nad Nyquistovou frekvencí, ale zároveň se zúženým lalokem hlavním (Obr.: 1.6).

Z toho vyplývá, že budou propuštěny frekvence nad polovinou vzorkovacího kmitočtu, které budou způsobovat aliasingové zkreslení. Navíc kvůli zúžení hlavního laloku budou částečně potlačovány užitečné kmitočty. To povede ke zhoršení kontrastu v detailech a tzv. “schodovitosti”. Z těchto důvodů by měla tato metoda být používána pouze v případech, kdy je obraz dostatečně nadvzorkován, v posledních krocích zpracování k předběžnému zobrazení, nebo v jednotlivých iteračních krocích zpracování jako jednoduchý odhad, který bude později dále upraven některou přesnější metodou.

Značně lepší, i když výpočetně náročnější je *bilineární interpolace* (BL) (nebo *trilineární* (TL) ve 3D prostoru). Interpolace se skládá ze 2 (3) vzájemně kolmých

jednorozměrných lineárních transformací s lineárním interpolačním jádrem (Obr.: 1.7).

Touto metodou ztratíme mnohem více informací na vysokých frekvencích než s NN, ale funkce je spojitá a aliasingové zkreslení je zde lépe potlačeno. Trilineární interpolace může být formulována jako

$${}_{TL}f_r(x, y, z) = f_r \mathbf{r} = \sum_{l=0}^1 \sum_{m=0}^1 \sum_{n=0}^1 f((k+l)\Delta, (i+m)\Delta, (j+n)\Delta) c_{l,m,n}, \quad (1.20)$$

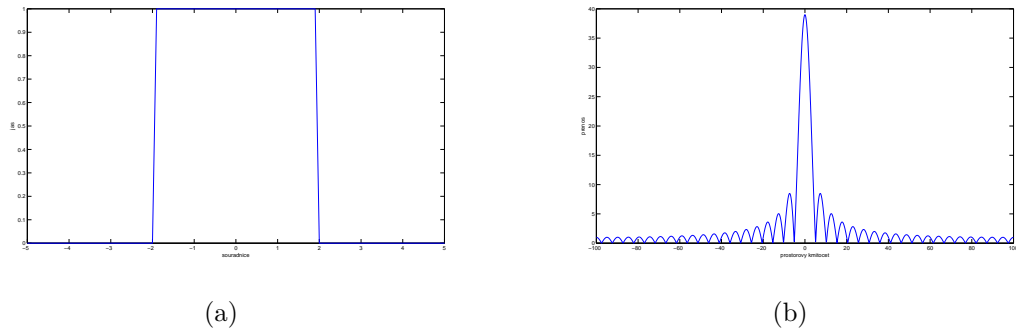
kde

$$\mathbf{r}_0 = [l, m, n] \Delta, \quad [d_x, d_y, d_z] = \frac{\mathbf{r} - \mathbf{r}_0}{\Delta}, \quad \mathbf{c} = \begin{bmatrix} (1-d_x)(1-d_y) & (1-d_x)d_y \\ d_x(1-d_y) & d_x d_y \end{bmatrix},$$

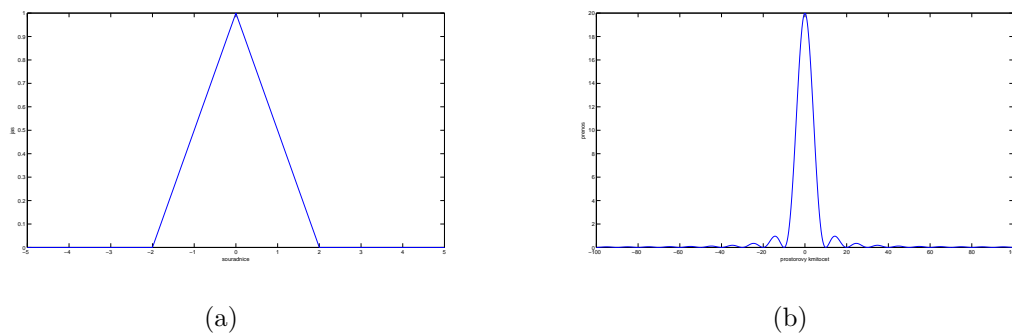
$$\mathbf{c}_{n=0} = (1-d_z)\mathbf{c}, \quad \mathbf{c}_{n=1} = d_z\mathbf{c}.$$

Z pohledu originálního prostoru je trilineární interpolace spojitá na okrajích jednotlivých interpolačních oblastí, nicméně derivace jsou zpravidla nespojité. Je vhodné si uvědomit, že TL funkce netvoří čtvercové roviny, ale spíše zakřivený prostor. Derivace jsou nespojité na okrajích, protože jsou brány v potaz pouze hodnoty nejbližších osmi vzorků. Pro zobrazení je již tato interpolace vcelku přijatelná, protože odstraňuje, “schodovitost”, která byla velmi výrazná při použití metody NN. Nicméně interpolovaný obraz je značně rozmazán. Je možné dosáhnout lepší ostrosti při použití složitějších metod.

Přesnějšími metodami jsou *splajnové interpolace*, mezi něž mohou být zařazeny interpolace *kubické* (bikubické pro 2D a trikubické pro 3D). Tyto interpolace značně eliminují rozmazání obrazu oproti předchozím metodám, avšak jsou velmi výpočetně náročné. Interpolace zaručují spojitost druhých derivací na okrajích interpolačních oblastí. Pro porovnání, bikubické interpolace využívají 16 okolních vzorků, trikubické 64 okolních vzorků. Výpočet těchto interpolací spočívá v aproximaci výchozí



Obr. 1.6: Interpolace nejbližší soused: interpolační jádro (a), frekvenční odezva (b)



Obr. 1.7: Lineární interpolace: interpolační jádro (a), frekvenční odezva (b)

jasové funkce polynomem třetího řádu. Nevýhodou je generace jasových hodnot za hranicemi povolené jasové meze, které nespádají do původní jasové funkce. Rozbor interpolačních metod byl čerpán ze zdrojů [3], [9] a [10].

1.3 Podobnostní kritéria

Úkolem podobnostních kritérií je určit míru podobnosti obrazů nebo jejich oblastí. K určení podobnosti dvou nebo více obrazů během slícování pomocí geometrických transformací se používá podobnostní kritérium v lokálním, ale také v globálním měřítku. V případě rozpoznávání vzorů nebo textur se využívá semiglobálního kritéria, které pokrývá určité oblasti v obraze. Pro disparitní analýzu musí být podobnost počítána lokálně, zabírající co nejmenší oblast obrazu, aby byl získán lokální disparitní vektor mezi pozicemi odpovídajících vlastností v porovnávaných obrazech. Globální podobnosti mohou být, až na několik výjimek založeny na lokálních kritériích. Dají se zjednodušeně odvodit z lokálních kritérií jako jejich součet, integrál, průměr, atd. Lokálního kritéria se také používá v případě hledání shody mezi jednotlivými páry ve skupině značek (*landmarks*) nebo pro vylepšení ručně přiřazených hrubých shod. Lokální kritéria jsou proto důležitá pro metody fúze obrazu. Kritéria mohou být založena na údajích o intenzitě nebo na informačním obsahu obrazů. Výše popsané postupy se řadí do skupiny tzv. intenzitních podobností (*intensity similarity*), přičemž může být porovnávána kromě intenzity také barva (intenzita jednotlivých barevných složek v obraze). Kritéria založená na informačním obsahu obrazů využívají výpočtu sdruženého histogramu, který znázorňuje četnost výskytu dvojic intenzit na shodných pozicích v obou obrazech. Samotný histogram však ještě nedefinuje míru podobnosti dvou obrazů. Informaci, kterou obsahuje tento histogram však může být přibližně přepsán jako pravděpodobnost výskytu dvojice hodnot na konkrétních souřadnicích v obou porovnávaných obrazech

$$p_{AB} \{l, m\} \approx \frac{1}{N} h_{l,m}^{AB}, \quad (1.21)$$

přičemž l, m jsou jasové hodnoty v obrazech. Obdobně budou definovány i pravděpodobnosti výskytu konkrétní jasové hodnoty na konkrétní souřadnici pro každý samostatný obraz

$$p_A \{l\} \approx \frac{1}{N} h_l^A, \quad (1.22)$$

$$p_B \{m\} \approx \frac{1}{N} h_m^B. \quad (1.23)$$

K zjištění vztahu mezi obrazy, musí být nejprve určena takzvaná *entropie obrazu*, což je jednoduše řečeno “míra neuspořádanosti”.

$$H_A = - \sum_{l=1}^q p(a_l) \log p(a_l), \quad (1.24)$$

Opět obdobně můžeme vyjádřit další entropii

$$H_B = - \sum_{m=1}^r p(b_m) \log p(b_m). \quad (1.25)$$

Dále může být definována *sdrúžená entropie*

$$H_{AB} = - \sum_{l=1}^q \sum_{m=1}^r p(a_l, b_m) \log p(a_l, b_m), \quad (1.26)$$

kde q, r jsou maximální hodnoty jasu v obrazech A, B a pravděpodobnosti $p(a_l)$, $p(a_l)$ a $p(a_l, b_m)$ odpovídají pravděpodobnostem z dříve uvedených rovnic (Rov.: 1.22, 1.23, 1.21. Na základě takto odvozené entropie, může být vyjádřen vztah pro *vzájemnou informaci (MI)*

$$I_{AB} = H_A + H_B - H_{AB}. \quad (1.27)$$

Vzájemná informace vyjadřuje míru závislosti porovnávaných dat. Hodnota vzájemné informace bude tedy maximální právě tehdy, budou-li oba srovnávané obrazy naprosto stejné, a tedy také úplně vzájemně závislé. V tomto případě bude hodnota vzájemné informace dána vztahem

$$I_{AB} = H_A = H_B = H_{AB} \quad (1.28)$$

Naopak, pokud budou obrazy rozdílné a navzájem úplně nezávislé, bude hodnota vzájemné informace nulová [5].

Dále budou zmíněna podobnostní kritéria založená na údajích o intenzitě. Obecně podobnostní kritéria porovnávají intenzitu ve dvou obrazových oblastech

A a B. Pokud budou předpokládány diskrétní obrazy, mající konečný počet pixelů, mohou být oblasti popsány vektory \mathbf{a} a \mathbf{b} . Tyto vektory jsou tvořeny N hodnotami intenzit v jednotlivých pixelech, kde N je celkový počet pixelů v oblasti, přičemž vektory byly získány zvoleným pravidlem (např. skenováním řádků). Rozměry obou vektorů a tedy i oblastí musí být totožné. Pokud se porovnávají oblasti po geometrické transformaci, musí být zajištěna totožná velikost oblastí, a tedy kompatibilní rozměry vektorů.

Jednou z nejjednodušších kritérií nepodobnosti je *součet čtvercových hodnot rozdílů (SSD)*,

$$C_{SSD}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^N |a_i - b_i|^2. \quad (1.29)$$

Podobným kritériem, které vznikne nepatrnou úpravou je *Eukleidovská vzdálenost*,

$$C_E(\mathbf{a}, \mathbf{b}) = |\mathbf{a} - \mathbf{b}| = \sqrt{\sum_{i=1}^N (a_i - b_i)^2}. \quad (1.30)$$

V obou případech je možné použít normalizaci podle N , aby bylo možno porovnat hodnoty v případě, že by oblasti měly rozdílné rozměry. Avšak principiálně je tento krok zbytečný, protože obě oblasti musí mít stejný rozměr. Obě kritéria popisují nepodobnost, tedy pokud budou obě oblasti přesně slícované a oba obrazy budou totožné, kritérium bude rovno nule. Tyto funkce zdůrazňují především větší nepodobnosti, které jsou symbolizovány hodnotou vždy větší než nula. Při menších rozdílech nebo při zatížení obrazů šumem však tato kritéria mohou selhat.

Další metodou je *kosinové kritérium* nebo také *úhlové kritérium*, které vyjadřuje kosinus úhlu mezi obrazovými vektory \mathbf{a} a \mathbf{b} . Tato funkce je výhodná zejména díky své dobré toleranci vůči lineárním změnám kontrastu a částečné toleranci vůči změnám nelineárním. Výstup může nabývat hodnot z intervalu $\langle -1; 1 \rangle$, přičemž při minimálním rozdílovém úhlu nabývá kosinus maximální hodnoty. Pro případ porovnávání obrazů může být řečeno, že výsledek bude pouze v intervalu $\langle 0; 1 \rangle$, protože hodnoty intenzit nenabývají záporných hodnot. Další výhodou této funkce je také nízká výpočetní náročnost. Výpočet je dán vztahem

$$C_A(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2} \sqrt{\sum_i b_i^2}}. \quad (1.31)$$

Pokud je z každého koeficientu odečtena střední hodnota vektorů, je získáno kritérium zvané *korelační koeficient*. Výhodou je invariantnost vůči lineární změně kontrastu, avšak je o něco více výpočetně náročné než kosinové kritérium. Výstup opět nabývá hodnot $\langle -1; 1 \rangle$, kde 0 značí nezávislé veličiny a ± 1 veličiny závislé. Rovnice pak nabývá tvaru:

$$C_{CC}(\mathbf{a}, \mathbf{b}) = \frac{\sum_i (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_i (a_i - \bar{a})^2 \sum_i (b_i - \bar{b})^2}}, \quad (1.32)$$

kde \bar{a} a \bar{b} jsou střední hodnoty

$$\bar{a} = \frac{1}{N} \sum_i a_i, \quad \bar{b} = \frac{1}{N} \sum_i b_i.$$

Čitatel z výše uvedené rovnice (Rov.: 1.32) se nazývá *kovarianční koeficient*

$$C_{CV}(\mathbf{a}, \mathbf{b}) = \sum_i (a_i - \bar{a})(b_i - \bar{b}). \quad (1.33)$$

Aby bylo možné výše uvedené podobnostní funkce požit také v globálním měřítku, musí být stále na paměti, že je třeba zachovat odpovídající rozměry obou porovnávaných obrazů. Také je nutné si uvědomit, že pokud je měřeno podobností kritérium pro dosažení co nejlepšího slícování s využitím geometrických transformací, musí být brán ohled na rozdílné interpolační výsledky po každém kroku registrace obrazu. Z toho vyplývá, že přesnost výpočtu podobnostního kritéria bude záviset také na použité geometrické transformaci a následné interpolační metodě. Informace z této kapitoly pochází ze zdrojů [3], [7] a [10].

1.4 Optimalizační metody

Složitější podobnostní funkce mohou vést na soustavu nelineárních rovnic, které jsou obtížně řešitelné. Obecně se optimalizační algoritmus snaží nalézt geometrickou transformační funkci T_α , přinášející nejlepší shodu tak, že maximalizuje (nebo minimalizuje) hodnotu globální podobnostní funkce. Jakmile je zvolen druh transformace, optimalizační metoda se zaměří na získání nejvhodnějšího vektoru α transformačních parametrů geometrické transformace [10].

Ve skutečnosti je registrační proces poněkud složitější vzhledem ke dvěma jevům. Prvním je pouze částečný překryv (Ω_α) transformovaného (A') a originálního obrazu (B). Druhým pak nutnost interpolace mezi známé body netransformovaného (A) obrazu. Proto musí být v každém optimalizačním kroku provedeny následující děje:

- nastavení vektoru α podle zvoleného optimalizačního postupu, využití údajů z předchozího průběhu optimalizace
- výpočet pozičních souřadnic \mathbf{x}_A (vektor souřadnic, pro 3D x,y,z) odpovídajících s pozicemi sítě uzlů v transformovaném obrazu $A' = T_\alpha(A)$
- určení oblasti překryvu Ω_α
- výpočet interpolovaných hodnot $A'(T_\alpha(\mathbf{x}_A))$ založených na bodech A
- výpočet výsledné funkce v závislosti na zvoleném podobnostním kritériu

- testování, zda bylo dosaženo optima; pokud ano, následuje ukončení algoritmu se současnými hodnotami α považovaným za α_0 , jinak pokračování od prvního bodu

Jednotlivé kroky se mohou vzájemně překrývat nebo probíhat paralelně. Aby nemusely být postupně procházeny všechny možné kombinace nastavení transformační funkce α (velká časová náročnost), je vhodné použít některou z později zmíněných optimalizačních metod.

Obecně mohou být optimalizační metody rozděleny na stochastické a deterministické, přičemž deterministické metody k nalezení nového stavu využívají znalosti o předešlém stavu a jeho vlastnosti vůči okolním bodům. Stochastické metody naopak proti tomu užívají pro nalezení nového stavu náhodných procesů založených na teorii pravděpodobnosti. Používané deterministické optimalizační funkce jako například *hledání gradientního sestupu*, *Newtonova metoda*, *Quasinevtonova metoda* a *metoda reflexe simplexu*. Mezi stochastické optimalizační funkce patří *řízené náhodné prohledávání* a *simulované žíhání*. Dále se budeme zabývat právě stochastickými metodami [3].

Rozhodnutí, zda má být optimalizační metoda ukončena, může být určeno několika možnostmi. Prvním typem je ukončení optimalizace dosažením maximálního počtu cyklů (používá se jako pojistka proti zacyklení programu při nedosažení odpovídajícího optima). Druhou možností je testování, zda dochází v průběhu cyklů ke zkvalitňování kritéria. Pokud nastane situace, kdy se během několika po sobě jdoucích iterací již výsledné kritérium příliš neliší, dojde k zastavení optimalizačního algoritmu. Problémem však je, že výpočet nemusí vždy dosáhnout globálního minima, ale může skončit v minimu lokálním.

1.4.1 Řízené náhodné prohledávání

Řízené náhodné prohledávání (CRS) je jednoduchý stochastický algoritmus pro hledání globálního extrému. V principu funguje algoritmus tak, že na jeho počátku je vygenerována populace P , která je tvořena N body v prohledávaném prostoru D . Následně je podle definované heuristické funkce vygenerován nový bod y . Zároveň je z populace odstraněn bod, který je nejhorší (z hlediska zvolené vlastnosti, kritéria, apod.) [10]. Podle použité heuristiky (může být použito i více heuristických funkcí) jsou nové body generovány stále blíže hledanému extrému. Algoritmus je zastaven ve chvíli, kdy již není možné vytvořit bod s vyšší (nižší) hodnotou, poslední nejlepší bod je pak brán jako výsledek prohledávání. Další podmínkou zastavení algoritmu může být dosažení maximální oblasti, ve které se nacházejí všechny body z vygenerované populace. Ukončení také může být způsobeno dosažením maximálního počtu kroků. Mezi základní heuristické funkce patří *reflexe simplexu*, *reflexe sim-*

plexu se zahrnutím nejlepší hodnoty, diferenciální evoluce nebo genetické algoritmy. Například při použití reflexe simplexu bude postup algoritmu následující:

1. generace $d + 1$ nekoplanárních bodů \mathbf{x}_n z množiny N - simplex S ,
2. nalezení bodu \mathbf{x}_H ze simplexu S s největší funkční hodnotou,
3. výpočet těžiště simplexu \mathbf{g} ,

$$\mathbf{g} = \frac{1}{d} \left(\sum_{n=1}^{d+1} \mathbf{x}_n - \mathbf{x}_H \right) \quad (1.34)$$

4. následuje reflexe simplexu (překlopení \mathbf{x}_H přes \mathbf{g}),

$$\bar{\mathbf{y}} = \bar{\mathbf{g}} + (\bar{\mathbf{g}} - \bar{\mathbf{x}}_H) = 2\bar{\mathbf{g}} - \bar{\mathbf{x}}_H, \quad (1.35)$$

kde $\bar{\mathbf{y}}$ je vzdálenost nového bodu od počátku souřadnic, $\bar{\mathbf{g}}$ je vzdálenost těžiště od počátku souřadnic a $\bar{\mathbf{x}}_H$ je vzdálenost bodu s největší funkční hodnotou od počátku souřadnic,

5. pokud je funkční hodnota bodu \mathbf{y} menší než funkční hodnota bodu \mathbf{x}_{HP} s největší, funkční hodnotou z populace P , pak bude bod \mathbf{x}_{HP} nahrazen bodem \mathbf{y}
6. opakování v cyklu, dokud nebude splněna podmínka ukončení algoritmu.

Odkazy na informační zdroje o CRS jsou [11], [12], [13] a [10].

1.4.2 Simulované žíhání

Simulované žíhání (SA) je optimalizační metoda, která vznikla z modelu tvorby krystalu během chlazení materiálu. Zahřátý krystal, který pomalu chladne tvoří krystalickou mřížku tak, aby zaujal stav s co nejmenší konečnou energií. Tento postup je používán k opravě krystalických defektů, protože při zahřátí na vysokou teplotu defekty mizí s vysokou pravděpodobností. Následným dostatečně pomalým chlazením je zajištěna malá pravděpodobnost vzniku nových defektů při tvorbě krystalové mřížky [10].

Analogicky je v optimalizaci hledán nejnižší koeficient nepodobnosti dvou obrazů. Můžeme si tedy představit, že energie z předchozího popisu je nahrazena optimalizovanou funkcí $f(\mathbf{x})$, kde vektor \mathbf{x} představuje momentální stav. Na počátku je náhodně vygenerován nebo zadán vektor \mathbf{x} , který je brán jako počáteční stav. Nový stav \mathbf{x}_{n+1} je generován opět náhodně podle vztahu

$$\mathbf{x}_{n+1} = G(\mathbf{x}_n, T), \quad (1.36)$$

kde G je funkce náhodné generace nového stavu a T je zadaná počáteční teplota systému. Tento nový stav však nemusí být přijat. Jeho přijetí závisí na pravděpodobnostní funkci, která může pro jednoduchost vypadat takto

$$h(\Delta E, T) = e^{-\frac{\Delta E}{T}}, \quad (1.37)$$

přičemž ΔE je rozdíl energií mezi nynějším a novým stavem

$$\Delta E = f(\mathbf{x}_n) - f(\mathbf{x}_{n+1}).$$

Pravděpodobnostní funkce však může být v závislosti na použitém postupu všeo-
becně rozdílná od výše uvedené. Po několika průbězích náhodné generace a případ-
ného přijetí nového stavu snížíme teplotu systému podle ochlazovacího plánu. Tento
plán může být opět zvolen různě. Jedním z nejjednodušších je například tento:

$$T_n = \alpha T, \quad (1.38)$$

kde α je tzv. koeficient chlazení. Celý proces je zastaven, jakmile teplota dosáhne
přednastavené minimální hodnoty. V takovém případě je poslední známý stav brán
jako stav optimální [4].

Při aplikaci této metody na registraci obrazů s využitím geometrické transfor-
mace je jako vektor stavu použit vektor koeficientů transformační funkce a jako
optimalizovaná funkce je vzato kritérium nepodobnosti. Výsledkem je pak takový
vektor koeficientů pro nastavení transformační funkce, aby bylo dosaženo co nejlep-
šího slícování obrazů.

Tuto metodu je možné realizovat standardně s využitím cyklů nebo také jako
neuronové sítě (tzv. Boltzmanův stroj) [14]. Nalezení globálního minima závisí na
vhodném nastavení pravděpodobnostní funkce, zvolení počtu kroků generace a přiřa-
zení pro každou teplotu, volbě ochlazovacího plánu a koeficientu chladnutí. Nalezení
minima je vysoce pravděpodobné, avšak není zaručeno, protože algoritmus může
dospět do minima lokálního, ze kterého již nebude dále pokračovat.

2 REALIZACE PROGRAMU

Výstupem této práce je program v prostředí MatLab[®], který postupně provádí načítání jednotlivých snímků mozku ve formátu DICOM, které uloží do dvou trojrozměrných matic (nativní a kontrastní). Po nastavení interpolační metody a výběru podobnostního kritéria je automaticky provedeno odstranění schodového artefaktu, patientského stolu a podložky z obou snímků. Následuje samotná registrace obrazů s využitím algoritmu CRS. Nakonec program dovoluje uložit transformované snímky opět do formátu DICOM. Samotný program je rozčleněn na hlavní soubor obsahující volání jednotlivých funkcí a jádro CRS algoritmu (*DP.m*), uživatelské rozhraní (*DP_gui.m*) a samotné funkce, které zajišťují načtení snímků DICOM (*dicom_nacteni.m*), odstranění klínových filtrů a patientského stolu (*prahovani.m*), odstranění schodového artefaktu (*sequen_distort_correct.m*) a její podfunkce (*GradOptim2D.m*, *phase_corr_2D_translat.m*, *Translation2D.m* a *translation_3D.m*), geometrickou afinní transformaci a interpolaci (*transformace.m*), výpočet podobnostního kritéria (*similarity.m*), provedení reflexe simplexu (*refsimpl.m*) nebo diferenciální evoluci (*difev.m*), následnou perturbaci (*perturb.m*) a uložení upravených snímků (*dicom_ulozeni.m*), zobrazení výsledků (*zobr.m/zobr2.m*), případně ověření pomocí vzájemné informace (*micheck.m*, *mi.m*, *hist2.m*). Dále pro zobrazení průběhu výpočtu je nutná funkce (*ParforProgMon.m*) a stejnojmenná složka obsahující java skripty.

2.1 Formát DICOM

Formát DICOM je všeobecně používaný formát v nejrůznějších lékařských přístrojích. Samotný soubor se skládá ze dvou částí, kde v jedné je uložen získaný obraz a ve druhé jsou vloženy různé informace (od patientských dat, přes různá nastavení a typ přístroje použitého pro snímání obrazu, až po informace o obrazu samotném). Po spuštění programu je třeba vybrat první soubor pro načtení snímků a zadat počet snímků, jež má být načteno. Snímky mohou být uloženy jako nové DICOM soubory nebo pouze uloženy jako programová proměnná ve formátu **.mat*. Pokud mají být soubory uloženy také ve formátu DICOM, je třeba potvrdit uložení v dotazovém okně. Pro načítání snímků je použita funkce *dicom_nacteni.m*,

```
1 function [im,v] = dicom_nacteni(pocet_snimku,s)
```

jejíž vstupní hodnoty udávají počet načítaných snímků a název s cestou k souboru. Výstupem je pak trojrozměrná matice obsahující všechny načtené snímky a vektor udávající rozměry trojrozměrné matice. Načítání snímků probíhá v cyklu, kde je postupně měněna první číslice. Takto vzniklý řetězec je použit jako vstup příkazu *dicomread*.

```

1  parfor i = 1:1:v(3) % cyklus pro nacteni sekvence snimku
2      s1 = sprintf(s,i); % formatovani retezce (s obsahuje nazev a
        cestu prvnioho souboru)
3      s1 = regexprep(s1,'%',''); % odstraneni prebytecneho '%' z
        retezce
4      im(:, :, i) = dicomread(s1); % nacteni i-teho snimku a jeho
        ulozeni do im
5  end;

```

Obdobným postupem je prováděno také ukládání snímků pomocí funkce *dicom_ulozeni.m*, jen s tím rozdílem, že je použito příkazu *dicomwrite*.

```

1  function [] = dicom_ulozeni(im,v,NewPath,NewName)

```

Vstupní parametry jsou trojrozměrná matice snímků, vektor s údaji o velikosti této matice, cesta k uložení souborů a název souboru.

2.2 Odstranění nežádoucích struktur

Podstatným problémem registrace obrazu (zejména za použití registrace pomocí globální intenzitní kritériální funkce) je přítomnost struktur, které jsou mimo oblast zájmu a působí rušivě. Tyto struktury znemožňují korektní registraci obrazu zájmových oblastí. Příkladem může být patientský stůl. Během akvizice nativních a kontrastních obrazových dat zůstává patientský stůl na stejném místě, avšak poloha pacienta může být změněna. V praxi pak může dojít k registraci podle patientské stolu a nikoliv podle zájmové oblasti (mozku). Tím je výsledek registrace naprosto znehodnocen. Z tohoto důvodu je nutné odstranit co největší množství těchto struktur.

2.2.1 Patientský stůl a podložka

Problémem znemožňujícím registraci obrazů je přítomnost podložky pod hlavou a patientského stolu v obrazech. (Obr.: 2.1(a)). Proto je třeba tyto komponenty z obrazů odstranit. Tuto část provádí funkce *prahovani.m*,

```

1  function [imc] = prahovani(im,v)

```

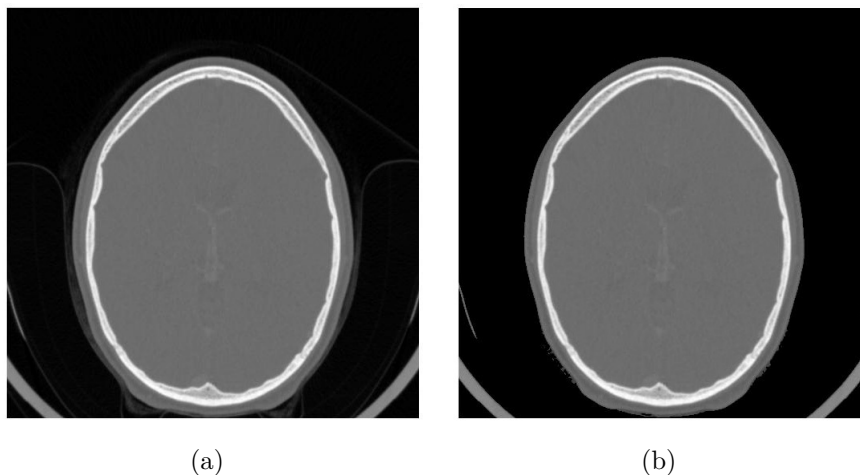
kde vstupem je trojrozměrná obrazová matice a její rozměry a výstupem je opět trojrozměrná obrazová matice s odstraněnými částmi. Zvolený postup odstranění spočívá ve dvou krocích. Prvním krokem je prosté prahování jednotlivých snímků s využitím Otsu metody k nalezení optimální hodnoty prahu [8]. Tato metoda je přímo zakomponována v programovém prostředí MatLab® jako příkaz *graythresh*.

```

1  prah = graythresh(im); % urceni hodnoty prahu Otsu metodou
2  imp = im; % ulozeni snimku do pomocne promenne imp
3  imp(imp<prah) = 0; % provedeni prahovani

```

Odstraněním všech hodnot nižších než takto určený práh ze snímků je dosaženo téměř dokonalého odstranění podložky (Obr.2.1(b)). Dále jsou znázorněny pro porovnání také histogramy obou obrazů (Obr. 2.2).



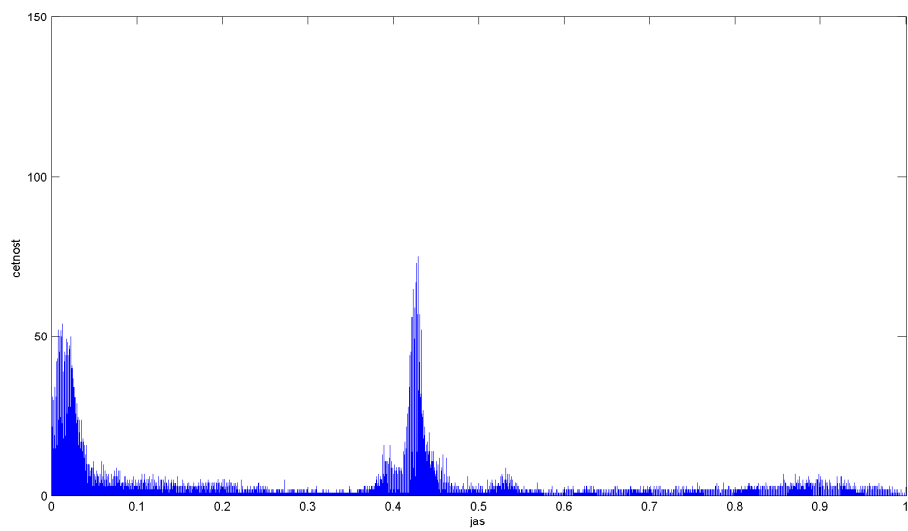
Obr. 2.1: Prahování obrazu: originální obraz (a), naprahovaný obraz s prahem určeným Otsu metodou (b)

Problém však nastává při prahování patientského stolu. Jeho jasové hodnoty totiž spadají do oblasti kostí, a tedy prahováním by byla odstraněna velká část lebky. Z tohoto důvodu je ve druhé fázi použit následující postup. Po prahování s prahem určeným metodou Otsu jsou převedeny naprahované snímky na binární obrazy. Tím vznikne binární maska, která stále obsahuje patientský stůl a části podložky (Obr. 2.3(a)).

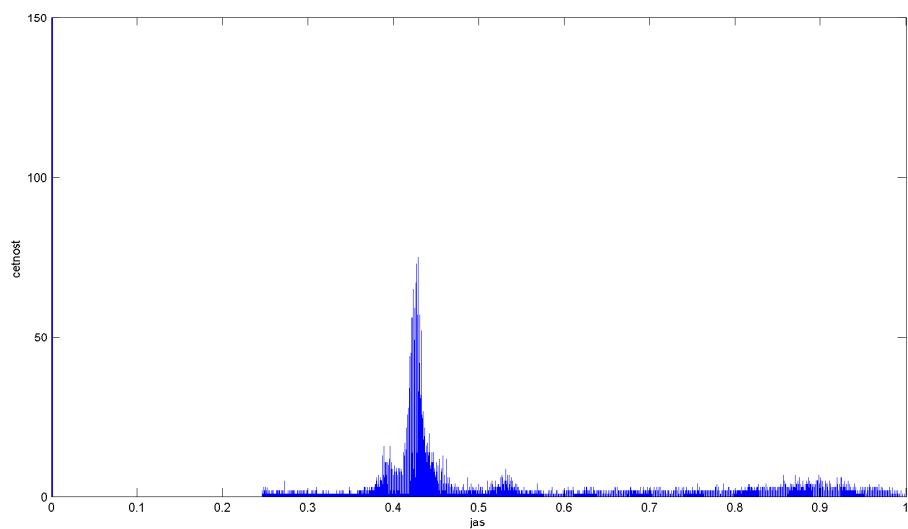
```
1   imin = im2bw(imp(:,:,i),prah); % převedení do binárního obrazu s urovní
    prahu
```

Nyní je provedeno očíslování jednotlivých spojených ploch této binární masky pomocí příkazu *bwboundaries*. Poté je příkazem *regionprops* proveden výpočet obsahů všech ploch. Protože je známo, že obraz mozku zabírá největší plochu snímku, může být řečeno, že binární objekt s největším obsahem představuje právě mozek. Ostatní menší plochy jsou tedy nežádoucí a mohou být odstraněny. Je tedy definována maska, která nabývá hodnoty jedna v oblasti mozku a v jeho okolí hodnoty nula (Obr. 2.3(b)).

```
1   [~,L] = bwboundaries(imin,'noholes'); % ocislovani jednotlivych ploch
2   S = regionprops(L,'ConvexArea'); % vypocet obsahu ploch
3   S = [S.ConvexArea]; % ulozeni hodnot obsahu ze struktury
4   Smax = max(S); % zjisteni nejvetsi plochy
5   hodnota = find(S == Smax); % zjisteni indexu nejvetsi plochy
6   L(L ~= hodnota) = 0; % odstraneni vseh ostatnich ploch
7   L(L == hodnota) = 1; % na misto indexu nejvetsi plochy zapise 1
```

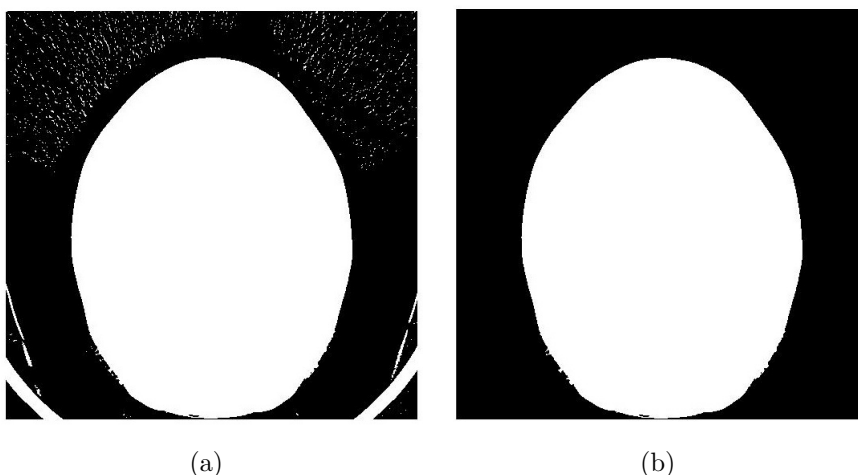


(a)



(b)

Obr. 2.2: Histogramy obrazů: originální obraz (a), naprahovaný obraz (b)



Obr. 2.3: Vytvoření masky: hrubá binární maska (a), Binární maska (b)

Posledním krokem je vynásobení původního snímku s takto vytvořenou maskou, čímž dojde k odstranění jak patientského stolu, tak zbytků podložky (Obr.: 2.4(a)).

Pro porovnání je opět zobrazen histogram (Obr. 2.4(b)), na kterém je vidět odstranění vrcholu kolem hodnoty jasu 0.5, která reprezentuje patientský stůl. Tento postup je v cyklu opakován postupně pro všechny načtené řezy. Pro urychlení výpočtu bylo použito paralelního zpracování.

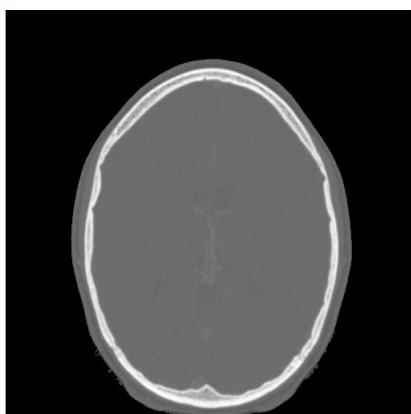
2.2.2 Schodový artefakt

Dalším problémem znemožňujícím korektní registraci obrazových dat je tzv. schodový artefakt. Tento jev vzniká při akvizici dat za použití prototypu hybridní iterační (iDose) rekonstrukční metody. K jeho odstranění je použito metody využívající fázové korelace a následné pomocí metody nejstrmějšího sestupu. Zde je použita funkce *sequen_distort_correct.m* a její pomocné funkce. Výsledek algoritmu je patrný z (Obr. 2.5) [15].

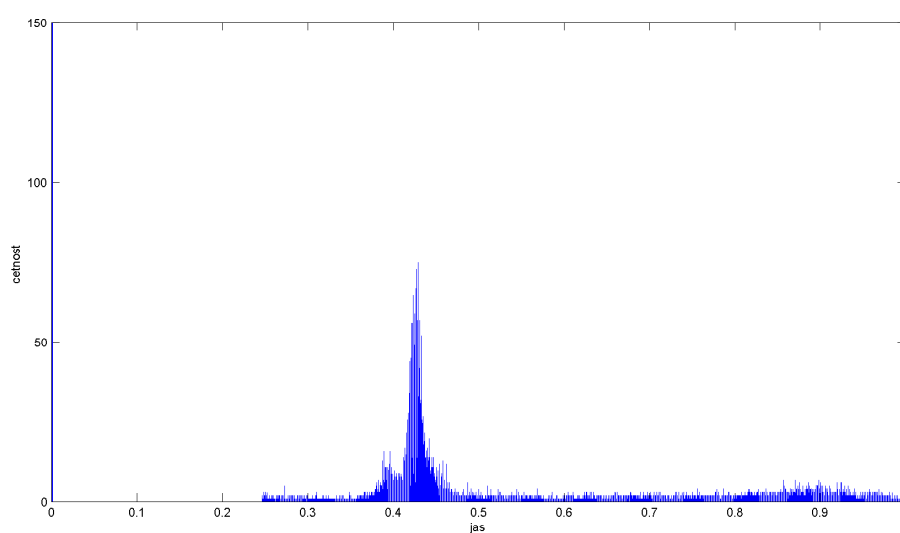
2.3 Geometrická transformace a interpolace obrazu

Transformace obrazů je programem prováděna maticovým násobením transformační matice s původními souřadnicemi. Pro tento účel je použita funkce *transformace.m*.

```
1 function[im_tr] = transformace(im,v,data,interptype)
```

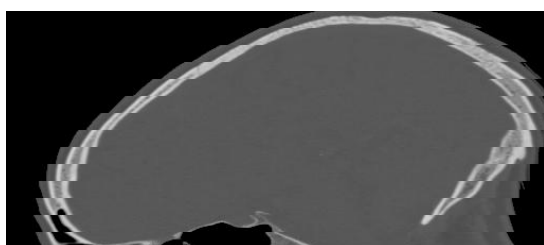


(a)

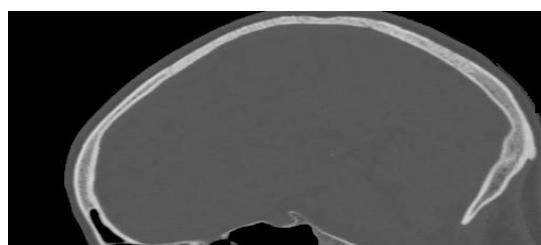


(b)

Obr. 2.4: Výsledný obraz s odstraněným patientským stolem a podložkou (a) a jeho histogram (b)



(a)



(b)

Obr. 2.5: Odstranění schodového artefaktu: sagitální pohled před odstraněním (a), sagitální pohled po odstranění (b)

Jejími vstupy jsou obrazová trojrozměrná matice, velikost této matice, transformační parametry a typ interpolační metody. Výstupem je transformovaná 3D obrazová matice. Program postupuje tak, že nejprve na základě transformačních parametrů vytvoří transformační matici dle vztahů zmíněných v kapitole (Kap. 1.1). Programově jsou transformační parametry zapsány do jednotlivých proměnných a následně přepsány do odpovídajících matic. Pro příklad je uveden zdrojový kód pro matici rotace.

```

1 %rotace ve stupních
2 thx = data(4); %rotace v ose x
3 thy = data(5); %rotace v ose y
4 thz = data(6); %rotace v ose z
5 % matice pro rotaci obrazu v osach x, y, z (rotation transformation)
6 Bz = [cos(thz) -sin(thz) 0; sin(thz) cos(thz) 0; 0 0 1];
7 By = [cos(thy) 0 sin(thy); 0 1 0; -sin(thy) 0 cos(thy)];
8 Bx = [1 0 0; 0 cos(thx) -sin(thx); 0 sin(thx) cos(thx)];
9 B = Bz*By*Bx;

```

Algoritmus poté postupně paralelně v cyklu vypočítá nové souřadnice transformovaného obrazu. Díky lineárnímu indexování souřadnic je možné použití pouze jednoho cyklu pro výpočet všech tří transformovaných souřadnic. Je proto možné zde s velkou výhodou použít paralelní zpracování dat a značně urychlit celkový výpočet.

```

1 % vypocetni cast
2 % vypocet transformovanych souradnic
3 % vytvoreni nulovych matic souradnic
4 x_tr = zeros(v(1),v(2),v(3));
5 y_tr = x_tr;
6 z_tr = x_tr;
7 % vytvoreni souradnicoveho systemu matic pred transformaci
8 [x,y,z] = meshgrid(1:v(1),1:v(2),1:v(3));
9 parfor i = 1:v(1)*v(2)*v(3) % cyklus transformace souradnic
10
11     r_tr = A*[x(i);y(i);z(i);1]; % vypocet afinni transformace souradnic
12
13     x_tr(i) = r_tr(1); % transformovane x
14     y_tr(i) = r_tr(2); % transformovane y
15     z_tr(i) = r_tr(3); % transformovane z
16 end;

```

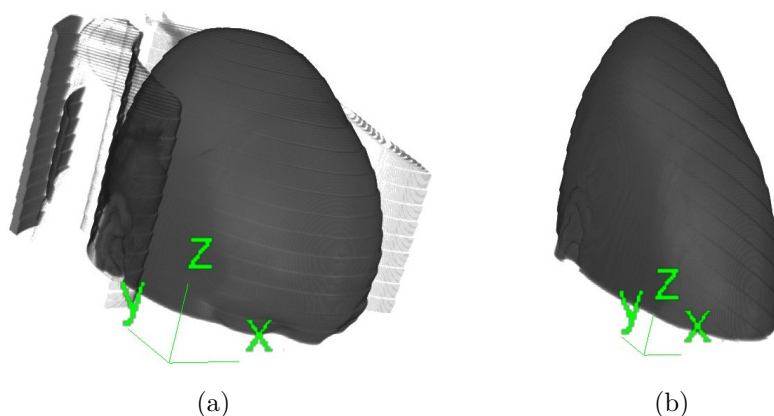
Takto získané matice souřadnic jsou použity interpolační funkcí *interp3*, která podle zvolené metody provede interpolaci obrazových dat.

```

1 % interpolace souradnic
2 im_tr = interp3(im,x_tr,y_tr,z_tr,interptype); % provedeni interpolace
3 im_tr(isnan(im_tr)) = 0; % nahrazeni neznamych hodnot nulou

```

Během transformace dochází k tomu, že se v obraze objevují neznámé nečíselné hodnoty (hodnoty mimo oblast původního obrazu). Pro další výpočty je proto nutné tyto hodnoty nahradit nulou. Ukázkou provedené transformace a odstranění rušivých elementů lze vidět na obrázku (Obr. 2.6).

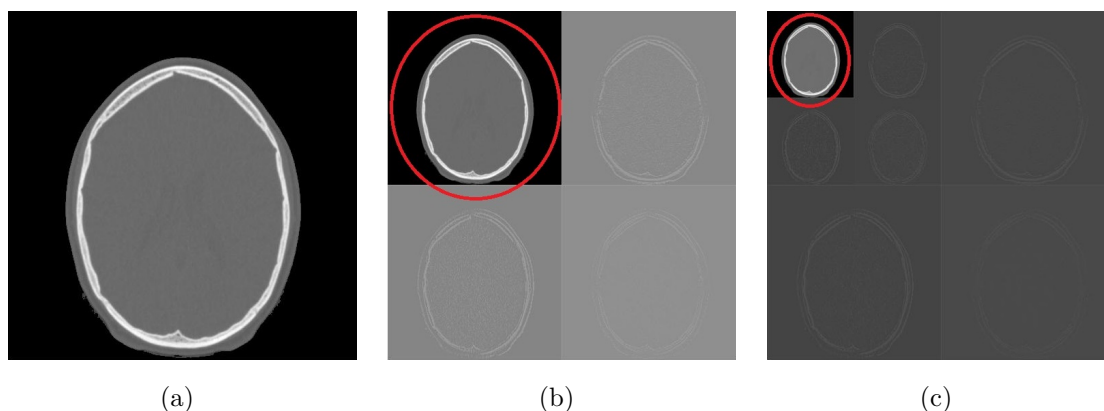


Obr. 2.6: Ukázka provedené transformace: originální scéna (a), transformovaná scéna (b), transformační parametry: škálování $x = 0.7$, $z = 1.2$, zkosení $zx = 0.2$

2.4 Podvzorkování obrazových dat

Vzhledem k velké výpočetní a časové náročnosti programu bylo nutné hledat různé metody, jak výpočet urychlit a snížit jeho hardwarové nároky. K tomuto účelu bylo použito paralelního zpracování dat, čímž došlo ke značné časové úspoře. Avšak nevýhodou paralelního zpracování je opět časová i výkonnostní závislost na hardwarovém vybavení (počet a výkon procesorů, RAM paměť). Z tohoto důvodu byla využita další metoda, která sníží jak dobu výpočtu, tak i zatížení hardwaru. Použitá metoda využívá postupné dekompozice obrazových dat pomocí 2D diskretní vlnkové transformace s Haarovou vlnkou. Zachována je pouze aproximovaná složka obrazu, matice koeficientů reprezentující detaily obrazu nejsou použity. Z vlastností vlnkové transformace vyplývá, že s každou úrovní dekompozice získáme dvakrát podvzorkovaný vstupní obraz. Tím je docíleno snížení objemu dat, avšak oproti paralelnímu zpracování má tato metoda nevýhodu ve zhoršení přesnosti registrace kvůli ztrátě detailů. Algoritmus postupně provádí podvzorkování jednotlivých řezů jak nativního, tak kontrastního obrazu. Původním záměrem bylo využití 3D vlnkové transformace, avšak v důsledku anizotropie voxelů (nižšího rozlišení obrazů v ose z) už dvojnásobným podvzorkováním ve směru osy z byla ztráta dat příliš velká, a nebylo možno provést korektní registraci. Nyní je využito čtyřnásobné podvzorkování ve směrech osy x a y se zachováním funkčnosti registrace obrazových dat. Naznačený postup výběru podvzorkovaných dat je patrný na (Obr. 2.7).

V praxi jsou před podvzorkováním oba obrazy upraveny odstraněním nežádoucích struktur a schodového artefaktu. Poté jsou takto upravené obrazy uloženy do pomocných proměnných a originály následně podvzorkovány. Po celou dobu průběhu CRS optimalizačního algoritmu program pracuje s podvzorkovanými obrazy. Jakmile jsou nalezeny transformační parametry odpovídající minimalizaci kritériální



Obr. 2.7: Ukázka použité vlnkové dekompozice: originální obraz (a), první dekompozice (b), druhá dekompozice (c)

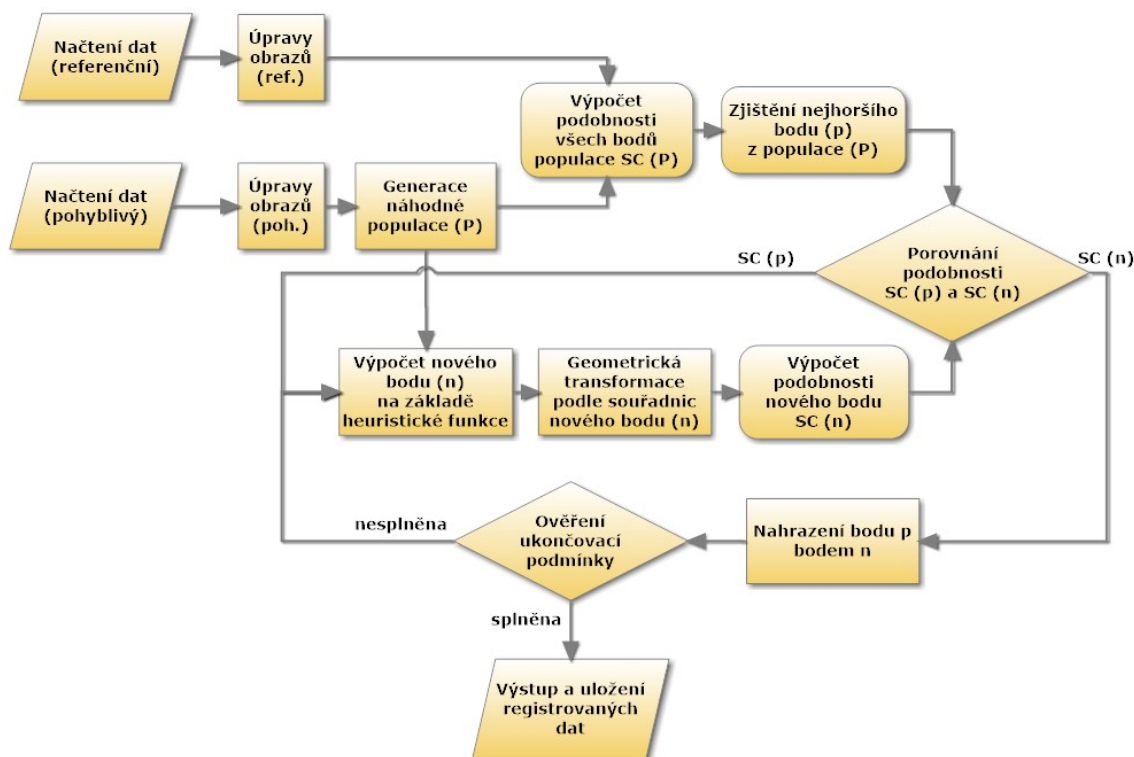
funkce na podvzorkovaných obrazech, jsou tyto parametry použity úplnou geometrickou transformací originálního (nepodvzorkovaného) pohyblivého obrazu. Poté je výpočet ukončen. Při aktivním paralelním zpracování trvá transformace obrazu (obraz: 512x512x224, PC: 12x2,54 GHz) bez použití podvzorkování přibližně 2 minuty, za použití čtyřnásobného podvzorkování pouhých 8 sekund. Výpočetní čas nutný pro podvzorkování obou obrazů je také přibližně 8 sekund. Další možnou volbou je pyramidový přístup. V tomto případě jsou transformační parametry nalezeny nejprve na čtyřnásobně podvzorkovaných obrazech. Poté jsou jako vstupní obrazy použita dvojnásobně podvzorkovaná data a hranice prohledávaného prostoru jsou určena transformačními parametry nalezené optimalizací čtyřnásobně podvzorkovaných obrazů. Opět je provedena optimalizace a jsou nalezeny nové transformační parametry. Obdobně je tento proces aplikován na originální obrazy s prostorem definovaným transformačními parametry po optimalizaci dvojnásobně podvzorkovaných dat. Po ukončení posledního optimalizačního kroku jsou nalezeny výsledné transformační parametry. Tento postup je pomalejší než předchozí, avšak dosahuje lepších výsledků slícování.

2.5 Algoritmus CRS

Optimalizačním algoritmem programu je kontrolované náhodné prohledávání. Zjednodušenou funkci programu je možno popsat blokovým diagramem (Obr. 2.8)

Jednotlivé bloky programu mohou být popsány takto:

- **načtení referenčních a pohybových dat** – načtení objemových obrazových dat nativních i kontrastních snímků (viz (Kap. 2.1))
- **úpravy obrazových dat** – do těchto bloků spadá dříve zmíněné odstranění



Obr. 2.8: Blokové schéma programu

nežádoucích struktur a dále také podvzorkování obrazů za účelem zkrácení doby výpočtu

- **generace náhodné populace (P)** – zde je vygenerována náhodná populace transformačních parametrů, její velikost (počet bodů) a rozptyl závisí na nastaveném prohledávaném prostoru (D) a zvolené velikosti populace (odvislá od dimenze prohledávaného prostoru). K jejímu vygenerování je použito funkce *rand*. Protože tato funkce generuje hodnoty od 0 do 1, je pro generaci v celém prostoru nutné použít pomocný výpočet (Rov. 2.1)

$$\mathbf{p}_i = \min(\mathbf{D}_i) + (\max(\mathbf{D}_i) - \min(\mathbf{D}_i)) \cdot \text{rand}(0; 1), \quad (2.1)$$

kde \mathbf{p}_i je i -tá souřadnice bodu populace, $\min(\mathbf{D}_i)$ je spodní hranice prohledávaného prostoru odpovídající i -té souřadnici, $\max(\mathbf{D}_i)$ je horní hranice a $\text{rand}(0; 1)$ je náhodně vygenerované číslo z intervalu $[0;1]$.

- **výpočet podobnosti SC (P)** – zde jsou vypočtena zvolená podobnostní kritéria každého bodu populace (P), přičemž tento krok v programu provádí funkce *similarity.m*

1 `function [SC] = similarity(v,Stype,im1,im2)`

Její vstupní parametry jsou velikost obrazových dat, typ použitého kritéria (součet čtvercový vzdáleností, Eukleidovská vzdálenost, kosinové kritérium

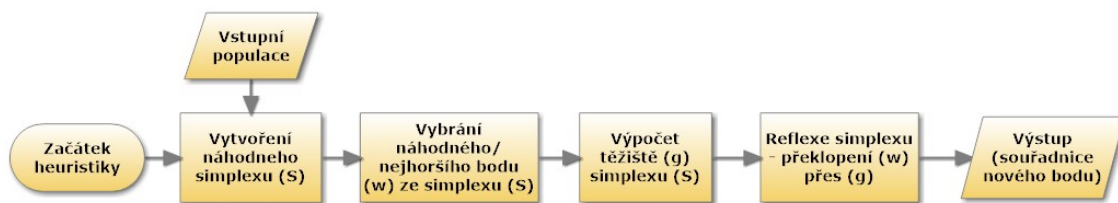
nebo korelační kritérium) a 3D matice referenčních a pohybových obrazových dat. Výstupem je pak vypočtená podobnost.

- **zjištění nejhoršího bodu (\mathbf{p})** – je nalezen bod (\mathbf{p}) z populace (P), který má extrémní (nejhorší) hodnotu podobnostního kritéria
- **heuristická funkce** – blok s některou možnou heuristickou funkcí (např. reflexe simplexu), zde je nalezena nová hodnota transformačních parametrů (\mathbf{n}). Více v (Kap. 2.5.1)
- **geometrická transformace (\mathbf{n})** – blok provádějící 3D afinní geometrickou transformaci prostorových souřadnic (viz (Kap. 2.3))
- **výpočet SC (\mathbf{n})** – výpočet podobnosti transformovaných obrazových dat, použita opět funkce *similarity.m*
- **porovnání podobnosti** – porovnává hodnotu podobnostního kritéria extrémního bodu (\mathbf{p}) z populace (P) s hodnotou podobnostního kritéria nového bodu (\mathbf{n}) (pokud je hodnota nového bodu lepší, dojde k přesunu na blok nahrazení; pokud je horší, dojde k přesunu na blok heuristické funkce)
- **nahrazení bodu (\mathbf{p}) bodem (\mathbf{n})** – nahrazení nejhoršího bodu (\mathbf{p}) z populace (P) novým bodem (\mathbf{n})
- **podmínka zastavení výpočtu** – rozhodovací pravidlo ukončení registrace obrazů (pokud je splněna, je algoritmus zastaven; pokud není splněna, návrat do bloku heuristické funkce). Více v (Kap. 2.5.2).
- **výstup a uložení registrovaných dat** – výstupní blok slícovaných dat (zde jsou data uložena funkcí *dicom_ulozeni.m* a prezentovány výsledky registrace funkcí *zobr.m* pro MatLab verze R2012a a novější nebo *zobr2.m* pro starší verze)

2.5.1 Heuristická funkce

Heuristická funkce je jádrem algoritmu CRS. Variant této funkce je velké množství lišící se provedením, složitostí a použitelností pro různé aplikace. Pro každou aplikaci mohou být některé heuristiky vhodnější z hlediska rychlosti konvergence, spolehlivosti nalezení globálního extrému a rychlosti výpočtu než jiné. Všechny heuristické funkce však mají za úkol vytvořit nový bod (n) ležící v prohledávaném prostoru (D) za pomoci bodů (p) z generované populace (P), který bude svou funkční hodnotou blíže k hledanému extrému. Pro testování programu byly vybrány některé základní heuristiky.

- a) reflexe simplexu – Základní heuristická funkce, která má několik modifikací. Nejběžnější je použití reflexe nejhoršího bodu z náhodného simplexu. Může být použita také znáhodněná reflexe simplexu, kde je reflektován náhodný bod z náhodného simplexu (Obr. 2.9).

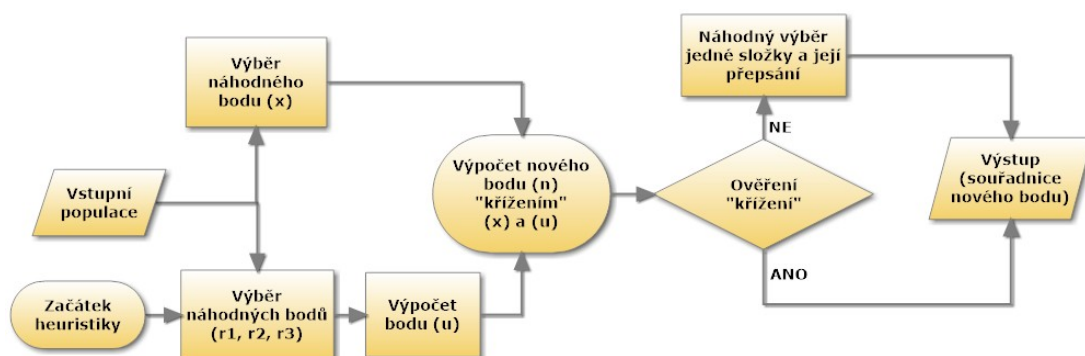


Obr. 2.9: Schéma použité reflexe simplexu

- b) diferenciální evoluce – Heuristika použitelná ve dvou variantách využívající pravděpodobnosti přepisu souřadnic bodů. Tato metoda vychází z náhodně vybraných bodů z populace (P). Nejprve je z celé populace (P) vybrán náhodný bod (\mathbf{x}), následně jsou vybrány 3 další náhodné body (\mathbf{r}). Na základě těchto bodů je vypočten pomocný bod (\mathbf{u}) podle rovnice (Rov. 2.2)

$$\mathbf{u} = \mathbf{r}_1 + F(\mathbf{r}_2 - \mathbf{r}_3), \quad (2.2)$$

kde \mathbf{u} je vypočtený pomocný bod, \mathbf{r} jsou náhodně vybrané body a F je vstupní parametr heuristické funkce. Následně je vypočten nový bod \mathbf{n} tzv. “křížením” vektorů \mathbf{u} a \mathbf{x} . Výpočet probíhá tak, že jsou postupně přepisovány jednotlivé složky vektoru \mathbf{x} složkami vektoru \mathbf{u} s určitou pravděpodobností. Pravděpodobnost přepisu je dána náhodně vygenerovanou veličinou U (z intervalu $[0;1]$) a vstupní proměnnou C (také z intervalu $[0;1]$). Pokud nastane ($U < C$), pak bude i -tá složka vektoru \mathbf{x} přepsána. Pokud bude ($U \geq C$), zůstane složka ponechána. Nakonec dojde k ověření zda došlo k alespoň jednomu přepisu souřadnice. Pokud ne, je přepsána jedna náhodně vybraná složka. Popsaný postup je patrný z (Obr. 2.10).



Obr. 2.10: Schéma použité diferenciální evoluce

Druhá varianta diferenciální evoluce je obdobná, avšak do výpočtu pomocného bodu \mathbf{u} ještě započítává nejlepší bod z celé populace (P). Rovnice (Rov. 2.2) je pak upravena na rovnici (Rov. 2.3):

$$\mathbf{u} = \mathbf{x}_{best} + F(\mathbf{r}_1 + \mathbf{r}_2 - \mathbf{r}_3 - \mathbf{r}_4), \quad (2.3)$$

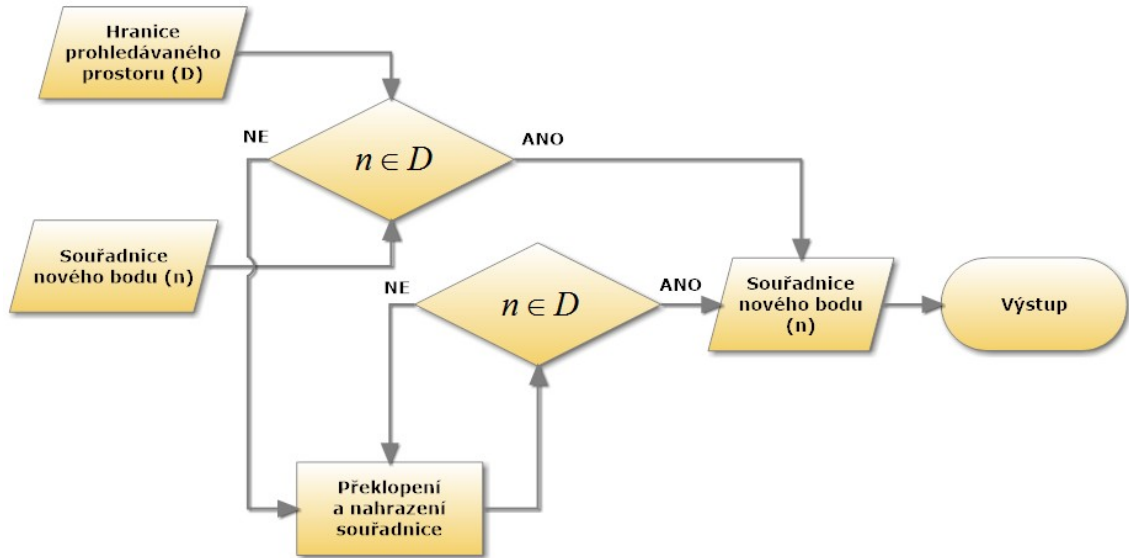
kde \mathbf{x}_{best} je nejlepší bod z populace (P), \mathbf{r} jsou opět náhodně vybrané body a F je vstupní parametr. Ostatní postup zůstává stejný jako v první variantě.

- c) *perturbace* – Doplněk všech heuristických metod, který zajišťuje jejich účinnější funkci. Jeho úkolem je zajistit, aby nově vygenerovaný bod náležel do prohledávaného prostoru. Zařazení tohoto bloku do programu značně urychlí výpočet, protože omezí použití transformační funkce a výpočtu podobnostního kritéria v případě, kdy nový bod leží mimo prohledávaný prostor (a má tedy téměř 100% pravděpodobnost, že jeho hodnota bude horší než hodnota nejhoršího bodu z populace (P)). Pokud je některá ze souřadnic nového bodu přesahuje hranice prohledávaného prostoru, bude přepočítána podle (Rov. 2.4) nebo (Rov. 2.5). Schématický popis jeho funkce je na (Obr. 2.11).

$$n_i > \max(D_i) \Rightarrow n_i = 2\max(D_i) - n_i \quad (2.4)$$

$$n_i < \min(D_i) \Rightarrow n_i = 2\min(D_i) - n_i, \quad (2.5)$$

kde n_i je i -tá souřadnice nového bodu a D_i jsou souřadnice definující prohledávaný prostor.



Obr. 2.11: Schéma použité perturbace

2.5.2 Podmínka zastavení výpočtu

Z důvodu nemožnosti odhadnout počet nutných iteračních kroků k dosažení “ideálního” slícování obrazových dat je použito automatické zastavení výpočtu při splnění

určité podmínky. Možností pro definici této podmínky je mnoho. Mezi metody, které jsou prakticky použitelné patří zastavení výpočtu při dosažení určité hodnoty podobnostního kritéria. Tato metoda je však nevhodná z důvodu neznalosti hodnoty, které má být dosaženo. Navíc samotná hodnota podobnostního kritéria nevypovídá o tom, zda došlo ke slícování obrazů. Zejména u nenormovaných podobnostních kritérií, jako je prostá eukleidovská vzdálenost, je její použití nevhodné. Další možností je zastavení výpočtu, pokud se podobnostní kritérium ve dvou nebo více po sobě jdoucích iteracích nemění, případně pokud je změna podobnosti menší než definovaná přesnost. Dále je možno využít výpočtu vzájemné vzdálenosti bodů populace a zastavit výpočet, pokud je jejich vzdálenost menší než definovaná hodnota.

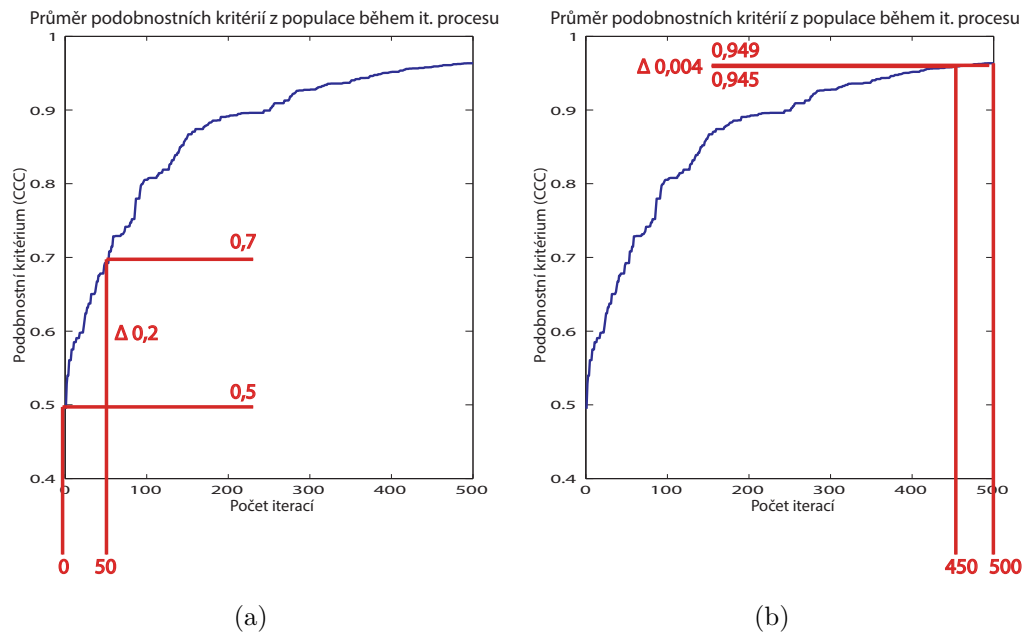
V této práci je využito podmínky, která provede zastavení výpočtu při dosažení určitého rozdílu podobnostního kritéria mezi poslední provedenou iterací a iterací před určitým počtem kroků. Tím je zajištěno, že nedojde k nechtěnému zastavení výpočtu, pokud po určitou dobu nedojde k nalezení lepší podobnosti. Podobnostní kritérium použité ve výpočtu je zjištěno z průměru podobností všech bodů populace. Podmínkou pro ukončení je tedy vzájemná vzdálenost jednotlivých bodů populace. Tento algoritmus má dvě nastavitelná kritéria, a to procentuální rozdíl vzdálenosti bodů (např. 2 % z počátečního vývoje rozložení) a velikost zpoždovacího okna (např. 50 iteračních kroků). Na obrázku (Obr. 2.12(a)) je znázorněn výpočet referenčního rozdílu při volbě okna 50 iteračních kroků. Rozdíl tedy vyšel 0,2. Během průběhu registrace je postupně počítán rozdíl mezi hodnotou na konci okna a na začátku okna. Výpočet bude zastaven, jakmile rozdíl těchto hodnot bude menší než definované procento z referenčního rozdílu. Při volbě rozdílu 2 % bude tedy výpočet zastaven, jakmile rozdíl bude menší než 0,004 (Obr. 2.12(b)).

2.6 Grafické rozhraní

Program je spouštěn přes grafické uživatelské rozhraní (GUI) z důvodu usnadnění nastavení jednotlivých parametrů a snadné prezentaci výstupu registrace. GUI se skládá z hlavního spouštěcího okna (Obr. 2.13(a)), informačního okna (Obr. 2.13(b)) a okna pro zobrazení funkce procesu (Obr. 2.13(c)). Program obsahuje i další pomocná dialogová okna, která jsou v informační technice běžně používána.

Hlavní okno obsahuje několik sekcí podle použití.

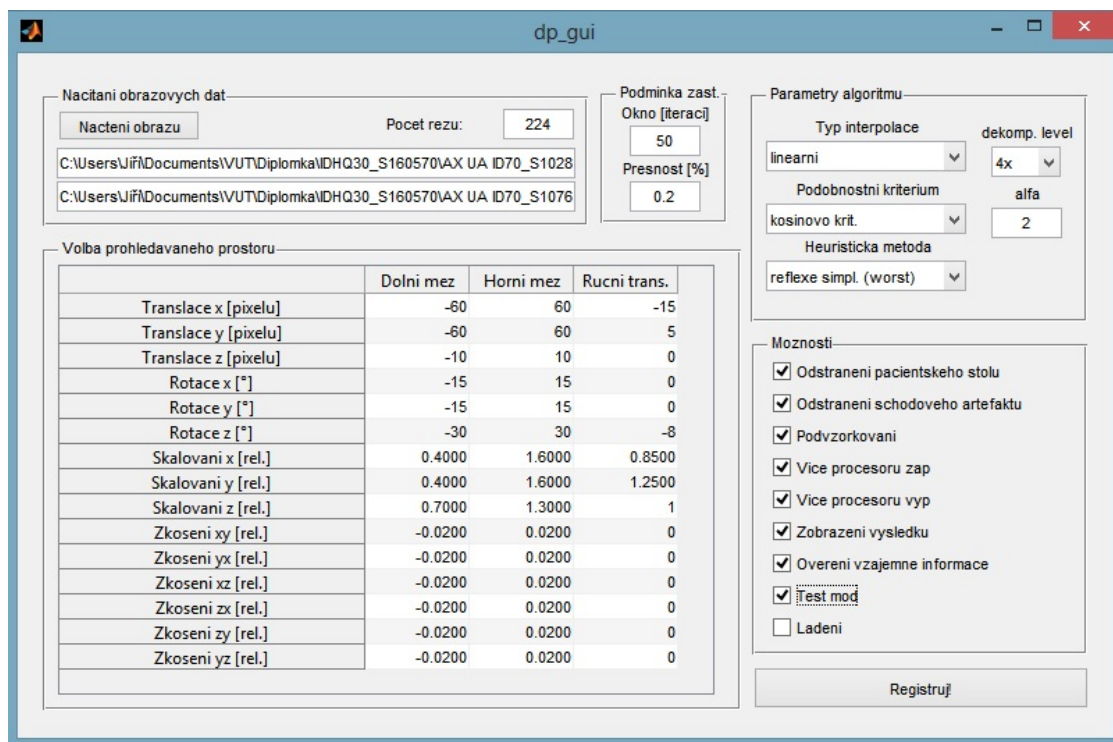
- Načítání obrazových dat – obsahuje tlačítko, pomocí kterého jsou vybrány oba obrazy. Dokud nejsou načtena obrazová data, není možné nic nastavit. Tato sekce dále obsahuje informace o cestě k nativnímu i kontrastnímu obrazu a informace o počtu řezů. Počet řezů je možné editovat, avšak některé části programu vyžadují kompletní trojrozměrná data pro svou funkci.



Obr. 2.12: Ukázka výpočtu zastavovací podmínky: výpočet referenčního rozdílu (a), výpočet rozdílu během výpočtu (b)

- Volba prohledávaného prostoru – obsahuje tabulku, ve které je možné nastavit hranice prohledávaného prostoru a v případě umělého zkreslení také transformační parametry.
- Podmínka zastavení výpočtu – umožňuje nastavit parametry automatického ukončení výpočtu.
- Parametry algoritmu – zde je možno volit různé interpolační metody, podobnostní kritéria, míru podvzorkování a heuristické metody. Pro určité heuristiky je také možná volba jejich vstupních parametrů.
- Možnosti – umožňuje nastavit funkci programu a vypínat/zapínat jednotlivé bloky. Políčka *Více procesorů zap* a *Více procesorů vyp* určují, zda bude před spuštěním registrace aktivován matlabpool (připojení procesorů pro paralelní zpracování) a naopak, zda má být po ukončení deaktivován. *Test mód* dovolí použití transformace pohybového obrazu pomocí ručně zadaných parametrů. *Ladění* slouží ke spuštění všech povolených bloků programu bez optimalizačního algoritmu (například umožní pouze odstranit patientský stůl). Ostatní políčka mají zřejmou funkci dle popisu.
- Registruj – spustí registraci podle nastavených parametrů.

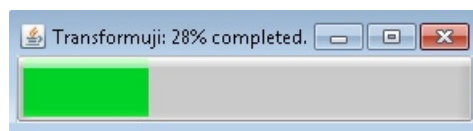
Informační okno zobrazuje informace o právě vykonávané operaci během funkce programu. Procesové okno ukazuje postup jednotlivých probíhajících procesů. Toto okno využívá *java* skript, aby bylo možné zobrazit průběh při paralelním zpracování.



(a)



(b)



(c)

Obr. 2.13: Grafické uživatelské rozhraní: hlavní okno (a), informační okno (b), okno průběhu procesů (c)

3 VÝSLEDKY

Výsledným programem byla testována registrace několika reálných i uměle zkreslených dat s různým nastavením. Byla testována různá podobnostní kritéria a heuristické metody, přičemž byly měřeny časy registrace dat a míra slícování výsledných obrazů. V této části jsou prezentovány některé dosažené výsledky a ukázky registrovaných obrazů.

3.1 Uměle zkreslená data

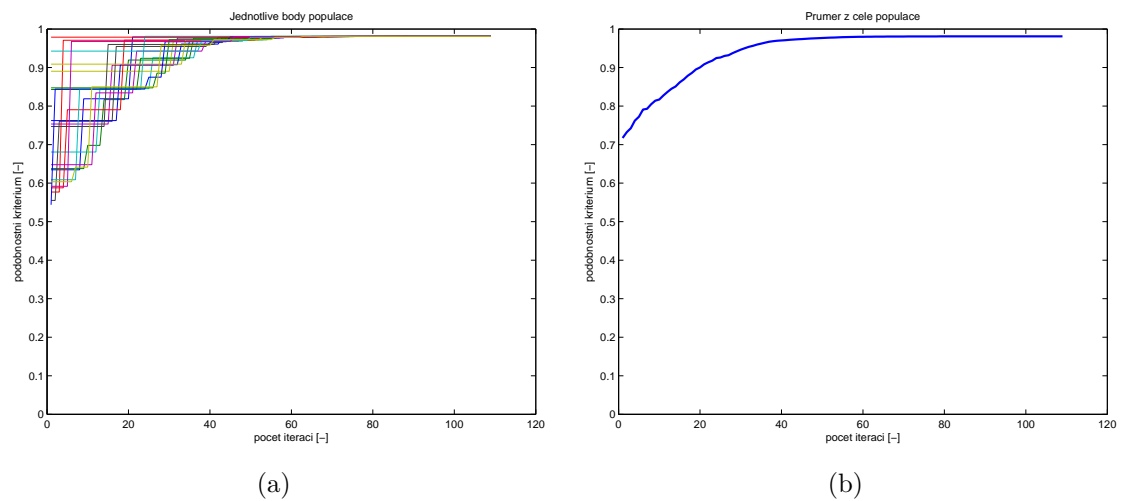
Pro ověření funkce algoritmu jako celku i jeho jednotlivých součástí bylo použito umělého zkreslení dat. Zkreslení bylo provedeno, tak, že před registrací obrazu byla na pohybový (kontrastní) obraz aplikována geometrická transformace se známými transformačními parametry. Jako referenční obraz byl použit obraz nativní. Při registraci pak byly testovány rozdíly v časech registrace a kvalitě slícování výsledného obrazu při stejných vstupních obrazech, ale rozdílných metod.

3.1.1 Translace obrazu

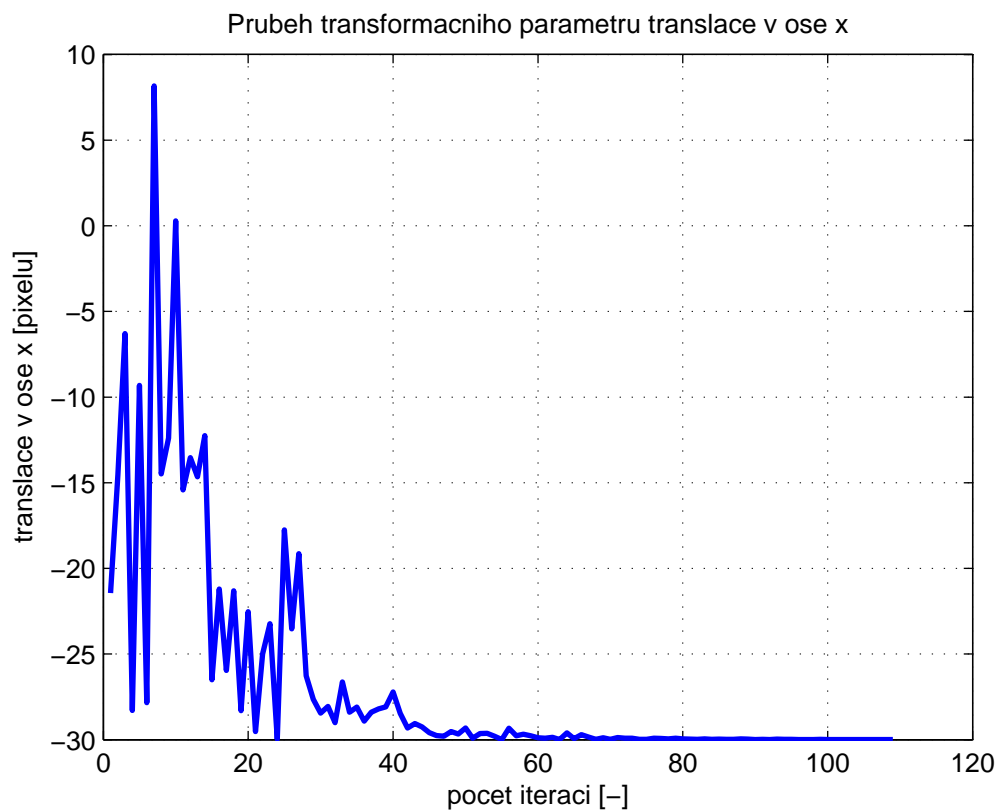
Bylo provedeno zkreslení obrazu pomocí translace o 30 pixelů doleva v ose x. Registrace probíhala s tímto nastavením: prohledávaný prostor ± 35 pixelů v ose x, okno ukončovací podmínky 50 iterací, přesnost 0,5 %, korelační podobnostní kritérium, lineární interpolace, 4x podvzorkování, využito paralelní zpracování, rozměry obrazových dat 512x512x224, nativní referenční obraz, kontrastní pohybový obraz, heuristická metoda reflexe simplexu ($\alpha = 2$). Registrace byla ukončena po 109 iteračních krocích s celkovou dobou trvání 46,5 minuty a konečná hodnota podobnostního kritéria byla 0,9810. Vzájemná informace obrazů byla před provedením registrace 0,6014 a po registraci 1,2510. Průběh podobnostního kritéria je patrný z (Obr. 3.1), kde (Obr. (a)) zobrazuje vývoj podobnosti pro jednotlivé body z vygenerované populace (jejich postupné nahrazování) a (Obr. (b)) zobrazuje průměrné podobnostní kritérium vypočtené ze všech bodů v populaci v daném iteračním kroku. Průběh transformačního parametru translace v ose x z (Obr. 3.2) a porovnání výsledných obrazů je vidět na (Obr. 3.3).

3.1.2 Rotace obrazu

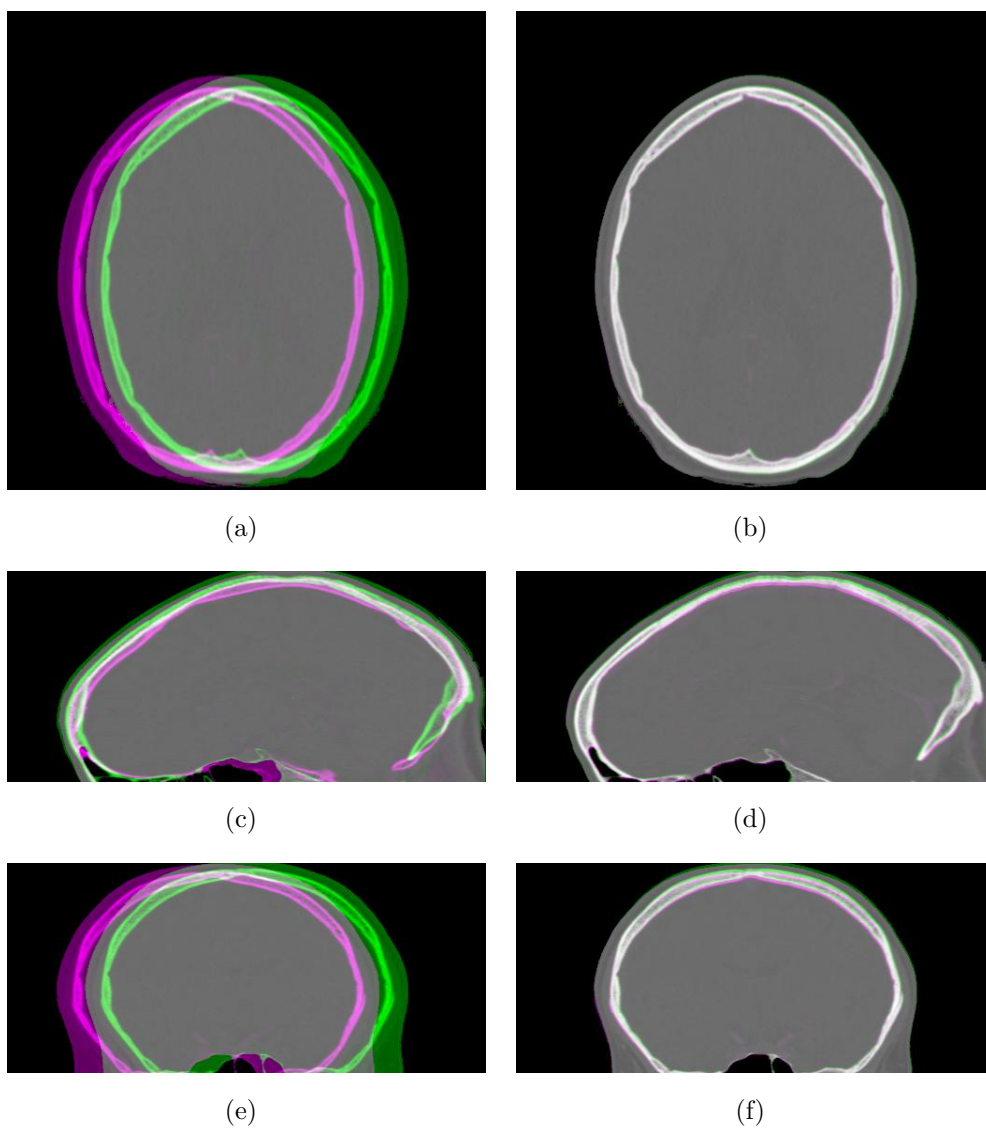
Obdobně bylo provedeno zkreslení obrazu pomocí rotace o 10° v ose rovině xy. Registrace probíhala s tímto nastavením: prohledávaný prostor ± 30 pixelů v ose x



Obr. 3.1: Průběh podobnostního kritéria (translace x): jednotlivé body populace (a), průměrná podobnost (b)

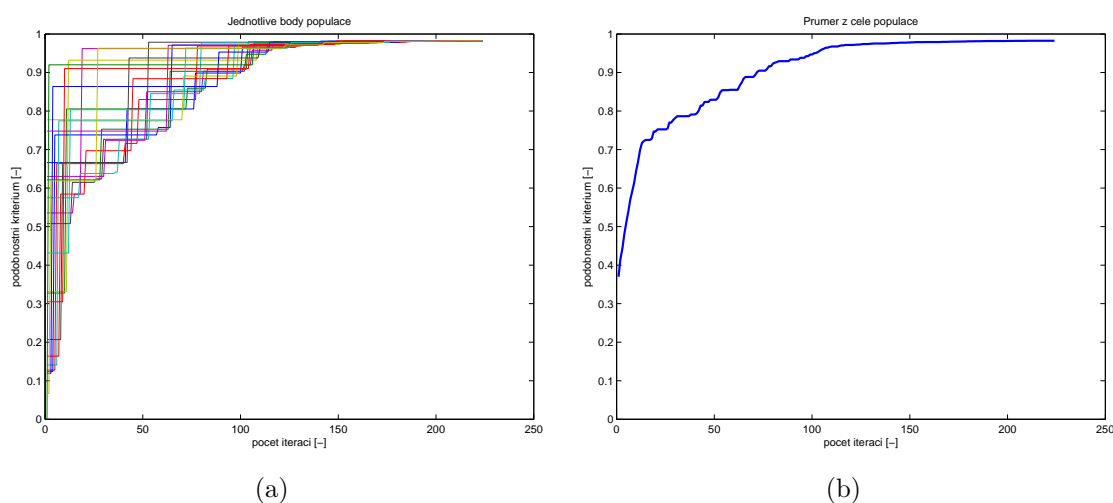


Obr. 3.2: Průběh transformačního parametru translace v ose x



Obr. 3.3: Ukázka registrace (translace v ose x): transverzální pohled před registrací (a), transverzální pohled po registraci (b), sagitální pohled před reg. (c), sagitální pohled po reg. (d), frontální pohled před reg. (e), frontální pohled po reg. (f) (zelená – referenční obraz, fialová – pohybový obraz, šedá – slícované obrazy)

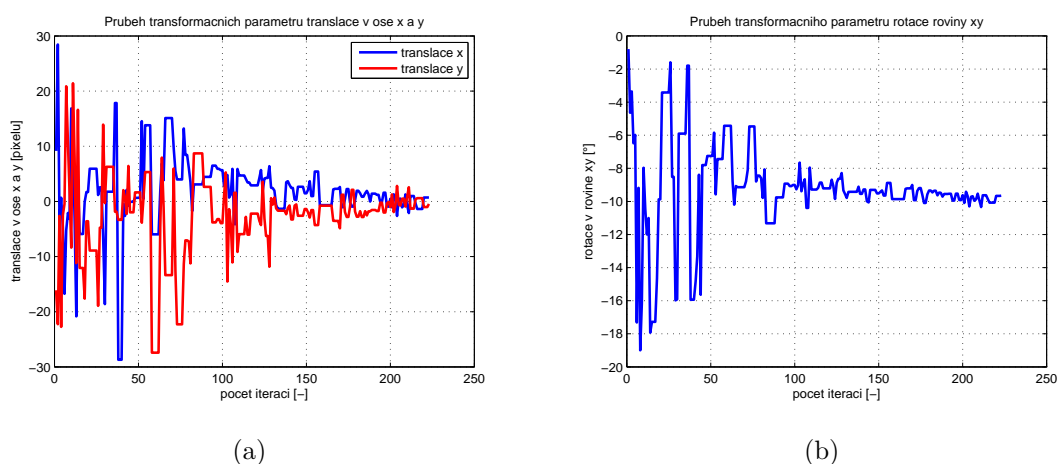
a y , $\pm 20^\circ$ v rovině xy , okno ukončovací podmínky 50 iterací, přesnost 0,5 %, korelační podobnostní kritérium, lineární interpolace, 4x podvzorkování, využito paralelní zpracování, rozměry obrazových dat 512x512x224, nativní referenční obraz, kontrastní pohybový obraz, heuristická metoda reflexe simplexu ($\alpha = 2$). Registrace byla ukončena po 224 iteračních krocích s celkovou dobou trvání 63,2 minuty a konečná hodnota podobnostního kritéria byla 0,9824. Vzájemná informace obrazů byla před provedením registrace 0,3289 a po registraci 1,3013. Průběh podobnostního kritéria je patrný z (Obr. 3.4), průběh transformačních parametrů translace v osách x a y z (Obr. (a)), transformačního parametru rotace v rovině xy z (Obr. (b)) a porovnání výsledných obrazů je vidět na (Obr. 3.6).



Obr. 3.4: Průběh podobnostního kritéria (rotace xy): jednotlivé body populace (a), průměrná podobnost (b)

3.1.3 Škálování obrazu

Bylo provedeno zkreslení obrazu pomocí škálování 0,85 v ose x a 1,25 v ose y . Registrace probíhala s tímto nastavením: prohledávaný prostor $\pm 0,5 - 1,5$ v osách x a y , okno ukončovací podmínky 50 iterací, přesnost 0,5 %, korelační podobnostní kritérium, lineární interpolace, 4x podvzorkování, využito paralelní zpracování, rozměry obrazových dat 512x512x224, nativní referenční obraz, kontrastní pohybový obraz, heuristická metoda reflexe simplexu ($\alpha = 2$). Registrace byla ukončena po 276 iteračních krocích s celkovou dobou trvání 73,00 minuty a konečná hodnota podobnostního kritéria byla 0,9822. Vzájemná informace obrazů byla před provedením registrace 0,2942 a po registraci 1,2954. Průběh podobnostního kritéria je patrný z (Obr. 3.7), průběh transformačního parametrů škálování v osách x a y z (Obr. 3.8) a porovnání výsledných obrazů je vidět na (Obr. 3.9).



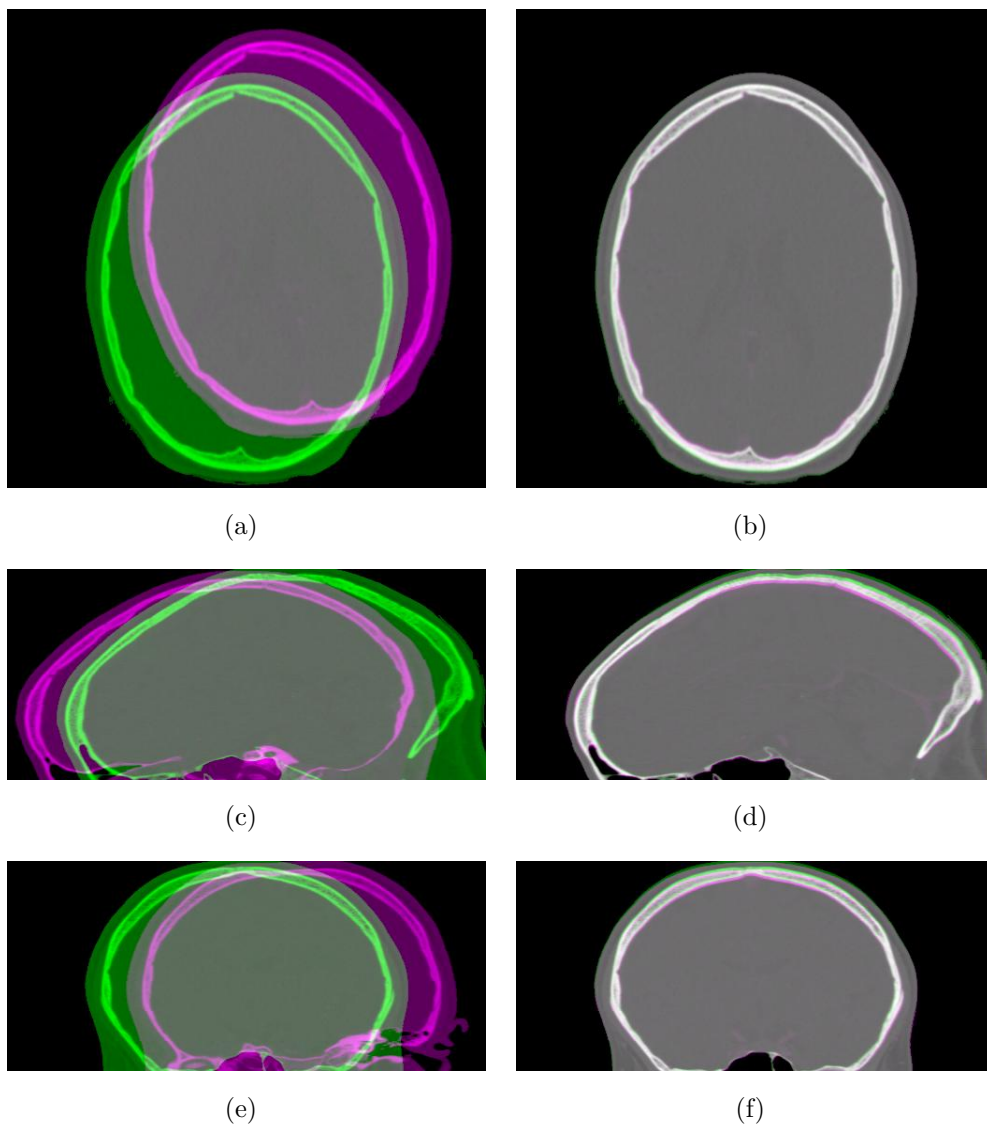
Obr. 3.5: Průběh transformačních parametrů: translace v osách x a y (a), rotace roviny xy (b)

3.1.4 Afinní transformace

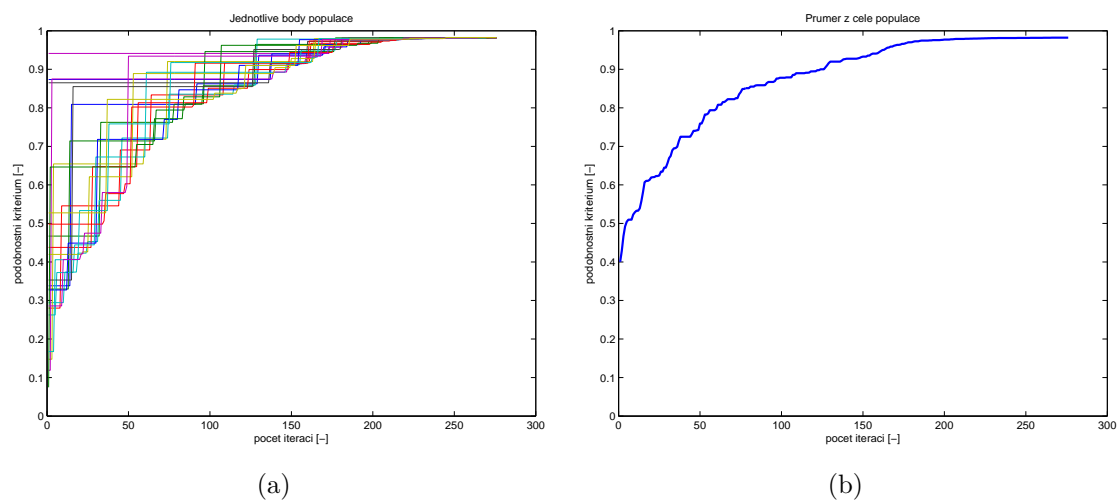
Bylo provedeno afinní zkreslení obrazu translace x 5 pixelů, translace y -5 pixelů, rotace xy -8° , škálování 0,85 v ose x a 1,25 v ose y. Registrace probíhala s tímto nastavením: prohledávaný prostor ± 30 pixelů translace v osách x a y, ± 10 pixelů v ose z, rotace $\pm 10^\circ$ v osách x a y, rotace $\pm 15^\circ$ v ose z, škálování 0,7 – 1,3 v osách x a y, škálování 0,9 – 1,1 v ose z, zkosení ve všech osách $\pm 0,001$. Okno ukončovací podmínky 50 iterací, přesnost 0,5 %, korelační podobnostní kritérium, lineární interpolace, 4x podvzorkování, využito paralelní zpracování, rozměry obrazových dat 512x512x224, nativní referenční obraz, kontrastní pohybový obraz, heuristická metoda reflexe simplexu ($\alpha = 2$). Registrace byla ukončena po 380 iteračních krocích s celkovou dobou trvání 54,62 minuty a konečná hodnota podobnostního kritéria byla 0,9179. Vzájemná informace obrazů byla před provedením registrace 0,5929 a po registraci 0,9297. Průběh transformačních parametrů (Obr. 3.10) a porovnání výsledných obrazů je vidět na (Obr. 3.11).

3.2 Reálná data

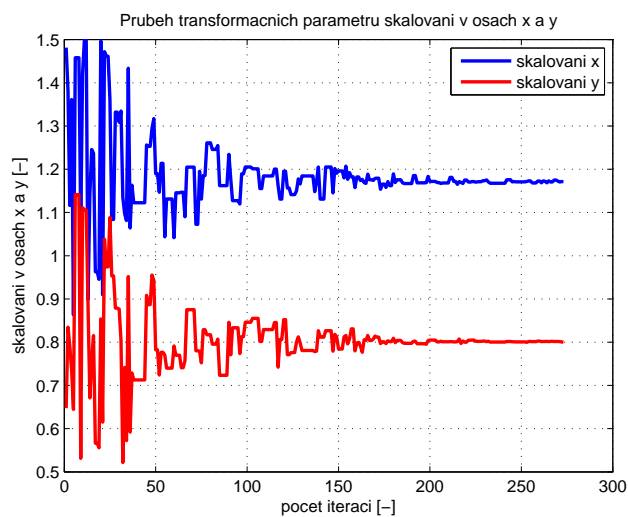
Kromě testování na uměle zkreslených datech byl také program aplikován na reálná data. Tyto obrazy nejsou zkresleny umělou transformací, ale jsou zatížena pohybovými artefakty a odlišným nastavením při akvizici dat. Ukázky dosažených výsledků lze vidět na následujících obrázcích (Obr.: 3.12).



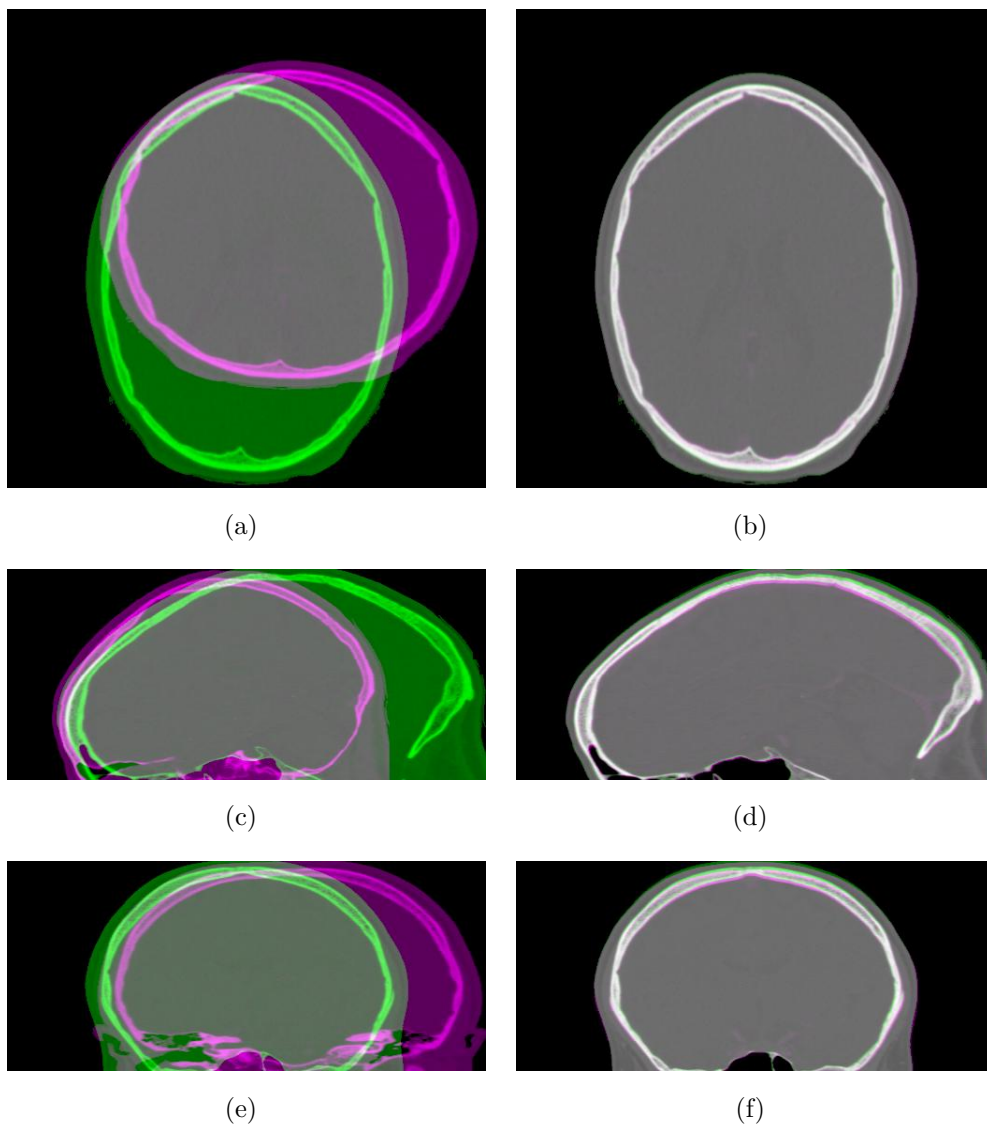
Obr. 3.6: Ukázka registrace (rotace v rovině xy): transverzální pohled před registrací (a), transverzální pohled po registraci (b), sagitální pohled před reg. (c), sagitální pohled po reg. (d), frontální pohled před reg. (e), frontální pohled po reg. (f) (zelená – referenční obraz, fialová – pohybový obraz, šedá – slícované obrazy)



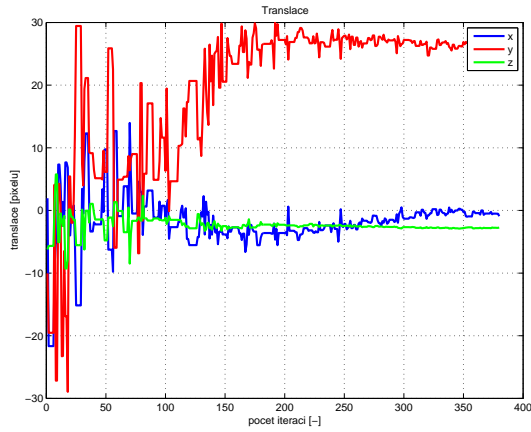
Obr. 3.7: Průběh podobnostního kritéria (škálování x a y): jednotlivé body populace (a), průměrná podobnost (b)



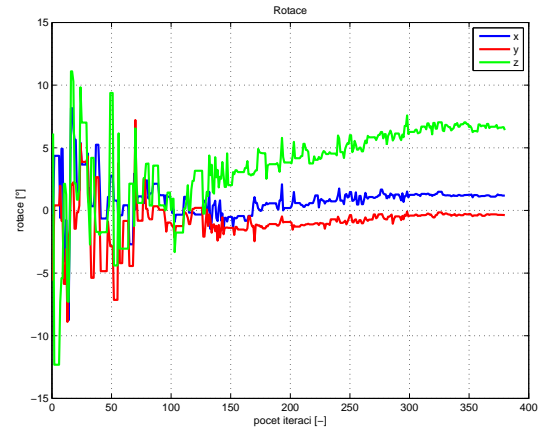
Obr. 3.8: Průběh transformačních parametrů škálování v osách x a y



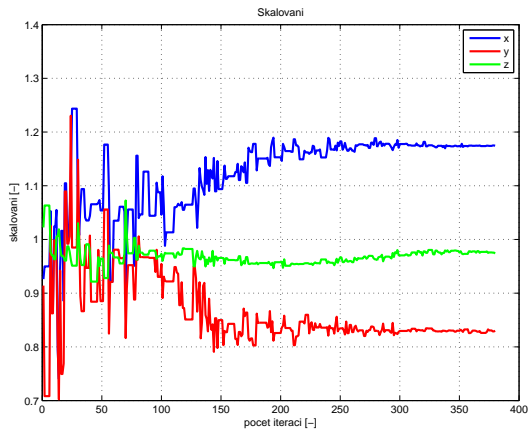
Obr. 3.9: Ukázka registrace (škálování v osách x a y): transverzální pohled před registrací (a), transverzální pohled po registraci (b), sagitální pohled před reg. (c), sagitální pohled po reg. (d), frontální pohled před reg. (e), frontální pohled po reg. (f) (zelená – referenční obraz, fialová – pohybový obraz, šedá – slícované obrazy)



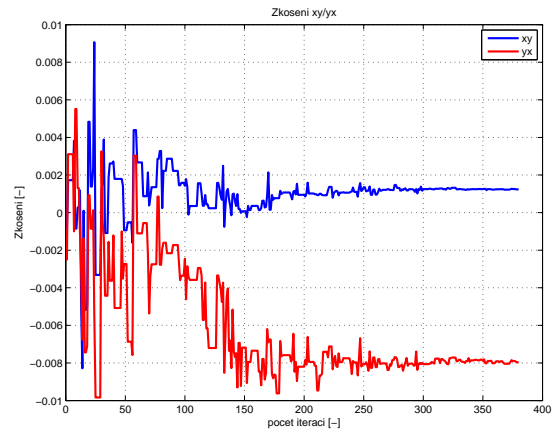
(a)



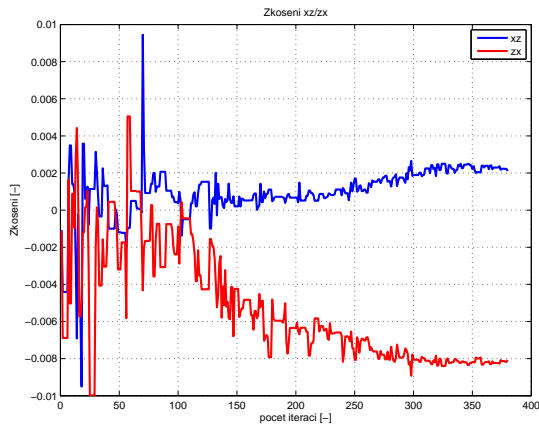
(b)



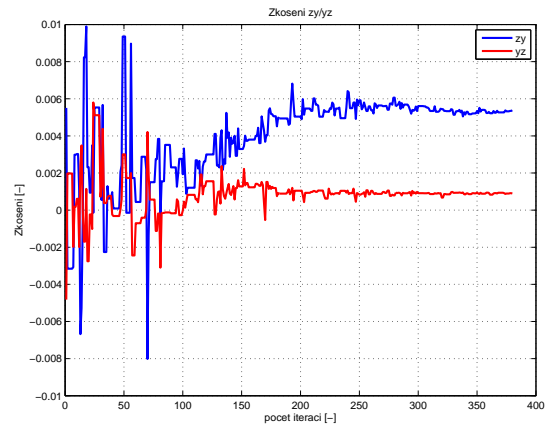
(c)



(d)

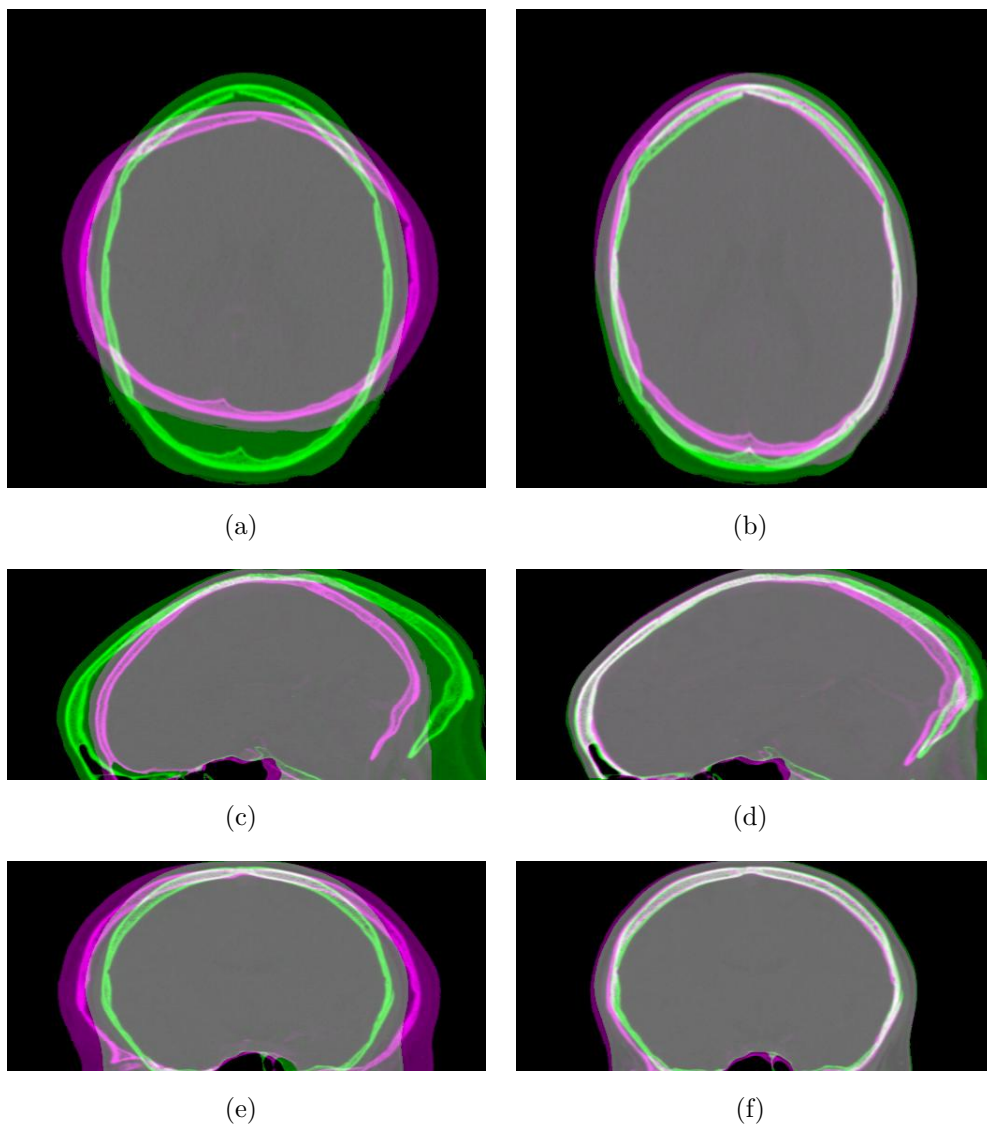


(e)

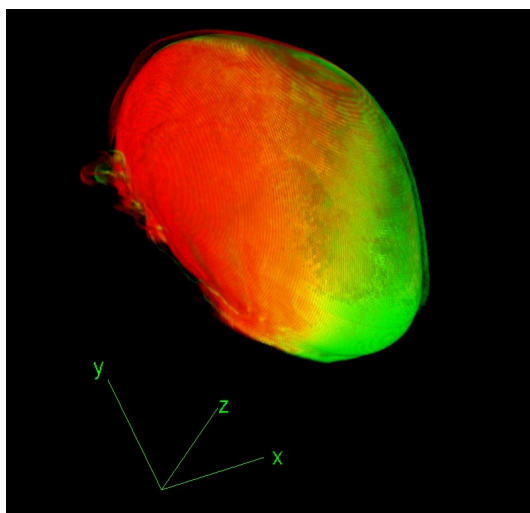


(f)

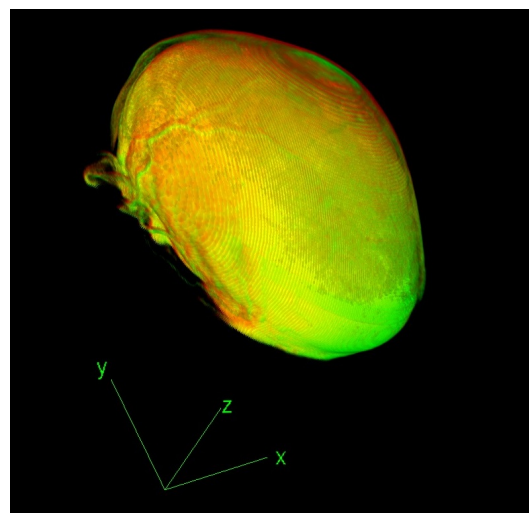
Obr. 3.10: Průběh transformačních parametrů: translace v osách x , y , z (a), rotace roviny xy , xz , yz (b), škálování v ose x , y , z (c), zkosení rovin xy , yx (d), zkosení rovin xz , zx (e), zkosení rovin yz , zy (f)



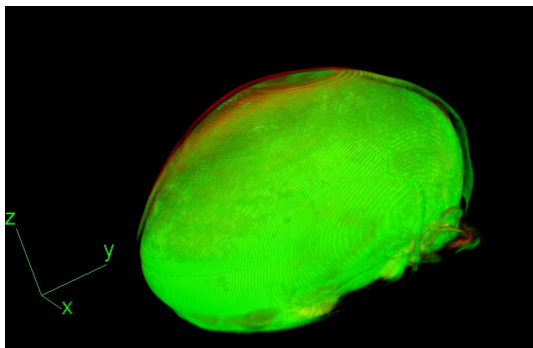
Obr. 3.11: Ukázka registrace (afinní): transverzální pohled před registrací (a), transverzální pohled po registraci (b), sagitální pohled před reg. (c), sagitální pohled po reg. (d), frontální pohled před reg. (e), frontální pohled po reg. (f) (zelená – referenční obraz, fialová – pohybový obraz, šedá – slícované obrazy)



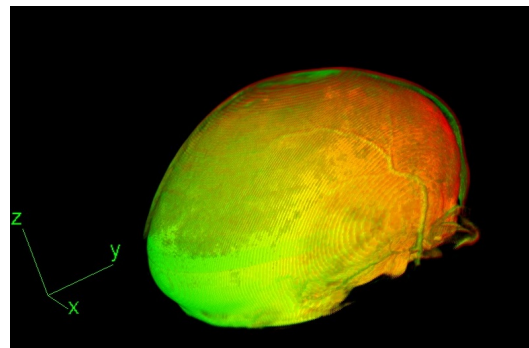
(a)



(b)



(c)



(d)

Obr. 3.12: Registrace reálných obrazů – obraz 1: před registrací (pohled a) (a), po registraci (pohled a) (b), před registrací (pohled b) (c), před registrací (pohled b) (d) (červená – referenční obraz, zelená – pohybový obraz, žlutá – slícované obrazy)

4 ZÁVĚR

V první části diplomové práce byla provedena literární rešerše problematiky registrace obrazů. Jmenovitě geometrických transformací, zejména afinní transformace, dále interpolačních metod včetně jejich definic pro 3D prostor, podobnostních kritérií a v neposlední řadě optimalizačních algoritmů. Detailněji byly rozebrány algoritmy řízeného náhodného prohledávání (CRS) a simulovaného žíhání (SA).

Druhou částí práce byla realizace programu v prostředí MATLAB®, který má za úkol nejdříve provést načtení dodaných obrazových dat ve formátu DICOM. Tento úkol byl zrealizován s využitím balíku příkazů pro práci s medicínskými daty formátu DICOM, který je přímo součástí programového prostředí MatLab®. Dalším úkolem programu je algoritmus pro odstranění patientského stolu a podložky pod hlavu. Problém byl vyřešen za pomoci dvou kroků, kde prvním krokem bylo prahování snímků s využitím Otsu metody pro nalezení optimálního prahu, druhým krokem bylo vytvoření binární masky za pomoci výpočtu maximální plochy v obrazu a následné vynásobení této masky s originálním obrazem. Hlavní částí programu je samotná geometrická transformace a optimalizační algoritmus CRS. Byl vytvořen výpočet afinní trojrozměrné transformace za použití jednoho cyklu a maticového násobení k výpočtu transformovaných souřadnic. Následně byly souřadnice interpolovány zvolenou interpolační metodou pomocí funkce trojrozměrné interpolace z programového prostředí MATLAB®. Optimalizační algoritmus umožňuje volbu podobnostního kritéria, hranic prohledávaného prostoru a heuristické metody. Ke snížení potřebného výpočetního času bylo použito paralelního zpracování dat a podvzorkování obrazu Haarovou vlnkou s využitím pyramidového přístupu. Posledním součástí je uložení transformované scény jako programové proměnné ve formátu **.mat*, případně jako nové sady DICOM snímků a následná prezentace dosažených výsledků. Program obsahuje grafické uživatelské rozhraní.

Součástí výpočtu je také měření výpočetního času prováděných úkonů, přičemž při kompletním průběhu algoritmu pro 224 snímků o velikosti 512x512 pixelů, s bitovou hloubkou 16 bitů odstínů šedi a při využití lineární interpolační metody trval výpočet registrace průměrně 1,5 hodiny. Tohoto času bylo dosaženo při paralelním zpracování dat a se čtyřnásobným podvzorkováním obrazů na počítačové stanici v konfiguraci 48 Gb RAM, 2x šesti jádrový procesor Intel Xeon E5649 2,53 GHz a grafická karta Nvidia Quadro 400. Veškeré části programu fungují dle předpokladů, jak je patrné z výše uvedených výsledků.

LITERATURA

- [1] DRASTICH, A. Netelevizní zobrazovací systémy. Brno, skriptum VUT v Brně, ISBN 80-214-1974-1, 2001
- [2] ELSEEN, P.A. VAN DEN, POL, E.-J.D., VIERGEVER, M.A. Medical Image Matching-a Review with Classification. IEEE Transaction on Engineering in Medicine and Biology Magazine, Vol. 12, No.1, pp. 26-39, 1993
- [3] JAN, J. Medical Image Processing, Reconstruction and Restoration: Concepts and Methods. CRC Press, ISBN 0-8247-5849-8, 2005
- [4] KIRKPATRICK, S., GELLAT, C. D. Jr., VECCHI, M. P. Optimisation by Simulated Annealing. Science, Vol. 220, No. 4598, pp. 671-680, May 1983
- [5] MAES, F., a kol. Multimodality Image Registration by Maximization of Mutual Information. IEEE Transaction on Medical Imaging, Vol. 16, No. 2, pp. 187-198, 1997
- [6] LAMOŠ, M. Modelování metod číslicového zpracování obrazu u digitální radiografie. Brno, VUT v Brně, 2010
- [7] MODERSITZKI, J. Numerical Methods for Image Registration. Oxford University Press, USA, ISBN: 0-1985-2841-8, 2004
- [8] OTSU, N. A Treshold Selection Method of Gray-Level Histograms. IEEE Transactions on System, Man and Cybernetics, Vol. 9, No. 1, pp. 62-66, 1979
- [9] PARKER, J.A., KENYON, R.V., TROXEL, D.E. Comparison of Interpolating Methods for Image Resampling. IEEE Transaction on Medical Imaging, Vol. 2, No. 1, pp. 31-39, 1983
- [10] PETER, R., MALINSKY, M., JAN, J. Internetová knihovna metod a algoritmů pro registraci a fúzi medicínských obrazů [online]. 2010, [cit. 29. 4. 2012]. Dostupné z URL: <<http://projects.ubmi.feec.vutbr.cz/loira>>
- [11] PRICE, W. L. A controled random search procedure for global optimisation. The Computer Journal, Vol. 20, No. 1, pp. 367-370, Feb 1976
- [12] TVRDÍK, J. Algoritmus řízeného náhodného prohledávání a alternující heuristiky. Automa, č. 1, s. 54-57, Led 2002
- [13] TVRDÍK, J. Evoluční algoritmy. skriptum VŠB-TU Ostrava, 2004
- [14] VONDRÁK, I. Neuronové sítě. skriptum VŠB-TU Ostrava, 2009

- [15] WALEK, P., JAN, J., OUREDNICEK, P., SKOTAKOVA, J., JIRA, I. Pre-processing for Quantitative Statistical Noise Analysis of MDCT Brain Images Reconstructed Using Hybrid Iterative (iDose) Algorithm. Journal of WSCG, Vol.20, No.1, pp.73-80, ISSN 1213-6972, Union Agency, 2012
- [16] ZITTOVÁ, B., FLUSSER, J. Image registration methods: a survey. Image and Vision Computing, Vol. 21, No. 11, pp. 977-1000, Oct 2003

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

3D	trojrozměrný
NN	nearest neighbor
BL	bilinear
TL	trilinear
MI	mutual information
SSD	Sum Squared Differences
CRS	Controlled Random Search
SA	Simulated Annealing
DICOM	obrazový formát užívaný v lékařství