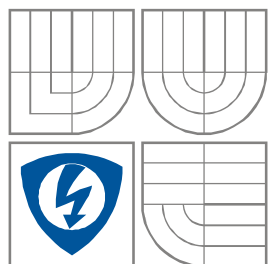




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF BIOMEDICAL ENGINEERING

## REGISTRACE OBRAZU POMOCÍ METODY OPTICAL FLOW

IMAGE REGISTRATION USING OPTICAL FLOW METHOD

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

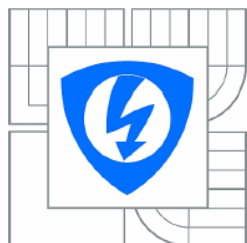
AUTOR PRÁCE  
AUTHOR

Bc. Jan Bistrý

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. Miloš Malínský

BRNO, 2011



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. Jan Bistrý  
**Ročník:** 2

**ID:** 98458  
**Akademický rok:** 2010/2011

**NÁZEV TÉMATU:**

**Registrace obrazu pomocí metody Optical Flow**

## POKYNY PRO VYPRACOVÁNÍ:

V rámci projektu prostudujte problematiku registrace obrazu pomocí metody Optical Flow. Navrhněte pro danou metodu program a odzkoušejte jej v programovém prostředí Matlab. Vyhodnoťte výsledky na testovacích datech dodaných vedoucím práce.

## DOPORUČENÁ LITERATURA:

- [1] JAN, J. Medical Image Processing, Reconstruction and Restoration - Concepts and Methods. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, 2005. 760 s. ISBN: 0-8247-5849-8.  
[2] HAJNAL, J.V., HILL, D.L.G., HAWKES, D.J. (Eds.), Medical Image Registration, CRC Press, Boca Raton, USA 2001

**Termín zadání:** 15.10.2010

**Termín odevzdání:** 20.5.2011

**Vedoucí práce:** Ing. Miloš Malínský

**prof. Ing. Ivo Provazník, Ph.D.**  
*Předseda oborové rady*

## UPOZORNĚNÍ:

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

## **Anotace**

Registrace medicínských obrazů je technika, která se rozvíjí ruku v ruce s novými hybridními diagnostickými zobrazovacími modalitami. V dnešní době je prováděna registrace jak monomodálních, tak multimodálních obrazových dat. První část této diplomové práce je zaměřena na problematiku registrace obrazů. Konkrétně na jejich transformace, interpolace, kritériální funkce a optimalizace. V následující části je představena realizovaná metoda Optical Flow a použitý algoritmus „Demon“ pro registraci obrazů. Dále se práce zaměřuje na prezentaci výsledného programu a realizovaného GUI. V poslední části diplomové práce je testována funkčnost navrženého programu na běžných obrazech i reálných CT snímcích.

## **Annotation**

Registration of medical images is technique, which is developing with new hybrid diagnostic imaging systems. Nowadays is a trend in image registration focused on monomodality and multimodality images registration. The first part of this master's thesis is intent on present basic information about image registration. Concretely is intending on image transformation, interpolation, criteria function and at optimalization. Further part present realized Optical Flow technique and used “Demon” algorithm for image registration. The next part is focused on presentation of program solution and GUI. In the last section of this master's thesis is verifying of created program on usual images and real CT images.

## **Klíčová slova**

Registrace obrazu, geometrická transformace, algoritmus „Demon“, Optical Flow, transformace, optimalizace, interpolace, podobnostní kritéria

## **Keywords**

Image registration, geometrical transformation, „Demon“ algorithm, Optical Flow, transformation, optimalization, interpolation, similar criteria

## **Bibliografická citace**

BISTRÝ, J. *Registrace obrazu pomocí metody Optical Flow*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 80s. Vedoucí diplomové práce Ing. Miloš Malínský

## **Prohlášení**

Prohlašuji, že svou diplomovou práci na téma Registrace obrazu pomocí metody Optical Flow 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 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.

V Brně dne 19. května 2011

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu diplomové práce Ing. Miloši Malínskému za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 19. května 2011

.....  
podpis autora

# Obsah

1. Úvod .....	6
1.1 Registrace obrazů .....	6
1.1.1 Transformace .....	7
1.1.2 Interpolace .....	12
1.1.3 Kriteriaální funkce .....	14
1.1.3.1 Lokální podobnostní kritéria .....	14
1.1.3.2 Globální podobnostní kritéria .....	20
1.1.4 Optimalizace .....	21
2. Metoda Optical Flow .....	23
2.1 Základní vlastnosti .....	23
2.2 Diferenční techniky .....	25
2.2.1 Metoda Horn&Schunk .....	26
2.2.2 Metoda Lucas&Kanade .....	27
2.2.3 Metoda Nagel .....	28
2.3 Algoritmus "Demon" .....	29
3. Programové řešení .....	37
3.1 Bloková schémata realizovaných registračních postupů .....	37
3.2 Struktura programu .....	39
3.2.1 Struktura programu iteračního registračního postupu .....	40
3.2.2 Struktura programu automatického registračního postupu .....	42
3.2.3 Presentace vytvořeného grafického uživatelského rozhraní GUI .....	45
4. Presentace a zhodnocení dosažených výsledků .....	48
5. Závěr .....	75
Seznam obrázků .....	77
Seznam grafů .....	78
Seznam zkratk .....	78
Seznam příloh .....	79
Reference .....	79

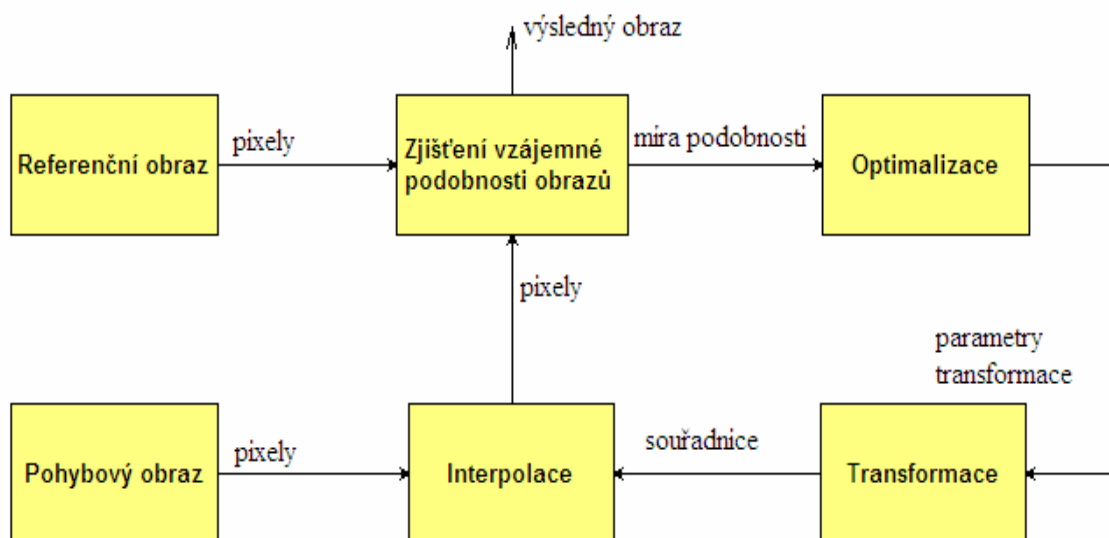
# 1.Úvod

Cílem této diplomové práce bude prostudování problematiky registrace obrazů pomocí metody Optical Flow. Součástí práce bude i návrh a realizace programu v prostředí Matlab. Funkčnost navrženého a realizovaného programu bude otestována na datech, která byla dodána vedoucím diplomové práce. V úvodní kapitole budou prezentovány základní informace vztahující se k registraci obrazů. Následující kapitola již bude zaměřena na metodu Optical Flow a realizovaný algoritmus „Demon“. Následně bude prezentováno vytvořené programové řešení a GUI. V poslední části této diplomové práce bude ověřena funkčnost realizovaného programu a budou zhodnoceny dosažené výsledky.

## 1.1 Registrace obrazů

Registraci obrazů obecně rozumíme slícování dvou (nebo i více) obrazů, které jsou vůči sobě vzájemně posunuty. Tyto změny mohou nastat v důsledku pohybu snímaného pacienta nebo např. fyziologickými či patologickými deformacemi měkké tkáně v získaných obrazech. Procesem registrace se snažíme nalézt geometrickou transformační funkci, která by vhodně popsala změny v jednotlivých obrazech. Výslednou transformační funkci můžeme použít k co nejlepšímu slícování celých obrazů, případně pouze jejich částí. Při hledání této transformační funkce se obraz posouvá mimo původní obrazovou mřížku, je tedy nezbytné provádět vhodnou interpolaci. Obrazy, které mají být mezi sebou registrovány mohou být monomodální či multimodální. V poslední době jsou čím dál více v oblibě hybridní zobrazovací modalitty. Potřebujeme tedy vytvořit vhodný algoritmus pro registraci obrazů, který multimodální obrazová data přesně slícuje. Aby byla diagnostická hodnota výsledných obrazů co nejvyšší, je nutné realizovat co nejpřesnější registraci obrazů. Velmi často se registrace obrazů využívá při pozorování identické scény měnící se v čase (např. nádorová onemocnění, která se postupně zvětšují). V praxi je možné využít algoritmů registrace obrazů k plánování radiační terapie (viz např. studie [12] ). Při registraci je jeden obraz označen jako referenční a druhý jako pohybový.

Na následujícím obrázku se nachází obecné blokové schéma procesu registrace obrazů:



Obrázek 1. – Blokové schéma procesu registrace obrazů

Ve stručnosti uvedme popis blokového schématu registrace obrazů. Na začátku procesu registrace zvolíme referenční a pohybový obraz. Pomocí zvolené kriteriální funkce zjistíme počáteční podobnost obrazů. V následujícím bloku dochází k optimalizaci parametrů transformace. Následně je obraz transformován a interpolován. Opět se pomocí kriteriální funkce zjistí vzájemná podobnost obrazů. Je-li podobnost nedostačující celý proces registrace se opakuje, je-li dostatečná, získáváme výsledný registrovaný obraz.

V úvodní části této diplomové práce budou postupně popsány funkce jednotlivých bloků obecného schématu registrace obrazů.

### 1.1.1 Transformace

Blok *transformace* slouží k nalezení vhodné transformační funkce mezi obrazy. Výstupem geometrické transformace je tedy mapovací funkce  $T$  mezi prostorovými souřadnicemi referenčního obrazu  $A$   $r = (x, y)$  a souřadnicemi pohybového obrazu  $A'$   $r' = (x', y')$  (viz [1]).

$$r' = T(r). \quad (1.1)$$

V případě, že uvažujeme mezi registrovanými obrazy pouhé posunutí, jsou hodnoty transformovaného obrazu  $A'$   $f(x', y')$  stejné jako v původním obrazu  $A$   $f(x, y)$ , což je možno zapsat následujícím způsobem (viz [1])

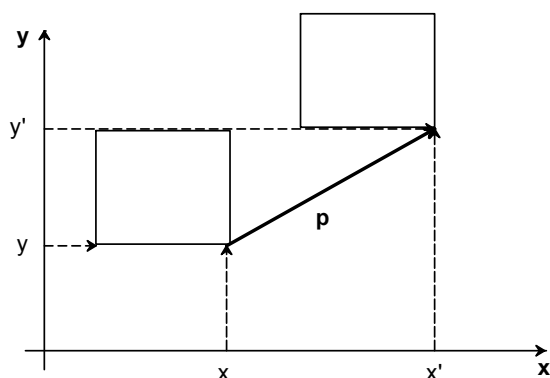
$$f(r') = f(T(r)). \quad (1.2)$$

O globální transformaci mluvíme tehdy, hledáme-li transformační funkci mezi celými obrazy. V případě potřeby je možné obrazy rozdělit na příslušné oblasti a pro každou oblast najít vhodnou transformační funkci zvlášť. Toto rozdělení je možné provést např. pomocí *piece-wise* transformace, která rozděluje obraz na trojúhelníkové oblasti.

**Rigidní transformace** - V případě rigidní transformace dochází mezi referenčním a pohybovým obrazem ke vzájemnému posunu, rotaci či kombinaci posuvu a rotace.

Nejjednodušším případem je prostá translace (posunutí), která posouvá obraz vůči počátku souřadnic. Translaci je možno zapsat následujícím způsobem (viz [1]):

$$r' = r + \Delta p, \quad \text{kde } \Delta p = [\Delta x, \Delta y]. \quad (1.3)$$



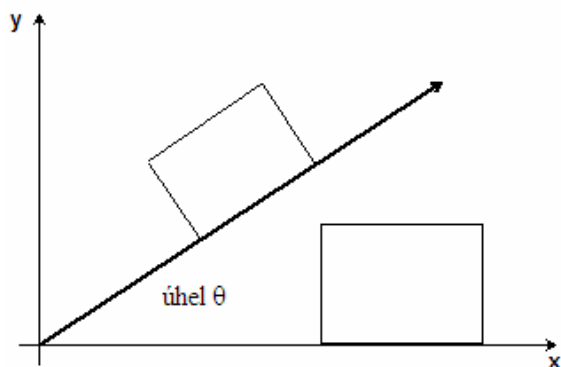
Obrázek 2. – Příklad prosté translace na jednoduchém objektu (převzato a upraveno z [3])

Dalším zmíněným případem je rotace. Dochází při ní k pootočení mezi referenčním a pohybovým obrazem. Rotaci je možno zapsat vztahem (viz [1])

$$r' = Br, \quad (1.4)$$

kde B je matice definující úhel natočení, jejíž tvar je následující (viz [1])

$$B = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \quad (1.5)$$



Obrázek 3. – Příklad rotace na jednoduchém objektu (převzato a upraveno z [3])

Ve většině případů dochází k translaci i rotaci současně. Cílem je tedy nalézt 6 transformačních parametrů, kterými jsou předchozí případy vyjádřeny. Vycházíme pak z matice (viz [1])

$$r' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & \Delta x \\ \sin\theta & \cos\theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (1.6)$$

Obecně lze říci, že čím je vazba mezi referenčním a pohybovým obrazem složitější, tím složitější je i výsledná soustava rovnic transformační funkce a je tedy potřeba určit i více parametrů.

**Flexibilní transformace** - Jedná se o transformaci při níž dochází ke změnám vzdáleností mezi body obrazu. Obrazy mohou být vůči sobě zvětšeny nebo naopak zmenšeny. Základní typ této transformace je rovinná změna měřítka. Můžeme se zde setkat i se zkosením obrazu, které je způsobeno posunem sloupců nebo řádků v obrazové matici.

Měníme-li měřítko, dochází ke zvětšení (nebo zmenšení) obrazu ve směru os. Můžeme vyjít z následujícího zápisu (viz [1])

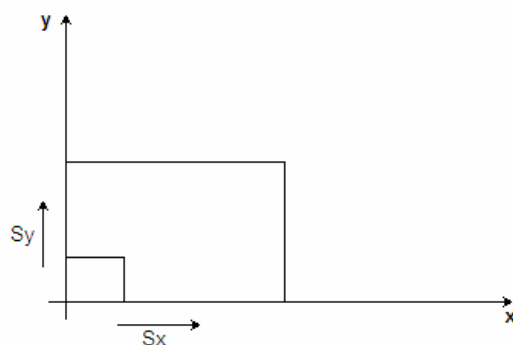
$$r' = Sr. \quad (1.7)$$

Písmeno S označuje matici (viz níže), obsahující konstanty, které ovlivňují zvětšení nebo zmenšení obrazu.



Tvar matice  $S$  je následující (viz [1])

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.8)$$



Obrázek 3. –Změna měřítka jednoduchého objektu (převzato a upraveno z [3])

Nyní se dostáváme k dalšímu případu flexibilní transformace, kterým je transformace měřítka. Jedná se o kombinaci rigidní transformace a změny měřítka. Tuto transformaci je možno popsat následujícím způsobem (viz [1])

$$r' = SRr. \quad (1.9)$$

Maticový zápis této transformace má dle výše uvedených vztahů následující tvar (viz [1])

$$SR = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & \Delta x \\ \sin\theta & \cos\theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_x \cos\theta & -s_x \sin\theta & s_x \Delta x \\ s_y \sin\theta & s_y \cos\theta & s_y \Delta y \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.10)$$

Definujme nyní zkosení obrazů. Jak již bylo uvedeno výše, dochází při něm k posunu řádků a sloupců v obrazové matici. Zkosení je definováno maticí  $G$  (viz [1])

$$G = G_x G_y, \text{ přičemž } G_x = \begin{bmatrix} 1 & g_{xy} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ a } G_y = \begin{bmatrix} 1 & 0 & 0 \\ g_{xy} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.11)$$

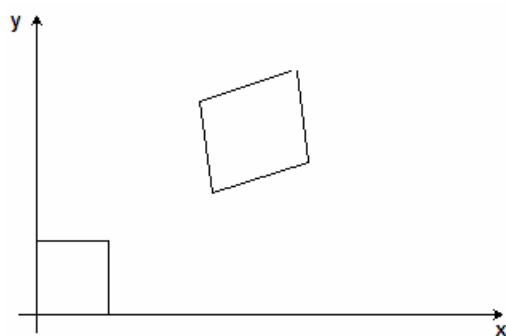
Mezi flexibilní transformace patří i transformace affíní. Tu tvoří kombinace zkosení, změny měřítka a dále kombinace rotace a translace. Obecně ji můžeme zapsat jako (viz [1])

$$r' = GSRr = Ar. \quad (1.12)$$

Odpovídající maticový zápis afiní transformace pro dvojrozměrné obrazy je následující (viz [1])

$$A = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.13)$$

Jak je z uvedeného maticového zápisu zřejmé, afiní transformace je popsána šesti nezávislými parametry. Při afiní transformaci se mění úhly mezi plochami a čarami, zůstává však zachována rovnost povrchu a čar i rovnoběžnost mezi nimi.



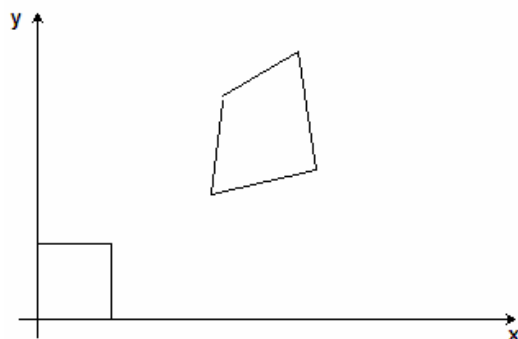
Obrázek 4. – Příklad afiní transformace na jednoduchém objektu (převzato a upraveno z [3])

Výše uvedené transformace patřily do skupiny lineárních transformací. Díky jejich linearitě lze snadno nalézt jednotlivé parametry.

Do skupiny nelineárních geometrických transformací patří transformace projektivní. Její základní vlastností je, že zachovává rovnost povrchů a čar, avšak rovnoběžnost mezi nimi zachována není. Tuto transformaci popisujeme více parametry než transformaci afiní, díky čemuž získáváme větší flexibilitu i složitost. Transformaci můžeme popsat následujícím vztahem (viz [1])

$$r' = \frac{Ar}{p^T r + \alpha}, \text{ kde } p^T = [p_1 \ p_2]. \quad (1.14)$$

Parametry transformace jsou  $A$ ,  $p$  a  $\alpha$ . V čitateli se nachází běžná afiní transformace, která je dělena hodnotou závislou na parametru  $r$ . Tímto způsobem je do transformace vnesena nelinearita.

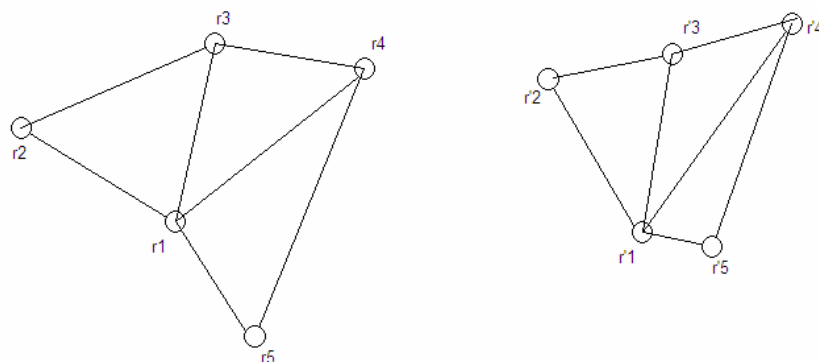


Obrázek 5. – Příklad projektivní transformace na jednoduchém objektu (převzato a upraveno z [3])

Mezi další druhy flexibilní transformace patří např. polynomiální transformace, perspektivní transformace nebo transformace využívající radiálních bazových funkcí.

Výše uvedené transformace patří do skupiny globálních transformačních funkcí. Uvedme nyní příklad lokální transformační funkce.

**Piece-Wise transformace** – Cílem této transformace je rozdělení obrazu na nepřekrývající se plochy různých velikosti, pro něž hledáme individuální transformační funkce. Oblasti můžeme rozdělit pomocí např. segmentace obrazu nebo pomocí parametrů oblastí (např. v oblasti přechodu kost-tkán).



Obrázek 6. – Příklad rozmístění řídicích bodů a vytvoření oblastí pro piece-wise transformaci (převzato a upraveno z [1])

Na předchozím obrázku je znázorněno možné uspořádání řídicích bodů  $r$  v referenčním obrazu a uspořádání řídicích bodů  $r'$  umístěných v pohybovém obrazu. Pro každou oblast, v našem případě trojúhelník, hledáme odpovídající transformační funkci. Popis této transformace je následující (viz [1])

$$\tilde{r}'_{k,i} = T(\tilde{r}_{k,i}) = A_{k,i} r_{k,i}, \text{ kde } i=1,2,3\dots \quad (1.15)$$

Obecně můžeme rozložení v obrázku č. 6 popsat soustavou pěti lineárních rovnic (viz [1])

$$\begin{aligned} x'_{k,i} &= a_{1,1}x_{k,i} + a_{1,2}y_{k,i} + t_1 \\ y'_{k,i} &= a_{2,1}x_{k,i} + a_{2,2}y_{k,i} + t_2, i=1,2,3\dots \end{aligned} \quad (1.16)$$

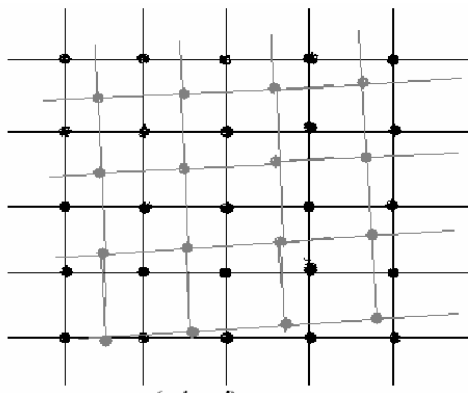
### 1.1.2 Interpolace

Další důležitou součástí procesu registrace obrazů je interpolace. Jejím cílem je převedení bodů obrazů do diskrétní obrazové mřížky. V této části práce uvedeme čtyři základní interpolační metody – nejbližší soused (NN), aproximace jasové funkce, bilineární a bikubická interpolace. Uvedené metody patří do skupiny interpolací v prostorové oblasti. Při nevhodně provedené interpolaci se mohou ve výsledném obrazu objevit artefakty či zkreslení.

**Aproximace jasové funkce** - Jedná se o základní interpolační úlohu. Cílem je převést obraz, který je posunut o neceločíselnou hodnotu, do obrazové mřížky. K tomu se využívají jasové hodnoty jednotlivých pixelů. Při této metodě předpokládáme, že je možné nalézt přesný rekonstrukční vztah mezi obrazovými mřížkami. U tohoto předpokladu musí být splněn vzorkovací teorém a za uzly mřížky považujeme vzorkovací body. Mějme oblast několika bodů a proložme je vhodnou aproximační funkcí. Výsledek aproximační funkce nám poskytne hledaný bod. Za aproximační funkci obvykle volíme polynom  $n$ -tého řádu. Mějme tedy diskrétní obrazová data  $f(k\Delta x, i\Delta y)$ , po aproximaci bude mít výsledná funkce následující tvar (viz [1])

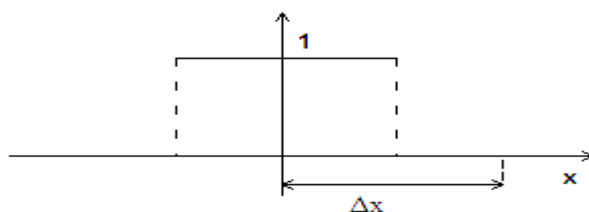
$$f_r(x, y) = \sum_i \sum_k f(k\Delta x, i\Delta y) r(x - k\Delta x, y - i\Delta y). \quad (1.17)$$

$\Delta x, \Delta y$  jsou kroky prováděné v jednotlivých směrech a  $r$  je zvolené interpolační jádro. Aby nedocházelo k početní náročnosti celého procesu aproximace, volíme jako interpolační jádro pouze omezený počet sousedících bodů.

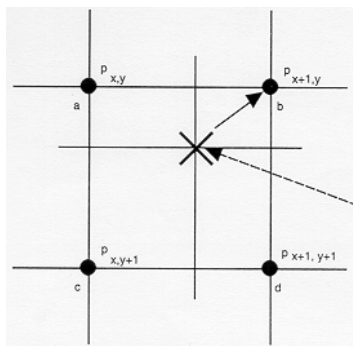


Obrázek 7. – Příklad vychýlení od diskrétních bodů obrazové mřížky (převzato a upraveno z [1])

**Interpolace nejbližší soused (NN)** - Jedná se o nejrychlejší a nejjednodušší interpolační metodu. Hodnota uzlu odpovídá hodnotě jasu nejbližšího bodu, který se nachází v posunuté mřížce. Princip této metody je patrný z obrázku 9.



Obrázek 8. – Jednorozměrné interpolační jádro metody NN (převzato a upraveno z [1])

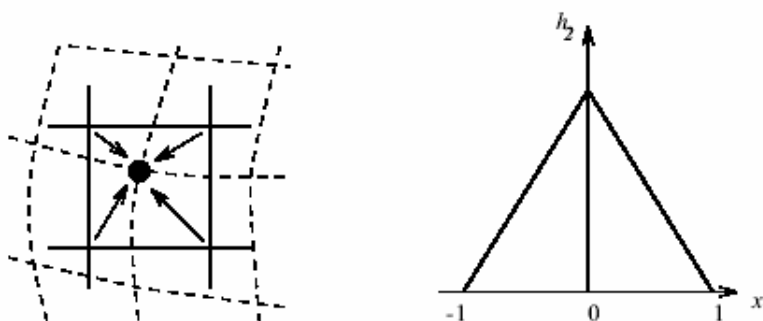


Obrázek 9. – Příklad interpolace nejbližší soused (převzato a upraveno z [3])

Obecně je možné interpolační funkci NN zapsat následujícím jednoduchým vztahem (viz [3])

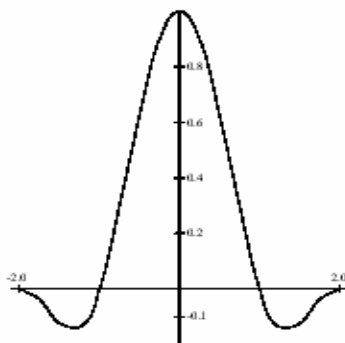
$$f_r(x, y) = f(\text{zaokrouhlení}(x), \text{zaokrouhlení}(y)). \quad (1.18)$$

**Bilineární interpolace (BL)** - Jedná se o náročnější interpolační metodu, která však dosahuje podstatně lepších výsledků než základní interpolace NN. K výpočtu jednoho bodu obrazové mřížky se používají hodnoty ze 4 bodů. Využívá se proložení jasové funkce třemi přímkami vynesené ze čtyř okolních bodů, přičemž výsledný bod je dán jejich váženým součtem. Váhy jednotlivých bodů jsou stejné.



Obrázek 10. – Princip bilineární interpolace a její interpolační jádro (převzato z [3])

**Bikubická interpolace** – Touto interpolační metodou, která je z výše uvedených nejsložitější, získáme obrazy nejlepší kvality (nejhladší), je však nejnáročnější na výpočet. Výsledná hodnota hledaného bodu je určena na základě aproximace hodnot z dalších 16 bodů. K aproximaci se využívá polynom třetího řádu.



Obrázek 11. – Interpolační jádro bikubické interpolace (převzato z [3])

### 1.1.3 Kriteriaální funkce

Jedná se o velmi důležitou oblast v procesu registrace obrazů, která slouží k výpočtu podobnosti mezi dvěma nebo více porovnávanými obrazy nebo jejich oblastmi. Podobnost se mezi jednotlivými obrazy vypočte po provedení interpolace. Dle velikosti porovnávaných oblastí rozlišujeme dvě základní skupiny kriteriaálních funkcí a to lokální a globální. Při globální podobnosti hodnotíme podobnost celé obrazové plochy a u lokální podobnosti se zaměřujeme na oblasti v obrazech. Neméně důležité je rozdělení dle parametru, se kterým dané kritérium pracuje. Kriteriaální funkce mohou být založeny na intenzitě, informačním obsahu, povrchu obrazu a sledovaných bodech v obrazech.

#### 1.1.3.1 Lokální podobnostní kritéria

U lokálních podobnostních kritérií mluvíme o porovnávání určených oblastí. Velikost těchto oblastí není ničím omezena, obvykle se však porovnávají oblasti jednotlivých obrazů zcela překrývají. Oproti globálním podobnostním kritériím bývají tyto oblasti menší a nebývají tak náchylné na jednotlivé změny parametrů registrace. V následující části této podkapitoly budou uvedena kritéria založená na intenzitě v obrazech a na informačním obsahu obrazů.

#### Podobnostní kritéria založená na intenzitě v obrazech

Dříve než přejdeme k definicím jednotlivých kritérií, uveďme nejprve pár základních pojmů. Mějme obrazové plochy **A** a **B** o různých intenzitách, které obsahují konečný počet pixelů. Většina lokálních kritérií vyžaduje stejnou velikost porovnávaných oblastí. Plochu **A** je možné charakterizovat vektorem **a**. Tento vektor je tvořen hodnotami  $N$  pixelů, které jsou uspořádány např. do řádku. Plocha **B** je definována stejným způsobem, tedy vektorem **b**, jehož hodnoty mohou být uspořádány např. do sloupce. Vzniklý vektorový prostor nese informace o intenzitních hodnotách jednotlivých pixelů. Hodnotu každého pixelu můžeme vyjádřit vektorem. Mluvíme tedy o intenzitním vektorovém prostoru. Tato úvaha je pro intenzitní podobnostní kritéria stejná. Snažíme se tedy o nalezení co nejvíce se podobajících vektorových polí. [1]

Kritéria, která budou uvedena, jsou primárně založena na porovnávání obrazů s podobným kontrastem. Můžeme tedy již teď říci, že jsou nejvhodnější pro porovnávání monomodálních obrazů nebo jejich oblastí. Vycházíme z předpokladu, že porovnávané dvojice bodů v jednotlivých obrazech mají podobnou intenzitu.

#### Euklidovský rozdíl

Jedná se o nejjednodušší kritérium, které se využívá ke zhodnocení značně rozdílných oblastí. Je založeno na Euklidovském rozdílu, který lze popsat následujícím způsobem (viz [1])

$$C_E(a, b) = |a - b| = \sqrt{\sum_{i=1}^N (a_i - b_i)^2} \quad (1.19)$$

Toto kritérium je značně závislé na absolutních délkách porovnávaných vektorů.

#### Sum of squared differences (SSD)

Jedná se o podobné kritérium jako Euklidovský rozdíl, je však přesnější. SSD je značně ovlivňováno rozdíly v absolutních délkách porovnávaných vektorů (stejně jako u Euklidovského rozdílu). Jedná se o kritérium, které je velmi citlivé na změny kontrastu

v porovnávaných obrazech a na šumu. Výsledné kritérium se může měnit při lineární změně kontrastu v porovnávaném obraze. Praktické využití této metody bývá pouze u porovnávání obrazů či oblastí, které jsou si velmi podobné. Pokud se obrazy rovnají, je součet kvadrátů rozdílů intenzit všech uvažovaných bodů nulový. Snažíme se tedy nalézt minimum následující funkce (viz [3])

$$SSD(a, b) = \frac{1}{N} \sum_i^N |a_i - b_i|^2, \text{ kde } N \text{ je počet shodujících se pixelů.} \quad (1.20)$$

### Cross-correlation

Toto podobnostní kritérium vzniká skalárním součinem vektorů  $\mathbf{a} \cdot \mathbf{b}$ . V literatuře bývá často doporučováno, je však značně nespolehlivé (viz [1],[2]). Zmíněná nespolehlivost je způsobena zanedbáním některých členů odvozeného vzorce pro výpočet cross-correlation kritéria. Tvar kritériální funkce je následující (viz [1])

$$C_C(a, b) = \sum_{i=1}^N (a_i b_i) = \frac{1}{N} \sum_{i=1}^N (a_i b_i). \quad (1.21)$$

### Kovarianční kritérium

Jedná se o kritérium odvozené od cross-correlation, které poskytuje přesnější hodnoty. Rozložíme-li následující výraz (viz [1])

$$\sum_{i=1}^N (a_i b_i) = \sum_{i=1}^N (a_i - \bar{a})(b_i - \bar{b}) + \sum_{i=1}^N \bar{a} \bar{b}, \quad (1.22)$$

$$\text{kde } \bar{a} = \frac{1}{N} \sum_{i=1}^N a_i \text{ a } \bar{b} = \frac{1}{N} \sum_{i=1}^N b_i$$

a odstraníme  $\sum_{i=1}^N \bar{a} \bar{b}$  získáme vzorec pro kovarianční kritérium, který má tento tvar (viz [1])

$$C_{CV}(a, b) = \sum_{i=1}^N (a_i - \bar{a})(b_i - \bar{b}). \quad (1.23)$$

### Úhlové (Cosinové) kritérium

Další možnost výpočtu kritériální funkce je založena na rozdílném úhlu mezi vektory obrazů  $\mathbf{a}$  a  $\mathbf{b}$ . Zde je zajímavé, že kritérium je naprosto odolné vůči lineárním změnám kontrastu (viz [1][2]). To je dáno předpokladem, že stejnou orientaci mohou mít i vektory, které mají rozdílnou intenzitu vzniklou multiplikativním faktorem. Experimentálně bylo ověřeno, že je možné tento předpoklad využít i při nelineární monotónní kontrastní transformaci [1]. Toto kritérium bývá často normalizováno v rozsahu od -1 do 1 a je možné jej využít při porovnávání malých změn v obrazech. Formulace kosinového kritéria je následující (viz [1]).

$$C_A(a, b) = \frac{|\mathbf{a}| |\mathbf{b}| \cos(\delta)}{|\mathbf{a}| |\mathbf{b}|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2} \sqrt{\sum_i b_i^2}} \quad (1.24)$$

### Korelační koeficient

Tato kritériální funkce, pracující ve vektorovém prostoru, je často používána a je založena na odečítání hodnot vektorů. Zápis funkce je následující (viz [1])

$$C_{cc}(a,b) = \frac{\sum_i (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_i (a_i - \bar{a})^2 \sum_i (b_i - \bar{b})^2}}, \text{ kde } \bar{a} = \frac{1}{N} \sum_i a_i \text{ a } \bar{b} = \frac{1}{N} \sum_i b_i \quad (1.25)$$

Jedná se v podstatě o složitější variantu úhlového kritéria. Je nutné stanovit hodnoty jednotlivých oblastí a odečíst je. Výsledné hodnoty tohoto kritéria bývají normalizovány, přičemž se snažíme nalézt maximum této funkce.

Z výše uvedených kritérií je nejspolehlivější a nejodolnější úhlové kritérium. V praxi se velmi často využívá i korelační koeficient. Z mnohých praktických studií a výzkumů, které se zabývaly problematikou kritériálních funkcí založených na intenzitních hodnotách jednotlivých pixelů, bylo zjištěno, že nejhorších výsledků a tedy největších chyb dosahují kritéria kovarianční a cross – corelation (dle [1],[2]).

Běžně se výše uvedená kritéria využívají při monomodální registraci, tedy při registraci obrazů získaných z jedné zobrazovací modalitě např. CT, RTG apod. Následující úvahou bude jednoduše ukázáno, že výše uvedená kritéria, která jsou založena na vektorovém součinu a délce vektorů, je možné využít i při multimodální registraci. V případě multimodální registrace (jeden registrovaný obraz je získán např. z CT a druhý např. z PET) se obrazy, které mají být porovnány vyjadřují ve vektorových hodnotách tedy (viz [1])

$$A = [a_{i,k}] = [(a_1, a_2, a_3, a_4, \dots, a_n)_{i,k}], \text{ a } B = [b_{i,k}] = [(b_1, b_2, b_3, b_4, \dots, b_n)_{i,k}]. \quad (1.26)$$

Pro zjednodušení porovnávání uvažovaných vektorů se vytváří nové vektory oblastí získaných z jednotlivých zobrazovacích modalit. Tedy (viz [1])

$$a = [a_1^T, a_2^T, a_3^T, a_4^T, \dots, a_n^T]^T \text{ a } b = [b_1^T, b_2^T, b_3^T, b_4^T, \dots, b_n^T]^T \quad (1.27)$$

Dále můžeme jednoduše odvodit, že (viz [1])

$$ab = [a_1^T, a_2^T, a_3^T, a_4^T, \dots, a_n^T]^T [b_1^T, b_2^T, b_3^T, b_4^T, \dots, b_n^T]^T = a_1 b_1 + a_2 b_2 + a_3 b_3 + \dots + a_n b_n \quad (1.28)$$

a tedy

$$|a| = |[a_1^T, a_2^T, a_3^T, a_4^T, \dots, a_n^T]^T| = \sqrt{|a_1|^2 + |a_2|^2 + |a_3|^2 + |a_4|^2 + \dots + |a_n|^2}. \quad (1.29)$$

Výše uvedené úhlové kritérium je pro multimodální registraci možné zapsat následujícím způsobem (viz [1])

$$C'_A(a,b) = \frac{ab}{|a|} = \frac{a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4 + \dots + a_n b_n}{\sqrt{|a_1|^2 + |a_2|^2 + |a_3|^2 + |a_4|^2 + \dots + |a_n|^2}}. \quad (1.30)$$



Velkou výhodou těchto přístupů je snížení početní náročnosti. Úhlové kritérium je možné vypočítat pro jednotlivé modalit zvlášť nebo dohromady.

#### Koeficient stejnorodosti obrazu (RUI)

Při monomodálním porovnávání obrazů se můžeme setkat s koeficientem stejnorodosti obrazu (RUI). Původním využitím tohoto přístupu byla registrace PET obrazů. V dnešní době byl tento přístup upraven a využívá se např. při porovnávání CT obrazů (dle [3]). Je založen na výpočtu poměrů intenzitních hodnot transformovaného a původního obrazu, přičemž se snažíme o minimalizaci funkce (viz[1])

$$r = \begin{bmatrix} \frac{a_1}{b_1} & \frac{a_2}{b_2} & \frac{a_3}{b_3} & \frac{a_4}{b_4} & \dots & \frac{a_N}{b_N} \end{bmatrix}^T. \quad (1.31)$$

Stejně jako bylo uvedeno u úhlového kritéria, je i tento přístup odolný vůči lineárním změnám kontrastu. Vzorec pro výpočet kritéria má následující podobu (viz[1])

$$C_{RIU}(a, b) = \frac{\sqrt{\sum_i (r_i - \bar{r})^2}}{\bar{r}}, \text{ kde } \bar{r} = \frac{1}{N} \sum_i r_i. \quad (1.32)$$

#### Kritérium lokální charakteristiky (local feature)

Jedná se o kritérium využívající parametrické obrazy. Základní vlastností je porovnávání obrazů (nebo jejich oblastí) o vypočtené intenzitě s vytvořenými parametrickými obrazy. Tyto parametrické obrazy vznikají vynásobením obrazu se zvoleným lokálním operátorem. Operátorem mohou být např. hranové detektory, filtry, průměrující jádra o různých velikostech aj. Vzniklé parametrické obrazy nevyužívají přímo původní rozsah informace o intenzitě. Touto metodou je možné porovnávat jak malé oblasti tak i jednotlivé pixely.

Vzorec pro výpočet kritéria má následující podobu (viz[1])

$$C_{LF}(\vec{a}_{rA}, \vec{b}_{rB}) = \left| \vec{a}_{rA} - \vec{b}_{rB} \right|. \quad (1.33)$$

V případě vhodně zvolených operátorů může dojít k lepšímu porovnání jednotlivých obrazů. Můžeme tedy mluvit o jistém druhu pre-processingu.

#### Podobnostní kritéria založená na informacích v obrazech

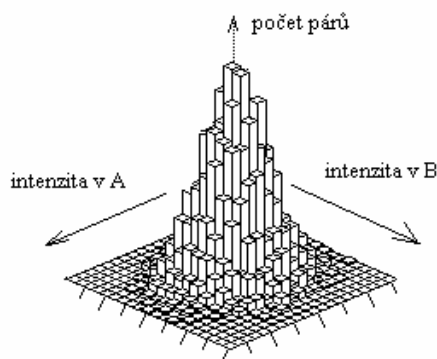
V případech, kdy nemůžeme využít údaje o intenzitě nebo je jejich použití nevhodné, vycházíme z metod založených na informačním obsahu obrazů. Předpokládáme, že hledanému bodu o jisté intenzitě odpovídá bod o intenzitě odlišné. Základem těchto metod je sdružený histogram, který nám poskytuje údaje o počtu pixelů jednotlivých intenzit, vyskytujících se v porovnávaných oblastech. Další využívaná metoda, která je založena na entropii, se nazývá Mutual Information (Vzájemná informace). Podívejme se nyní na jednotlivé přístupy podrobněji.

#### 2D sdružený histogram

Princip sdruženého histogramu je následující. Mějme porovnávané oblasti A a B, které mají stejnou velikost a jsou ve stupních šedi. Histogram je matice  $h^{AB}$ , jejíž velikost je dána rozměry  $q \times r$ , přičemž  $q$  je počet stupňů šedi v obraze A a  $r$  je stejné množství v obraze B. Na

jednotlivých pozicích histogramu se vyskytují odpovídající si hodnoty intenzit v obraze A a B. Sdružený histogram se vytváří prohledáním všech pozic v obrazech a zjištěním jednotlivých intenzit. Obsahuje tedy informace o intenzitách v porovnávaných obrazech na odpovídajících si pozicích. Sdružený histogram je možné využít jak pro porovnání celých obrazů, tak i k porovnání jednotlivých malých oblastí v těchto obrazech.

V případě, že jsou si porovnávané obrazy zcela rovny, leží prvky na hlavní diagonále matice sdruženého histogramu. Dále, pokud je jeden obraz odvozen od druhého pomocí jasové transformační funkce, jsou prvky matice soustředěny do křivky, jejíž průběh odpovídá právě dané jasové funkci. Pomocí sdruženého histogramu lze analyzovat podobnost mezi obrazy získanými rozdílnými zobrazovacími modalitami. Zde nastává problém se zobrazením jednotlivých tkání, neboť jejich intenzitní hodnota (šedotónová) může být různá u jednotlivých zobrazovacích modalit (např. CT, PET). Mezi multimodálními obrazy stejné geometrie existují jisté vztahy mezi intenzitními hodnotami. Těmito vztahy rozumíme shluknutí intenzitních hodnot v matici sdruženého histogramu. Pokud jsou obrazy podobné, dochází k těsnějšímu shluknutí, pokud jsou rozdílné, dochází naopak k jejich rozproštění.



Obrázek 12. – Příklad sdruženého 2D histogramu (převzato a upraveno z [1])

### Sdružená entropie

V tomto odstavci bude představena a popsána metoda založená na předpokladu pravděpodobnostního rozložení prvků a na teorii informace. Entropií myslíme míru informace, která je obsažena v obraze. Abychom lépe pochopili tuto metodu, uveďme nejprve její základní teoretické předpoklady. Předpokládejme nyní novou sdruženou proměnnou (A,B) mající vektorovou hodnotu (l,m). Tento vektor se skládá z hodnot intenzit v obrazech A a B na odpovídajících si pozicích jednotlivých pixelů. Nová proměnná má tedy následující tvar [viz(1)]

$$p_{AB} \{l, m\} \approx \frac{1}{N} h_{l,m}^{AB}. \quad (1.34)$$

Informační obsah je dán pravděpodobnostním rozložením mezi hodnotami jednotlivých pixelů [1]. Minimalizace sdružené entropie způsobuje shlukování intenzitních hodnot. Entropii můžeme rozdělit na samostatnou, která je označena  $H_A$  a  $H_B$  a sdruženou, označenou  $H_{AB}$ . Vzorce pro jejich výpočty jsou následující [viz(1)]

$$H_A = -\sum_{l=1}^q p(a_l) \log p(a_l), \quad H_B = -\sum_{m=1}^r p(b_m) \log p(b_m) \quad (1.35)$$

a

$$H_{AB} = - \sum_{l=1}^q \sum_{m=1}^r p(a_l, b_m) \log p(a_l, b_m). \quad (1.36)$$

Hodnoty vypočtené těmito vzorci udávají množství informace obsažené v samostatných obrazech A a B a dále množství informace při spojení AB. Je tedy zřejmé, že proměnná  $H_{AB}$  nám poskytuje informaci o podobnosti porovnávaných obrazů. V případě, že jsou jednotlivé obrazy zcela rozdílné, je hodnota parametru  $H_{AB}$  rovna součtu entropií jednotlivých obrazů. Naopak pokud si obrazy zcela odpovídají, je  $H_{AB}$  rovna nule.

### Mutual Information (Vzájemná informace)

Mutual Information je nepoužívanější multimodální porovnávací technikou (dle [2]). Tato metoda vychází z výše uvedeného předpokladu sdružené entropie a teorie informací. Jedná se o rozdíl mezi entropiemi samostatných obrazů  $H_A$ ,  $H_B$  a sdruženou entropií  $H_{AB}$  (viz [1])

$$I_{AB} = H_A + H_B - H_{AB}. \quad (1.37)$$

Oproti sdružené entropii se snažíme vypočítat maximální hodnotu této funkce. Jsou-li obrazy zcela rozdílné, je hodnota  $I_{AB}$  rovna nule. Dochází k odhadu podobnosti souborů dat. Tuto metodu je možné využít i při multimodální registraci obrazů. Hodnotu podobnosti MI je nutné stanovovat pouze z oblastí obrazů, které se vzájemně překrývají.

Princip této metody je založen na předpokladu, že na porovnávané obrazy, konkrétně na intenzity šedi, nahlížíme jako na náhodné veličiny. Vycházíme z předpokladu jistého pravděpodobnostního výskytu hodnot intenzitních složek v porovnávaných obrazech a z předpokladu již zmíněné entropie. Odhady pravděpodobností lze určit jako normované histogramy samostatných obrazů A a B a jako normovaný sdružený histogram AB.

Počáteční rozložení pravděpodobnosti normalizované l-tým řádkem (v případě obrazu B) a m-tým sloupce (v případě obrazu A) sdruženého histogramu, je možné zapsat následujícími způsoby (viz [1])

$$p_{B|A}\{m | l\} \approx \frac{1}{N_l} h_{l,m}^{AB}, \text{ kde } N_l = \sum_{m=1}^r h_{l,m}^{AB} = h_l^A \quad (1.38)$$

a

$$p_{A|B}\{l | m\} \approx \frac{1}{N_m} h_{l,m}^{AB}, \text{ kde } N_m = \sum_{l=1}^q h_{l,m}^{AB} = h_m^B. \quad (1.39)$$

Pro počáteční entropie samostatných obrazů A a B můžeme stanovit následující zápis (viz [1])

$$H_{A|m} = - \sum_{l=1}^q p_A(l | m) \log p_A(l | m) \text{ a } H_{B|l} = - \sum_{m=1}^r p_B(m | l) \log p_B(m | l). \quad (1.40)$$

Nyní zbývá určit počáteční sdruženou entropii. Uvedme tedy její příslušné vztahy (viz [1])

$$H_{B|A} = - \sum_{l=1}^q p_A(l) H_{B|l} = - \sum_{l=1}^q \sum_{m=1}^r p_{AB}(m, l) \log p_{B|A}(m | l) \quad (1.41)$$

podobně

$$H_{A|B} = -\sum_{m=1}^r p_B(m) H_{A|m} = -\sum_{m=1}^r \sum_{l=1}^q p_{AB}(l, m) \log p_{A|B}(l | m). \quad (1.42)$$

Takto provedené zápisy sdružených entropií poskytují informaci zbývající např. v obraze A, známe-li obraz B.

Zápis Mutual Information můžeme tedy rozepsat následujícím způsobem (viz [1])

$$I_{AB} = H_A - H_{A|B} = H_B - H_{B|A} \quad (1.43)$$

V případě, že chceme získat normalizované hodnoty MI, můžeme využít následující vzorec (viz [1])

$$I_{AB} = \frac{H_A + H_B}{H_{AB}} \quad (1.44)$$

Jak již bylo uvedeno výše, je možno tuto metodu vhodně využívat u multimodální registrace. Dále ji lze využít i při parametrické prezentaci porovnávaných obrazů.

MI kritérium bylo využito i při výsledném porovnávání podobnosti v realizovaném programu GUI. Nejedná se sice o porovnávání obrazů z rozdílných zobrazovacích modalit, ale o registraci obrazů prekontrastních a postkontrastních, toto kritérium však bylo využito z důvodu flexibility vytvořeného programu. Na postkontrastní obraz se změněnou jasovou složkou, která vzniká přidáním kontrastní látky, však bylo pohlíženo jako na obraz z jiné zobrazovací modalit.

### 1.1.3.2 Globální podobnostní kritéria

Nyní budou probrána základní globální podobnostní kritéria, konkrétně kritéria založená na intenzitě a informačním obsahu v obrazech, bodech v obrazech a povrchu obrazů.

#### Globální podobnostní kritéria založená na intenzitě v obrazech

Při porovnávání podobností celých obrazů můžeme využít všechny výše zmíněné přístupy založené na intenzitě v obrazech. Při globálním porovnávání dochází ke změnám překrytí, které vznikají v kroku optimalizace. Zde je patrný rozdíl oproti lokálním kritériím, kde se porovnávané oblasti téměř nemění. Změny intenzity mohou být v obou případech způsobeny vlivem interpolace.

Tato kritéria slouží k porovnávání monomodálních obrazů, které by si měly co nejvíce odpovídat nebo by měly mít lineárně závislé intenzity. Při globálním porovnávání dvou obrazů se nejčastěji využívá korelačního koeficientu ([2]), avšak volba použitého kritéria závisí především na řešeném problému.

#### Globální podobnostní kritéria založená na informačním obsahu v obrazech

V této skupině se při globálním porovnávání nejčastěji využívá Mutual Information. Jak již bylo uvedeno výše, hodí se tato metoda na porovnávání multimodálních obrazů. Obrazy z různých zobrazovacích modalit mohou mít rozdílné kontrasty pro dané tkáň. S tímto omezením se však MI bez problémů vyrovná. Nevýhodou může být větší časová a

početní náročnost při stanovování sdružených histogramů. Existuje však možnost jak tomuto zabránit a to snížit počet úrovní v histogramu např. na 32 nebo 64 úrovní.

### **Globální podobnostní kritéria založená na bodech („landmarks“) v obrazech**

Tato metoda je založena na doplnění obrazu určitými značkami („landmarky“), které slouží k provedení optimální registrace obrazů. Tyto značky se umísťují do významných míst v obrazech např. na hrany, přechody tkáň-kost, cévní řečiště apod. Aby bylo možné tuto metodu vhodně využít, provádí se před samotnou registrací většinou preprocessing obrazů (např. pro stanovení hran atd.). Pouze vhodné umístění značek zajistí úspěšnou registraci obrazů. Situace je mnohem jednodušší, pokud jsou do pacienta vloženy jisté markery.

### **Globální podobnostní kritéria založená na povrchu obrazů**

Jedná se o metodu, využívající ke stanovení podobnosti obrazů zvolené povrchy nebo oblasti zvolených povrchů. Tyto povrchy mohou být vhodně rozděleny např. piece-wise funkcí nebo jinými funkcemi. Metoda je vhodná i pro multimodální porovnávání. Např. máme-li obrazy s vyšším rozlišením získané z magnetické rezonance, a nižším rozlišením z PET. Na obraz z PET (nižší rozlišení) bude nahlíženo jako na soubor bodů a na obraz z magnetické rezonance (vyšší rozlišení) jako na spojitou plochu. Poté se pro každý bod z PET obrazu hledá nejbližší bod v obrazu z magnetické rezonance. Hledáme tedy odpovídající si dvojice. Tento přístup se využívá při registraci obrazů mozku a pouze při rigidní transformaci [1][2][20].

#### **1.1.4 Optimalizace**

Posledním blokem v námi uvažovaném schématu registrace obrazů je optimalizace. Jejím cílem je získat nejlepší hodnoty vektoru  $\alpha$  transformačních parametrů. Úspěšnou optimalizací získáme úplné překrytí a tedy úplnou registraci dvou či více obrazů, proto je v procesu registrace téměř klíčová. Zpracováváme-li konkrétní soubor obrazů, je třeba na něj aplikovat vyhovující optimalizační funkci. Optimalizaci je možné rozdělit do dvou základních skupin a to na techniky stochastické a deterministické.

*Stochastické techniky* zavádí do procesu optimalizace jistou náhodnost a pracují s pravděpodobnostmi. Výhodou je, že tyto metody mohou zpracovávat i nespojitě funkce.

*Deterministické techniky* se snaží analyticky nalézt minimum dané spojité funkce. Jedná se o klasické matematické postupy (např. Newtonova metoda). Pro názornost si uveďme alespoň jeden příklad od každé skupiny.

Optimalizace využívající klasickou Newtonovu metodu (deterministická technika): z obecné matematiky víme, že optimalizovaná funkce má v extrémech první derivaci rovnu nule. Cílem této metody je tento nulový bod nalézt. Platnost výsledků musí být následně ověřena na základě druhé derivace. Iterační vzorec Newtonovy optimalizační metody má následující tvar (viz [3])

$$x_{k+1} = x_k - H(x_k)^{-1} \nabla f(x_k), \quad (1.45)$$

kde  $\nabla f(x_k)$  je gradient v bodě  $x_k$ ,  $H(x_k)^{-1}$  je inverzní (Hessova) matice optimalizující bod  $x_k$ . Největší problém této metody je výpočet a samotné stanovení inverzní matice, jejíž hodnotu je dle iteračního vzorce nutné ověřovat při každém iteračním kroku, což je početně

velmi náročné. Je tedy výhodnější využít např. Kvazi-Newtonovské metody, kde dochází k numerickému výpočtu inverzní matice.

Optimalizace využívající simulované žíhání (stochastická technika): pro nalezení minima se zde využívá iterační postupu a tato metoda se jeví jako dobrý kompromis mezi optimalizací a početní složitostí. Tato metoda vychází z poznatků o tvorbě krystalů, které vznikají při řízeném procesu ochlazování. Snažíme se nalézt stav s co nejnižší energií. Jedná se o metodu odolnou vůči vzniku parazitních jevů.

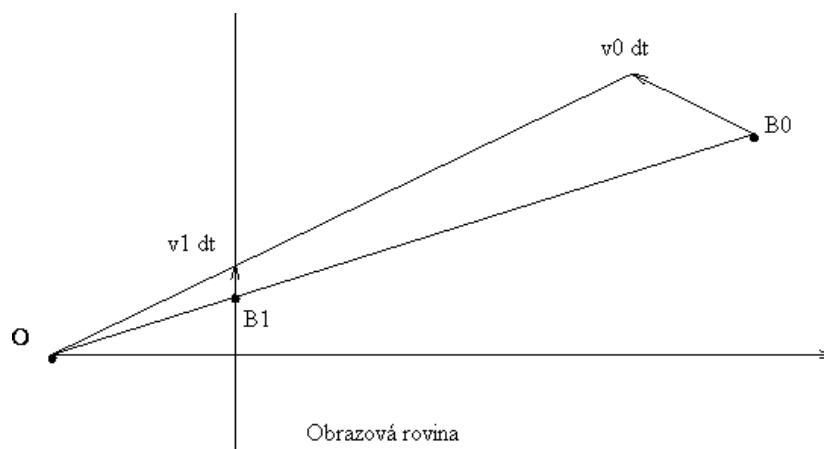
Označme si funkci, která má být optimalizována jako  $f(x)$  a její energii jako  $E = f(x)$ . Hodnota vektoru  $x$  je dána aktuálním stavem systému. Pro získání hodnoty v následujícím kroku vyžijeme náhodnou funkci  $G = (x, T)$ , která s určitou pravděpodobností  $h = (\Delta E, T)$  vygeneruje následující hodnotu  $x_n = G(x, T)$ .  $T$  označuje absolutní teplotu systému a  $\Delta E$  rozdíl energií předchozího a následně vygenerovaného stavu. Algoritmus této techniky je založen na postupném prohledávání a postupném ochlazování systému. Zajistíme-li dostatečně pozvolné chlazení, dojdeme ke globálnímu minimu optimalizované funkce.

## 2. Metoda Optical Flow

V první části této kapitoly budou probány základní vlastnosti této registrační metody a dále budou podrobněji popsány diferenční techniky pro výpočet Optical Flow. V druhé části kapitoly bude podrobně popsán algoritmus “Demon” a jeho kombinace s metodou Optical Flow. Tento algoritmus byl pro své dobré vlastnosti použit při ověření registrační metody na reálných CT obrazech, které byly dodány vedoucím této práce.

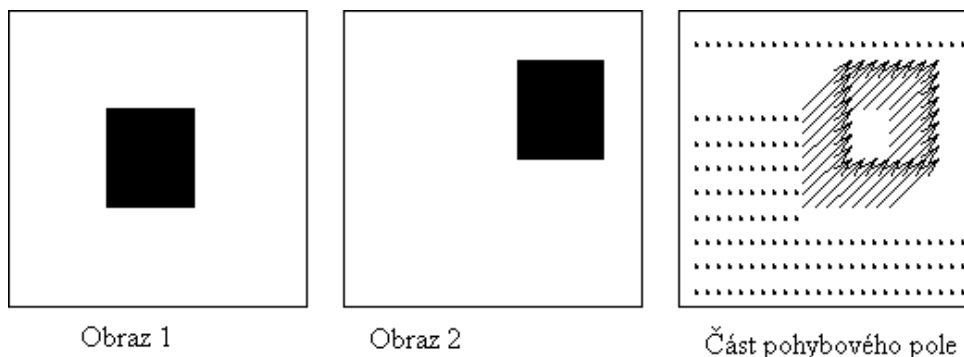
### 2.1 Základní vlastnosti

Pod pojmem Optical Flow rozumíme dvojdimenzionální pohyb, který je nutné detekovat ze sekvence obrazů. Vycházíme z předpokladu, že obsahy pohybujících se objektů v obrazech se v čase nemění. Změny intenzity  $f(x, y, t)$  jsou tedy dány pouze pohybem. V této metodě uvažujeme pole vektorů, které pro každý bod v daném obraze určuje intenzitu pohybu a jeho směr vůči dalšímu obraze. Optical Flow tedy úzce souvisí s pohybovým polem obrazu. Pokud se objekty před kamerou pohybují, vznikají odpovídající změny v samotném obraze. Jestliže se tedy např. bod  $B_0$  pohybuje rychlostí  $v_0$ , pak může být bod  $B_1$  popsán vektorem  $v_1$  v dané obrazové rovině. Soubor těchto vektorů pak nazýváme pohybovým polem. Toto pole je uvedeno na následujícím obrázku.



Obrázek 13. - Pohybové pole (převzato a upraveno z [7])

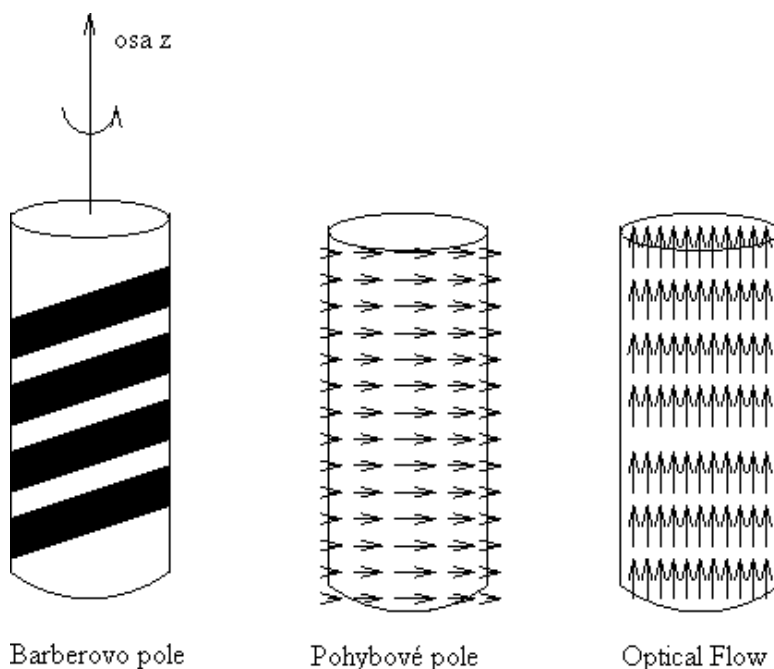
V případě, že uvažujeme v obraze pouze základní rotaci a translaci, bude pohybové pole spojitě, tedy kromě siluet hran v uvažovaných objektech.



Obrázek 14. - Příklad spojitěho pohybového pole (převzato a upraveno z [7])

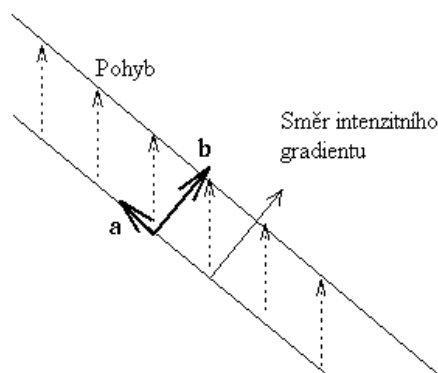
V případě plné translace kamery je směr pohybu určen paprskem, který promítá prostorový bod do obrazové roviny. Bod divergence nebo konvergence všech pohybových vektorů daného pole nazýváme ohniskem expanze. Pohybujeme-li se směrem ke snímanému objektu, dochází k jeho zvětšení, v případě pohybu od objektu, naopak k jeho zmenšení.

Pravděpodobně největší omezení metody Optical Flow je dáno tzv. problémem apertury. Tento problém nastává, měříme-li složku pohybu kolmou k jasovému gradientu. Příklad tohoto problému bude znázorněn na následujících obrázcích. Mějme tzv. Barberovo pole.



Obrázek 15. - Srovnání polí na Barberově rotačním válci (převzato a upraveno z [7])

Uvažujme Optical Flow jako pohyb jasových vzorů v obraze, které mohou, ale nemusí, odpovídat pohybovému poli. Na obrázku č. 15 je jasně patrný rozdíl mezi jednotlivými poli (simulováno na rotačním Barberově poli). Abychom byly schopni získat potřebné údaje o Optical Flow, je nutné jej měřit ve směru intenzitního gradientu. Parametry změřené v tečném směru vůči gradientu není možné získat. Tento problém je znázorněn na následujícím obrázku.



Obrázek 16- Zobrazení problému apertury(převzato a upraveno z [7])



Problém apertury je limitující z důvodu nedostatečné informace o směru a velikosti rychlosti pohybu. Z tohoto důvodu je nutné definovat další rovnici, z níž nalezneme parametry hledaného vektoru ve směru tečny. Z hlediska výpočtu je možno techniky Optical Flow rozdělit do následujících skupin:

- diferenční techniky
- porovnávací techniky
- energetické metody
- fázové metody

Společným znakem uvedených metod je předpoklad zachování intenzity.

## 2.2 Diferenční techniky

Jedná se o základní skupinu technik výpočtu Optical Flow. K získání vektoru rychlosti se využívá technika výpočtu v časoprostorové derivaci intenzity obrazu nebo pomocí filtrovaných obrazů. V závislosti na řešené úloze se využívají derivace prvního či vyššího řádu.

Nejprve vyjdeme z derivace prvního řádu, která je založena na translaci v obrazu (viz [7]):

$$I(x, t) = I(x - vt, 0), \quad \text{kde } v = (u, v)^T \quad (2.1)$$

a dále uveďme, že  $(u, v)$  je vektor, který udává směr a velikost rychlosti v určitém bodu obrazu a definuje tedy Optical Flow. Vyjdeme-li z předpokladu zachování intenzity  $dI(x, t)/dt = 0$ , je možné snadno odvodit rovnici gradientního omezení (viz [7]):

$$\nabla I(x, t) \cdot v + I_t(x, t) = 0, \quad (2.2)$$

kde gradient  $\nabla I(x, t)$  můžeme zapsat jako  $\nabla I(x, t) = (I_x(x, t), I_y(x, t))^T$ ,  $I_t(x, t)$  je parciální časová derivace  $I(x, t)$ . V rovnici 2.4 je uvedena normálová komponenta pohybu prostorových kontur při konstantní intenzitě v obrazu. Dále definujeme normálovou rychlost  $s$  a normálový směr  $n$ , přičemž platí vztah  $v_n = s \cdot n$ . Matematicky lze rychlost a směr popsat následujícími vzorci (viz [7]):

$$s(x, t) = \frac{-I_t(x, t)}{\|\nabla I(x, t)\|} \quad (2.3)$$

$$n(x, t) = \frac{\nabla I(x, t)}{\|\nabla I(x, t)\|} \quad (2.4)$$

Z rovnice 2.4 tedy dostaneme dvě neznámé  $v$ , definované pouze jednou lineární rovnicí. Pro nalezení obou parametrů  $v$  je nezbytné provést další omezení.

Dále je možné vycházet z diferenčních metod založených na derivacích druhého řádu. Opět platí zákon zachování intenzity. Řešení vede na soustavu rovnic (viz [7]):

$$\begin{bmatrix} I_{xx}(x,t) & I_{yx}(x,t) \\ I_{xy}(x,t) & I_{yy}(x,t) \end{bmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \begin{pmatrix} I_{tx}(x,t) \\ I_{ty}(x,t) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (2.5)$$

Metody, které využívají derivace druhého řádu, se využívají pouze zřídka, mnohem častěji vycházíme z derivací prvního řádu.

### 2.2.1 Metoda Horn&Schunk

Jedná se o diferenční metodu, pomocí níž se snažíme nalézt složky rychlosti označené jako  $u$  a  $v$ . Jak již bylo popsáno v předchozí části práce, pracujeme s určitým omezením. Tedy pro jeden pixel získáváme pouze jednu rovnici obsahující dvě neznámé. Cílem je zjistit posun pozorovaného bodu, označeného např.  $E(x,y,t)$ , vzhledem k dalšímu obrazu sekvence. Z obrazu je nutné získat dostatečné množství informací, které nám určí přesný posun bodu  $E(x,y,t)$ . Předpokladem metody je, že rychlost pohybu sousedních pixelů je velmi podobná. Tedy kromě hraničních oblastí pohybujících se objektů. Jedná se o předpoklad tzv. pomalých změn. Pomocí rovnice 2.6 je možno určit míru, s jakou se Optical Flow liší od předpokladu pomalé a spojitě změny v obrazu (viz [7])

$$S = \int_0^N \int_0^M \left( \frac{\delta u}{\delta x} \right)^2 + \left( \frac{\delta u}{\delta y} \right)^2 + \left( \frac{\delta v}{\delta x} \right)^2 + \left( \frac{\delta v}{\delta y} \right)^2 dx dy. \quad (2.6)$$

Z předchozího vzorce je patrné, že jde o součet druhých mocnin parciálních derivací, přičemž každá z nich odpovídá změně rychlosti v daném směru. Aby byl splněn předpoklad pomalých změn, je třeba nalézt minimální hodnotu tohoto výrazu. Vycházejme tedy z předpokladu, že  $E_x u + E_y v + E_t = 0$ , kde  $E_x$ ,  $E_y$  a  $E_t$  jsou parciální derivace jasové funkce v jednotlivých směrech obrazu. Abychom byli schopni porovnat odchylku složek  $u$  a  $v$  od uvažované nulové hodnoty, je nutné vypočítat následující integrál, vztahující se k celkové ploše obrazu (viz [7])

$$C = \int_0^N \int_0^M \left( \frac{\delta E}{\delta x} u + \frac{\delta E}{\delta y} v + \frac{\delta E}{\delta t} \right)^2 dx dy. \quad (2.7)$$

Ideální by bylo, kdyby hodnota parametru  $C$  a tedy celková chyba byla rovna nule. V praxi však tohoto ideálního stavu nelze vlivem určité kvantizační chyby a šumu dosáhnout. Zavádí se proto tzv. váhovací parametr označovaný  $\lambda$ . Celková chyba v čase se měnícím obrazu, je tedy dána součtem uvedených rovnic (viz [7])

$$\xi = S + \lambda C, \quad (2.8)$$

přičemž parametrem  $\lambda$  lze ovlivňovat celkovou hodnotu  $C$ . V praktických příkladech se hodnota tohoto parametru stanovuje experimentálně vzhledem ke každému konkrétnímu případu.

V posledním kroku této metody dochází k nalezení minimální hodnoty parametru  $\xi$ . Využívá se k tomu variačního počtu, přičemž obdržíme dvě rovnice o dvou neznámých  $u$  a  $v$ . Připomeňme, že se jedná se o námi hledané složky rychlosti pohybu (viz [7]).

$$\begin{aligned} E_x^2 u + E_x E_y v &= \lambda \nabla^2 u - E_x E_t, \\ E_x E_y u + E_y^2 v &= \lambda \nabla^2 v - E_y E_t. \end{aligned} \quad (2.9)$$

### 2.2.2 Metoda Lucas&Kanade

Jedná se o další diferenční metodu pro získání Optical Flow. Vycházíme zde z prostorových a časových parciálních derivací, přičemž stále platí aperturní omezení (viz úvahy výše). Mějme tedy opět rovnice  $E_x u + E_y v + E_t = 0$ . Základním předpokladem je, že jednotlivé parciální derivace jasové funkce obrazu budou konstantní v určitém malém okně a z toho plyne, že budou v okolí pixelu konstantní i rychlostní složky  $u$  a  $v$ . Velikost okna je dána hodnotou  $m \times m$ , přičemž musí být splněna podmínka, že  $m > 1$ . Počet prvků v daném okně je možné stanovit z rovnice  $m = n^2$ . Pro obecný počet pixelů v definovaném okně, je možné zapsat následující soustavou rovnic (viz [7]).

$$\begin{aligned} E_{x1}u + E_{y1}v &= -E_{t1} \\ E_{x2}u + E_{y2}v &= -E_{t2} \\ E_{x3}u + E_{y3}v &= -E_{t3} \\ &\vdots \\ E_{xn}u + E_{yn}v &= -E_{tn} \end{aligned} \quad (2.10)$$

V případě, že budeme mít okno o velikosti  $4 \times 4$ , získáme celkem 16 takovýchto rovnic. Tyto rovnice je možno zapsat v maticovém tvaru jako (viz [7])

$$\begin{bmatrix} E_{x1} & E_{y1} \\ E_{x2} & E_{y2} \\ \vdots & \vdots \\ E_{xn} & E_{yn} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} E_{t1} \\ E_{t2} \\ \vdots \\ E_{tn} \end{bmatrix} \quad (2.11)$$

tedy, obecně (viz [7])

$$Av = -b. \quad (2.12)$$

Je třeba uvést, že řešení vede k minimalizaci výrazu. Pomocí metody nejmenších čtverců se snažíme nalézt minimální hodnotu součtu druhých mocnin pro jednotlivé výsledky stanovených rovnic. Minimalizujeme tedy následující výraz (viz [7])

$$\sum_{i=1}^n (E_{xi}u + E_{yi}v + E_{ti})^2. \quad (2.13)$$

Dále je opět možné vyjít z obecné matematické formulace metody nejmenších čtverců (viz [7])

$$A^T Av = A^T (-b), \text{ osamostatníme-li } v \text{ pak } v = (A^T A)^{-1} A^T (-b) \quad (2.14)$$

S pomocí této obecné formulace stanovíme následující maticový zápis (viz [7])

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum E_{xi}^2 & \sum E_{xi}E_{yi} \\ \sum E_{xi}E_{yi} & \sum E_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum E_{xi}E_{ti} \\ \sum E_{yi}E_{ti} \end{bmatrix} \quad (2.15)$$

Jak již bylo uvedeno výše, cílem této metody je získat Optical Flow na základě derivací obrazové funkce ve všech směrech. Výsledkem této techniky je ovšem řidší pole vektorů pohybu. Čím více jsou plochy v obraze jasově homogenní, tím méně vektorů obdržíme.

### 2.2.3 Metoda Nagel

Jedná se o metodu pro výpočet Optical Flow, která vychází z derivací druhého řádu. Základem je stejná myšlenka, jako u předchozí techniky. Nagel vycházel z předpokladu orientovaného hladkého omezení, přičemž hladkost není předepsána napříč intenzitním gradientem. Tato technika se snaží nalézt minimální hodnotu následující funkce (viz [7])

$$\iint (\nabla I^T v + I_t)^2 + \frac{\alpha^2}{\|\nabla I\|^2 + 2\delta} \left[ (u_x I_y - u_y I_x)^2 + (v_x I_y - v_y I_x)^2 + \delta (u_x^2 + u_y^2 + v_x^2 + v_y^2) \right] dx dy. \quad (2.16)$$

Parametr  $\alpha$  omezuje proměnnost toku  $\nabla v$  v kolmém směru vzhledem ke gradientu. Konstantou  $\delta$  je možno ovlivňovat početní stabilitu procesu. Pro řešení je možno použít např. Gauss-Seidelovu iterační metodu (viz [7])

$$u^{k+1} = \xi(u^k) - \frac{I_x(I_x \xi(u^k) + I_y \xi(v^k) + I_t)}{I_x^2 + I_y^2 + \alpha^2} \quad (2.17)$$

$$v^{k+1} = \xi(v^k) - \frac{I_y(I_x \xi(u^k) + I_y \xi(v^k) + I_t)}{I_x^2 + I_y^2 + \alpha^2} \quad (2.18)$$

kde  $k$  značí počet iterací, přičemž  $\xi(v^k)$  a  $\xi(u^k)$  získáme z následujících vztahů (viz [7])

$$\begin{aligned} \xi(u^k) &= u^{-k} - 2I_x I_y u_{xy} - q^T (\nabla u^k) \\ \xi(v^k) &= v^{-k} - 2I_x I_y v_{xy} - q^T (\nabla v^k) \end{aligned} \quad (2.19)$$

$u_{xy}^k$  a  $v_{xy}^k$  označují parciální derivace  $v^k$ ,  $v^{-k}$  a  $u^{-k}$  jsou průměry lokálních sousedních bodů  $v^k$  a  $u^k$ .

Parametr  $q$  obdržíme ze vztahu (viz [7])

$$q = \frac{1}{I_x^2 + I_y^2 + 2\delta} \nabla I^T \left[ \begin{pmatrix} I_{yy} & -I_{xy} \\ -I_{xy} & I_{xx} \end{pmatrix} + 2 \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} W \right] \quad (2.20)$$

Maticí  $W$  rozumíme váhovací matici, která má následující tvar (viz [7])

$$W = (I_x^2 + I_y^2 + 2\delta)^{-1} \begin{pmatrix} I_y^2 + \delta & -I_x I_y \\ -I_x I_y & I_x^2 + \delta \end{pmatrix} \quad (2.21)$$

## 2.3 Algoritmus „Demon“

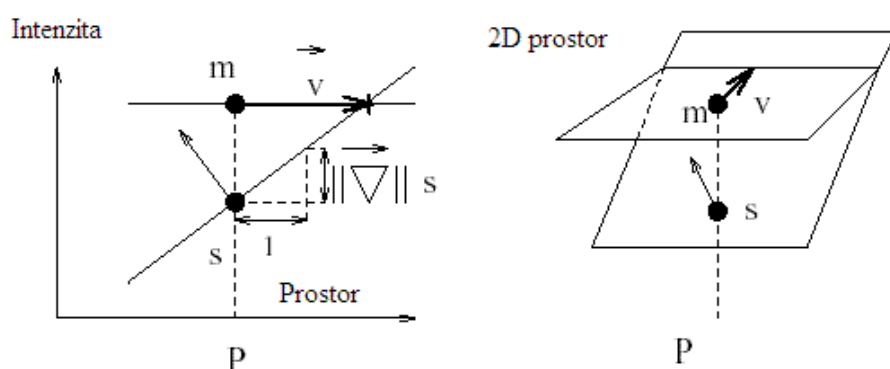
V této části bude popsána perspektivní a relativně mladá metoda pro registraci obrazů, která je porovnatelná s fluidní registrací. Tuto metodu představil J. P. Thirion v roce 1996 a je založena na pohybu pixelů na hranách objektů. Od té doby byla tato technika zdokonalována a dala podnět mnoha studiím (např. [12][14][20][21][22] aj.). Jedná se o vysoce přesnou metodu monomodální registrace obrazů. V další části této diplomové práce budou představeny a zhodnoceny výsledky registrace pomocí tohoto algoritmu.

Jak bude ukázáno dále, koncept této techniky úzce souvisí s Optical Flow. Základem techniky je předpoklad, že se hrany objektů v jednom z registrovaných obrazů chovají jako semipermeabilní membrány. Druhý obraz se chová jako model s deformovanou mřížkou. Cílem je provést difúzi mezi těmito obrazy, která je uskutečňována tzv. „vnitřními silami“, které jsou „umístěny“ na hranách objektů. Tato technika má úzkou spojitost s tzv. „Maxwellovými démony“ a vychází z konceptu termodynamiky. Můžeme říci, že tento algoritmus je v podstatě mezikrokem při určování Optical Flow, a že je založen na určitém přitažení objektů v obrazech. Existuje i řada dalších algoritmů (např. „Snake“), které jsou však početně náročnější.

V případě, že porovnáme rychlost algoritmů založených na bodech („landmarks“) v obrazech s algoritmem „Demon“ můžeme dle provedených studií říci, že výpočet algoritmů založených na bodech bývá rychlejší [2]. Samotná doba průběhu algoritmu však nemusí být vždy relevantním údajem. Výsledná doba potřebná pro přesné definování bodů („landmarků“) může být stejná jako doba uspořádaná. Nehledě na to, že nepřesným vložení bodů může vznikat výrazná registrační chyba.

V následujících odstavcích bude diskutováno spojení algoritmu „Demon“ s výpočtem Optical Flow a bude představen koncept difusních modelů.

Mějme dva porovnávané obrazy. Obraz A, který je statický a B, který je pohybový. Danému bodu P bude v obraze A odpovídat intenzitní hodnota  $s$  a v obraze B intenzitní hodnota  $m$ .



Obrázek 17. – Okamžitá rychlost v obrazech ve statickém a pohybovém obraze (převzato a upraveno z [11])

Opět vyjděme z uvedeného předpokladu pro výpočet Optical Flow, které říká že intenzita jednotlivých bodů v pohybujících se obrazech je konstantní.

V případě, že jsou rozdíly v porovnávaných obrazech velmi malé je možné pro výpočet Optical Flow využít následující vzorec (viz[11])

$$\vec{v} \cdot \vec{\nabla} s = m - s . \quad (2.22)$$

Z tohoto základního vzorce však není možné vypočítat  $\vec{v}$  (viz výše).  $\vec{v}$  je považováno za rychlost u obrazů („frames“), které se v čase mění. Při porovnávání statických obrazů mluvíme o stanovení posunu (nepřesnosti). Pro řádné stanovení  $\vec{v}$  se nabízí dvě varianty výpočtu. Jedna varianta je založena na stanovení lokálních hodnot  $\vec{v}$ . Druhá varianta předpokládá, že koncový bod  $\vec{v}$  je nejbližší bod plochy  $m$  v souřadném systému  $(x,y,z)$ . Tuto úvahu je možno vyjádřit následujícím způsobem (viz[11])

$$\vec{v} = \frac{(m-s)\vec{\nabla}s}{(\vec{\nabla}s)^2}, \text{ kde } \vec{\nabla}s \text{ je gradient obrazu A.} \quad (2.23)$$

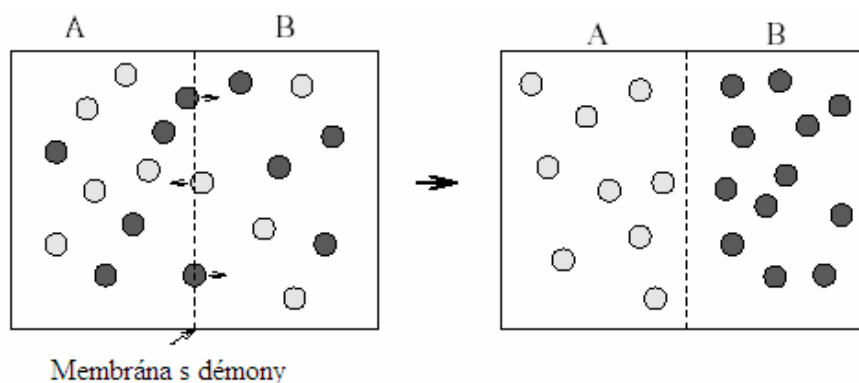
Tento předpoklad však selhává v případě malých hodnot gradientu v obraze A ( $\vec{\nabla}s$ ). Rozšíříme-li předchozí výraz, bude algoritmus vůči tomuto omezení odolnější. Tvar vzorce pak bude následující (viz[11])

$$\vec{v} = \frac{(m-s)\vec{\nabla}s}{(\vec{\nabla}s)^2 + (m-s)^2}, \text{ kde } (m-s)^2 \text{ zajišťuje stabilitu výpočtu.} \quad (2.24)$$

### Difusní modely

Difusní modely, které zde budou představeny, jsou odvozeny od tzv. „Maxwellových demonů“. Tento koncept má počátek již v 19. století a je založen na paradoxu termodynamického zákona.

Uveďme nejprve základní předpoklad jednoduchého difúzního modelu, který bude prezentován na následujícím obrázku.



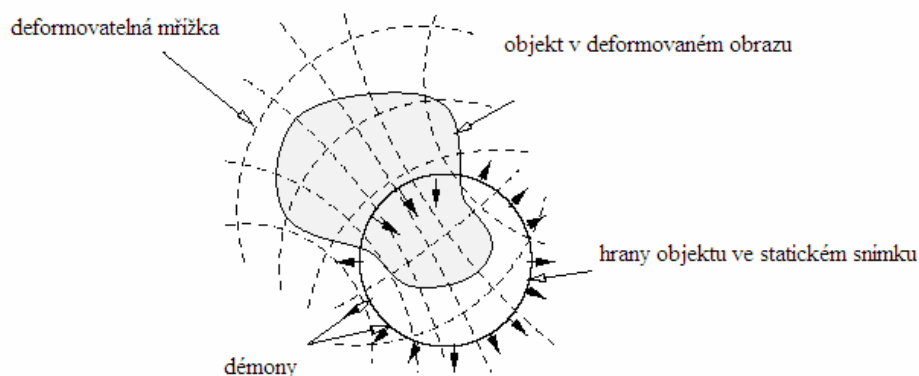
Obrázek 18. - Prvotní pokus s „Demony“ na semipermeabilní membráně propouštějící částice dvou různých plynů (převzato a upraveno z [11])

Mějme prostředí, ve kterém jsou dva různé plyny. První plyn obsahuje částice  $a$ , druhý plyn částice  $b$ . Jednotlivé částice jsou rozprostřeny v uvažovaném prostředí, ve kterém se uprostřed nachází polopropustná membrána. Tato membrána obsahuje soubor tzv. „Demonů“, který je schopen klasifikovat jednotlivé částice. Dále zajišťuje to, že částice  $a$  mohou difundovat pouze do prostředí A a částice  $b$  pouze do prostředí B. Dochází zde k paradoxu, neboť neplatí druhý zákon termodynamiky. Bylo zjištěno, že tento paradox je způsoben právě těmito

„Demony“, které vytváří vyšší entropii potřebnou k identifikaci jednotlivých částic. Výsledkem je tedy zvýšení entropie systému.

#### Praktické využití „Demon“ algoritmu

V případě, že chceme využít algoritmus „Demon“ pro registraci pohybového a statického obrazu, musí si dané obrazy co nejvíce odpovídat. Na následujícím obrázku bude naznačeno uspořádání „Demonů“. Předpokládejme, že se ve statickém obraze nachází objekt O a že jsou na jeho hranách rozprostřeni „Demony“.

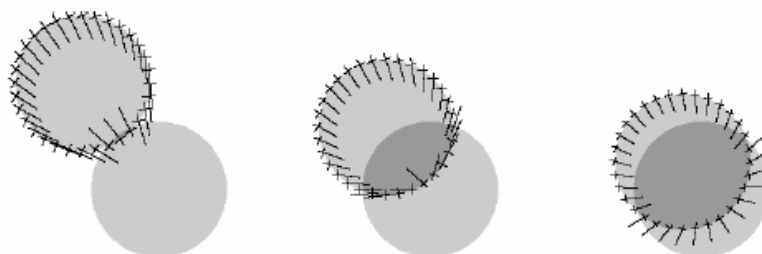


Obrázek 19. - Základní předpoklad difúzního modelu založeného na „demonech“ (převzato a upraveno z [11])

Pro každý bod, ze kterého je složena hrana objektu O ve statickém obraze, je možné lokálně určit vektor, který je k uvažovanému bodu kolmý. Tento vektor směřuje ven z objektu O a je možné jej určit na základě výpočtu gradientu ve statickém obraze. Na pohybový obraz je možné nahlížet jako na deformovanou mřížku. Vrcholy této deformované mřížky mohou být označeny jako „vnitřní“ nebo „vnější“ části. Od jednotlivých vztahů mezi „vnitřními“ a „vnějšími“ částmi je odvozena rigidita deformované mřížky pohybového obrazu. Definujeme tedy pohybový obraz jako difúzní model.

Pokusme se nyní zformulovat definice pro námi uvažované „demony“. Jedná se tedy o ovlivňující částice, umístěné v bodech, které jsou na hraně daného objektu. Difúzní model (pohybový obraz) je vtahován dovnitř objektu O, jestliže je odpovídající bod v pohybovém obraze označen jako „vnitřní“ a naopak je-li bod označen jako „vnější“ dochází k vytlačování z objektu O. Z této úvahy tedy plyne závěr, že pro difúzní modely je důležitá polarita částí obrazu.

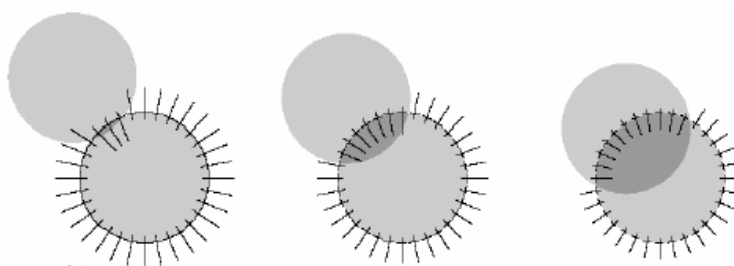
Podívejme se nyní podrobněji na vlastnosti a příklady difúzních modelů. Uvažujme rigidní transformaci mezi dvěma vzájemně posunutými disky (viz obrázek)



Obrázek 20. – Model přitažení mezi disky – 3 iterace (převzato z [11])

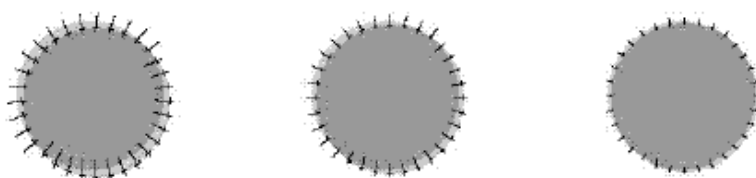
Na předchozím obrázku uvažujeme tři iterační kroky metody založené na přitažení. Pro tuto metodu je typické pravidelné rozprostření vzorků umístěných po obvodu (tedy na hraně) objektu. Cílem je nalézt nejbližší odpovídající si body. Velikost přitažení mezi disky je v každé iteraci konstantní. V případě konvergence dochází k poklesu velikosti přitažení. Výhodou oproti difúznímu modelu je, že metoda přitažení pracuje, i když se disky na začátku vůbec nepřekrývají. Je však náchylnější ke sklouznutí do lokálního minima.

Na následujícím obrázku jsou opět tři kroky iterační metody, tentokrát ovšem založené na principu difúzního modelu. U tohoto přístupu je důležitá polarita obrazu. Stejně jako u metody přitažení, dochází ke vzorkování hran, ovšem s tím rozdílem, že každý vzorek se stává „Demonem“. Vliv „Demonů“ může mít směr „vnitřní“ nebo „vnější“ a to podle odpovídajících bodů modelu. Pohyb disků je dán v každé iteraci všemi vzniklými faktory („forces“). V každé iteraci se do vyhodnocení pohybu zapojuje stále více „Demonů“.

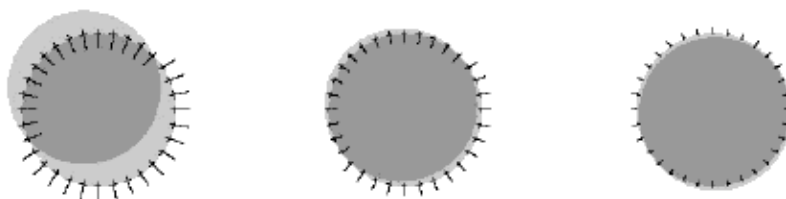


Obrázek 21. – Model difúze mezi disky – 3 iterace (převzato z [11])

Pro zajímavost uvedme následující tři iterace obou metod. Obě vedou k registraci disků. Ovšem jak bylo popsáno, každá trochu jiným způsobem.



Obrázek 22. – Model přitažení mezi disky – 6 iterací (převzato z [11])



Obrázek 23. – Model difúze mezi disky – 6 iterací (převzato z [11])

Při stanovování Optical Flow se setkáváme se dvěma přístupy. Jedná se o Optical Flow chovající se jako difúzní model a Optical Flow chovající se jako deformovaný model založený na přitažení. Uvedme ve stručnosti pár poznámek ke každému přístupu.

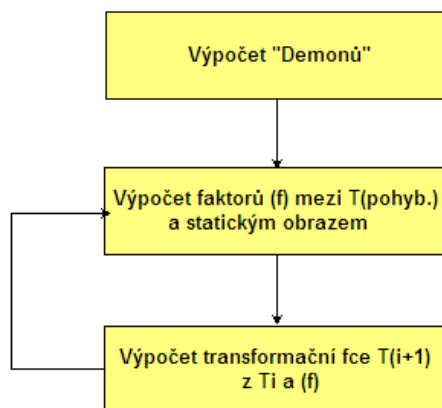


Optical Flow chovající se jako difúzní model - V případě, že využíváme tuto techniku, je nutné určit, které body pohybového obrazu budou „vnitřní“ a které „vnější“ (viz výše). Toto rozdělení můžeme provést následujícím způsobem. Pro každý bod  $P$  ve statickém obrazu, jehož gradient se nerovná nule, je kontura  $s=I$ .  $I$  je rovno intenzitě a dále platí, že  $I=s(P)$  je konstantní. Na hraně si můžeme představit rozhraní mezi vnitřní a vnější oblastí. Je-li  $s < I$  jedná se o vnitřní oblast, je-li  $s > I$  jedná se o oblast vnější. Uvažované oblasti je možné rozdělit na „vnitřní“ a „vnější“ automaticky a to porovnáním intenzitních hodnot pohybového obrazu s intenzitou  $I$ . Během jednoho iteračního kroku dojde k porovnání proměnné  $\vec{v}$  se základním vlivem („elementary force“)  $\vec{f}$ . Směr  $\vec{f}$  je odvozen od gradientu statického obrazu a orientace je dána poměry mezi  $m$  a  $I$ .

Optical flow chovající se jako deformovaný model založený na přitažení – Vycházíme-li z informací o souřadnicích bodu  $P$  a jeho gradientu, můžeme předpokládat, že hledaný bod je dán vztahem  $P'=P + \vec{v}$ , a že má v obou obrazech stejnou intenzitu. Předpoklad techniky přitažení tedy hledá bod  $P'$  v pohybovém obrazu, který dle výše zmíněných omezení nejvíce odpovídá bodu  $P$  ve statickém obrazu.

V následujícím odstavci bude popsáno praktické řešení difúzních modelů, konkrétně se bude jednat o iterační postup. Postup se dělí na dvě základní části, a to pro stanovení „Demonů“ a část iterační. Celý postup může být ovlivněn volbou rozložení „Demonů“, uvažovanou transformací (affiní, rigidní apod.), formou výpočtu faktoru  $\vec{f}$  nebo vhodnou interpolační metodou.

Pro registraci obrazů, která je založena na difúzním modelu, se využívá klasický iterační postup (nebo i postup automatický). Cílem tohoto procesu je nalézt výslednou transformační funkci  $T$  mezi statickým a pohybovým obrazem. Výsledná funkce je dána kombinací všech možných dílčích transformačních funkcí, tedy  $T \subset \{T_0, T_1, T_2, T_3, \dots, T_i\}$ . V každém iteračním kroku se transformační funkce inkrementuje o jedničku ( $T_{i+1}$ ). Tato funkce je omezena „vnitřními“ a „vnějšími“ faktory („forces“). „Vnitřní“ vznikají na základě vztahů mezi jednotlivými body difúzního modelu a „vnější“ vznikají vzájemným vztahem transformační funkce  $T$  se statickým obrazem. Základní iterační postup je znázorněn na následujícím obrázku.



Obrázek 24. – Základní iterační schéma registraci obrazů na základě difúzních modelů (převzato a upraveno z [11])

Blok s názvem *Výpočet „Demonů“* vypočítává, jak je již z jeho názvu patrné, soubor „Demonů“ statického obrazu. Za soubor „Demonů“  $D_s$  můžeme považovat celou mřížku obrazu nebo pouze její určitou část. V případě celé mřížky plní každý pixel funkci „Demonu“. Naopak, uvažujeme-li pouze určitou část, např. získanou pomocí hranových operátorů, omezuje se soubor  $D_s$  pouze na tuto konkrétní část. Každému „Demonu“ je přiřazena odpovídající informace např. o jeho pozici, rozhraní, směru apod.

Iterační část blokového schématu slouží, jak již bylo uvedeno výše, ke stanovení transformační funkce  $T$ . Na začátku uvažujeme počáteční funkci  $T_0$ . V každém iteračním kroku se pro jednotlivé „Demony“ obsažené v souboru  $D_s$  vypočte základní faktor („force“)  $\vec{f}_i$ , jehož hodnota závisí na polaritě a směru. Z určené transformační funkce  $T_i$  a všech  $\vec{f}_i$  se vypočte transformační funkce následujícího kroku, tedy  $T_{i+1}$ .

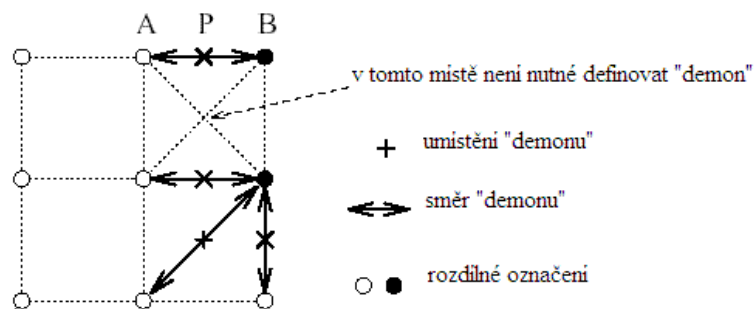
Pro praktické využití můžeme do iteračního procesu implementovat tři základní varianty rozložení „Demonů“. Jedná se o „Demony“ rozložené na celé mřížce obrazu, pouze na hranách nebo pouze v určitých segmentech obrazu. Není možné říci, která z variant je nejlepší. Je nutné vycházet z předpokladů řešeného problému a individuálního přístupu k porovnávání a registraci obrazů. Uvedme si nyní pár základních vlastností jednotlivých variant rozložení [dle 11].

*Rozložení „demonů“ na celé obrazové mřížce* – Za „Demonu“ je považován každý bod obrazové mřížky, jehož gradient ve statickém obrazu není roven nule. Pro každý „Demon“ stanovujeme základní rozdíl  $\vec{d}$ . Pomocí Gaussovského filtru vytvoříme pravidelné rozdílové pole. Tuto filtraci provádíme v každé iteraci zvlášť. Vzniklé vyhlazené pole poté musíme vhodně interpolovat. Pomocí Optical Flow (rovnice 2.24) získáme faktor („force“) „demonu“. K tomu využíváme pouze informace o souřadnicích každého bodu a jeho gradientu. Výsledkem je aplikace stanoveného faktoru („force“) během daného iteračního kroku, jedná se tedy o rozdíl, kde  $\vec{d} = \vec{v}$ . Zde je patrný jednoduchý příklad propojení Optical Flow a difúzního procesu, který v praxi dává uspokojivé výsledky (viz např. [12]). Tato technika může být v jistém ohledu příliš pomalá. Jejího zrychlení můžeme dosáhnout pomocí pyramidového přístupu. Jedná se o využití hrubšího měřítka během prvních iterací. Se stoupajícím počtem iterací se měřítka zlepšuje. V literatuře se tento přístup uvádí jako „corse – to – fine“ [12].

*Rozložení „demonů“ na hranách* – Z názvu této techniky vyplývá, že jsou „Demony“ rozmístěny na hranách objektů, vyskytujících se v obrazech. Pro získání souboru „Demonu“  $D_s$  se využívají hranové operátory (např. Canny, Sobell apod.). Poté se metodou nejmenších čtverců určí globální transformační funkce  $T$  a provede se vhodná interpolace. Tato technika bývá rychlejší než při rozložení „Demonů“ na celou obrazovou mřížku, avšak stanovení „Demonů“ již není primárně založeno na Optical Flow [11]. Pro body na hranách statického obrazu se stanoví intenzitní hodnota a orientace normály (dovnitř nebo ven z objektu). Jednotlivé body tvoří rozhraní mezi oblastmi. Tato technika již využívá pravděpodobnost rozložení pomocí K-funkce. Toto rozložení slouží k pravděpodobnostnímu stanovení, zda bude daný bod na hraně objektu považován za „vnitřní“ nebo „vnější“.

*Využití souboru „demonů“ na segmentované obrazy* – Poslední uvažovaná technika pracuje se segmentovanými obrazy a její využití může být výhodné pro registraci a porovnávání obrazů zejména např. snímků mozku [14]. Statický i pohybový obraz je rozdělen do vhodných

segmentů. Cílem je nalézt odpovídající si body v segmentovaných strukturách porovnávaných obrazů. Každý bod obsahuje popis vztahující se k uvažované struktuře. Nejprve dojde k definování bodů statického obrazu, které vytvoří soubor „Demonů“  $D_s$ . Následně opět hledáme transformační funkci  $T$ . Krok interpolace se v tomto případě vynechává. Orientace jednotlivých bodů je dána jejich označením (viz obrázek níže). Uvedme nyní chování „Demonů“ v této technice. Předpokládejme dva sousedící voxely, které jsou spojeny. Toto spojení ilustruje následující obrázek



Obrázek 25. – Chování „demonů“ při technice segmentace obrazů (převzato a upraveno z [11])

Body A a B představují sousední voxely. Jsou označeny  $s_{in} = s(A)$  a  $s_{out} = s(B)$ . Bod P označuje „Demon“ na pozici  $(A+B)/2$ . Faktor „Demonu“ („force“)  $\vec{f}(P)$  je určen hodnotami  $s_{in} = s(A)$  a  $s_{out} = s(B)$ . Výhoda metody spočívá v interakci s jednotlivými strukturami.

V následujícím odstavci bude ukázáno, jakým způsobem je možné účinně zrychlit celý „Demon“ algoritmus. Toto zrychlení se děje na základě přidání jistého aktivního faktoru („active force“) do základního vzorce. Dle [12] vede přidání tohoto faktoru až ke 40% zrychlení oproti základnímu přístupu a vzniká tak větší odolnost vůči základnímu omezení, které uvádí, že porovnávané obrazy by si měly co nejvíce odpovídat. Snahou je co nejefektivnější využití „Demonů“ ve statickém a nově i v pohybovém obraze. Základní myšlenka tohoto zrychlení je využití i gradientu pohybového obrazu. Tedy nevytvářet pouze faktory („forces“), které umožňují deformovanému objektu difundovat do odpovídajícího statického objektu, ale vytvořit i faktory, které umožňují statickému objektu difundovat do objektu pohybového.

Zformulujme nyní vztah, který bude odpovídat výše zmíněné úvaze. Základní vzorec (2.24) označme jako „pasivní faktor“ („passive force“)  $\vec{f}_s$ . Z pohybového obrazu nyní vytvořme „aktivní faktor“ („active force“)  $\vec{f}_m$  (viz[12])

$$\vec{f}_m = -\frac{(s-m)\vec{\nabla} m}{(\vec{\nabla} m)^2 + (s-m)^2}. \quad (2.25)$$

„Pasivním“ faktorem je myšleno získání gradientní informace pouze ze statického obrazu. Naopak faktorem „aktivním“ je myšleno získání gradientní informace z obrazu pohybového, přičemž výraz „aktivní“ by měl vyjadřovat jistou vyšší aktivitu, neboť se tento faktor vypočítává v každém iteračním kroku.

Kombinací „pasivního“ a „aktivního“ faktoru získáme rozšířený, zrychlující vztah (viz[12])

$$\vec{f} = \vec{f}_s + \vec{f}_m = (m-s) \left( \frac{\vec{\nabla} s}{|\vec{\nabla} s|^2 + (s-m)^2} + \frac{\vec{\nabla} m}{|\vec{\nabla} m|^2 + (s-m)^2} \right). \quad (2.26)$$

V případě, že pracujeme s obrazy ze zobrazovací modality CT, je pro registraci a porovnání vhodné využít následující vztah, který zavádí jistý normalizační faktor  $\alpha$  (viz[12]).

$$\vec{u} = (m-s) \left( \frac{\vec{\nabla} s}{|\vec{\nabla} s|^2 + \alpha^2 (s-m)^2} + \frac{\vec{\nabla} m}{|\vec{\nabla} m|^2 + \alpha^2 (s-m)^2} \right) \quad (2.27)$$

Jsou-li porovnávané obrazy značně rozdílné, je možné za normalizační parametr zvolit malé hodnoty. V případě, že v dalších iteračních krocích dochází ke konvergenci algoritmu, je možné hodnotu tohoto parametru zvýšit.

V případě vhodně zvolených parametrů Gaussovského filtru a parametru  $\alpha$  je možné při registraci obrazů získat velmi dobré výsledky (viz např. studie [13]).

V závěru této kapitoly provedme malé shrnutí vztahující se k technice „Demon“. Jedná se o přístup pro porovnávání a registraci pohybového a statického obrazu. Technika vychází z difúzních modelů a je založena na informacích o polaritě v obrazech. O difúzi mluvíme z toho důvodu, že prostřednictvím hran objektů v jednom obrazu, které se chovají jako polopropustné membrány, dochází k difúzi do druhého obrazu. V teoretické úvaze výše bylo ukázáno, že metoda Optical Flow je jedním z kroků registrace obrazů a dále, že je Optical Flow určité pomyslné rozhraní mezi klasickou porovnávací metodou založenou na přitažení a difúzním přístupem. V praktické části této diplomové práce budou prezentovány výsledky registrace obrazů z CT zobrazovací modality. Pro praktickou část byl zvolen přístup rozložení „Demonů“ po celé obrazové mřížce.

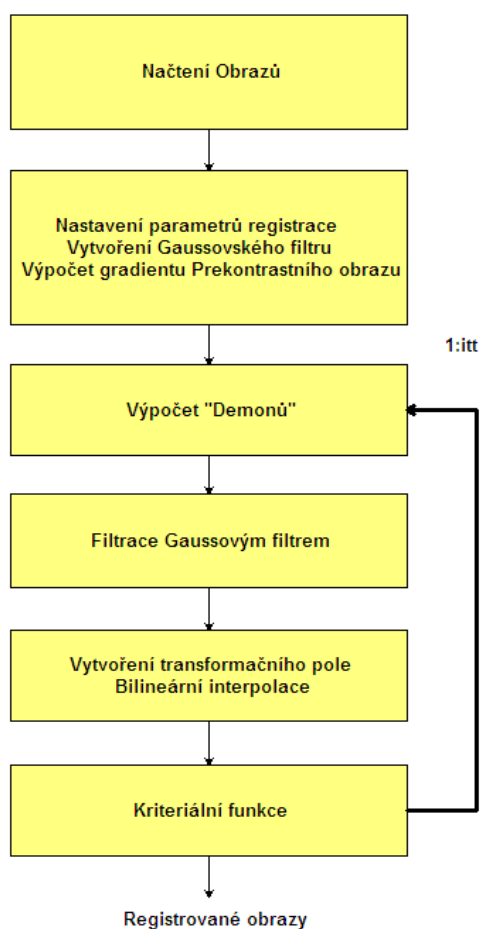
### 3. Programové řešení

V této kapitole bude představeno zvolené řešení praktické části diplomové práce. Nejprve budou prezentována hrubá bloková schémata realizovaných registračních postupů, poté bude prezentován kód vytvořený v programovém prostředí Matlab a v poslední části dojde k prezentaci řešení grafického rozhraní (GUI), opět vytvořeného v programovém prostředí Matlab.

#### 3.1 Bloková schémata realizovaných registračních postupů

Pro praktické ověření zadání této diplomové práce, byl po konzultaci s vedoucím práce vybrán algoritmus „Demon“. Dodané testované obrazy byly získány ze zobrazovací modality CT. Jedná se o dva soubory dat. Jeden obsahuje 22 prekontrastních a 22 odpovídajících postkontrastních obrazů. Druhý soubor dat obsahuje 37 prekontrastních a 37 odpovídajících postkontrastních obrazů.

Na následujícím obrázku je znázorněno hrubé blokové schéma iterační formy registrace obrazů pomocí algoritmu „Demon“.



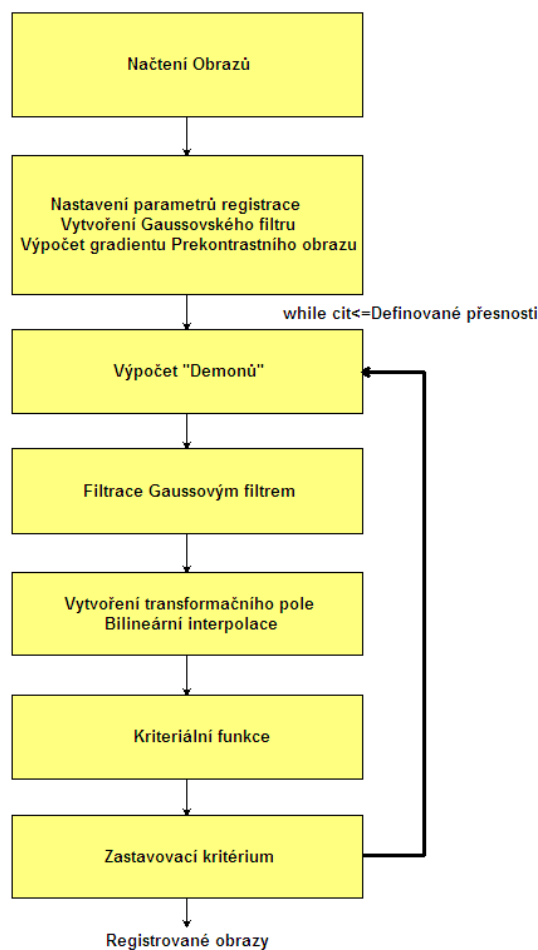
Obrázek 26. – Blokové schéma iterační registrace obrazů realizované v programovém prostředí Matlab

V první části dochází k načtení prekontrastního a postkontrastního obrazu. V dalším kroku dochází, za předpokladu využití rozšířeného vzorce, k definování normalizačního parametru  $\alpha$ . Jak bude ukázáno později, bylo během hodnocení výsledků odzkoušeno několik hodnot parametru  $\alpha$ . Poté se vytvoří vhodný Gaussovský filtr a vypočte se gradient prekontrastního obrazu.

Další částí tohoto blokového schématu je část iterační. Zde dochází k samotnému výpočtu „Demonů“ dle zvoleného vzorce. V případě, že zvolíme rozšířenou formu vzorce pro výpočet „Demonů“, dochází i ke stanovení gradientu postkontrastního obrazu. Vypočtené transformační pole se filtruje vytvořeným Gaussovským filtrem a interpoluje do původní obrazové mřížky pomocí bilineární interpolace. V posledním bloku smyčky dochází k výpočtu kritériální funkce Mutual Information a k výpočtu globálních podobností obrazů COR, NCC, COS, SSD.

Kritériální funkce Mutual Information se primárně využívá k porovnávání podobnosti obrazů, které byly získány různými zobrazovacími modalitami. V našem případě se jedná o porovnávání obrazů získaných z jedné zobrazovací modality (CT). Na obrazy je však nahlíženo jako na multimodální a to z toho důvodu, že je do zobrazované scény dodán kontrast.

Základní iterační schéma pro registraci obrazů bylo přeprogramováno do automatické formy, jejíž hrubé blokové schéma je uvedeno na následujícím obrázku.



Obrázek 27. – Blokové schéma automatické registrace obrazů realizované v programovém prostředí Matlab

### 3.2 Struktura programu

V této části bude prezentován vytvořený kód, vztahující se jak k iteračnímu, tak i k automatickému postupu registrace obrazů.

Načtení obrazů z CT zobrazovací modality, které byly dodány vedoucím diplomové práce.

```
%% Nacteni Pre_kontrastniho & Post_kontrastniho obrazu

load ('1nohy3DPre.mat'); % Nacteni Pre_kontrastniho obrazu
Pre_img = zeros (512,512);
Pre_img(1:512,1:512) = nohy3DPre(1:512,1:512);

load ('1nohy3DPost.mat'); % Nacteni Post_kontrastniho obrazu
Post_img = zeros (512,512);
Post_img(1:512,1:512) = nohy3DPost(1:512,1:512); %I2
```

Obrázek 28. – Kód pro načtení obrazů realizovaný v programovém prostředí Matlab

Na následujícím obrázku je uvedena část kódu vztahující se k nastavení parametrů registrace obrazů. Gaussovský filtr je vytvořen pomocí funkce *fspecial*, která obsahuje předdefinované typy filtrů. Velikost jádra je v tomto případě 60 x 60 a parametr sigma má hodnotu 10. Po vytvoření transformačního pole se vypočte gradient postkontrastního obrazu.

```
%% Nastaveni parametru algoritmu "Demon"

Gauss_filt = fspecial('gaussian',[60 60],10); % Vytvoreni 2D Gausovskeho filtru (k filtraci "rozostreni" souradneho pole)

Tx=zeros(size(Pre_img)); % Vytvoreni transformačního pole ve smeru x
Ty=zeros(size(Pre_img)); % Vytvoreni transformačního pole ve smeru y

[Sy,Sx] = gradient(Post_img); % Vypocet gradientu Post_kontrastniho obrazu

itt=100;

%% Pomocne promenne registrace obrazu

M=Pre_img; % Pomocna promenna

m=1; %Pomocna promenna
```

Obrázek 29. – Kód pro nastavení parametrů algoritmu při iteračním i automatickém schématu pro registraci obrazů realizovaném v programovém prostředí Matlab

### 3.2.1 Struktura programu iteračního registračního postupu

V této části práce bude prezentován vytvořený kód v programovém prostředí Matlab pro iterační metodu registrace obrazů.

```
%% Smyčka iteracní registrace dvou obrazu

for i=1:itt

    CC(m) = corr2(M,Post_img); % Vypocet hodnoty Normovaného korelačního koeficientu

    Iroz = M - Post_img; % Odectení Post_kontrastního a Pre_kontrastního obrazu

    % Zakladni (puvodni) vzorec (pouze s vyuzitim tzv."Internal force")

    if Mode==0;

        Ux = -(Iroz.*Sx)/((Sx.^2+Sx.^2)+Iroz.^2);
        Uy = -(Iroz.*Sy)/((Sx.^2+Sx.^2)+Iroz.^2);

    end

    if Mode==1;

        [My,Mx] = gradient(M);

        % Rozsireny vzorec - uvazujeme tzv "external force & internal force", ve vypoctu se projevi
        % hodnota konstanty alpha - její fce je mira "rozostrení" na hranach

        Ux = -Iroz.*((Sx./((Sx.^2+Sx.^2)+alpha^2*Iroz.^2)))+(Mx./((Mx.^2+My.^2)+alpha^2*Iroz.^2));

        Uy = -Iroz.*((Sy./((Sx.^2+Sx.^2)+alpha^2*Iroz.^2)))+(My./((Mx.^2+My.^2)+alpha^2*Iroz.^2));

    end

    Ux(isnan(Ux))=0; % Osetrení delení nulou
    Uy(isnan(Uy))=0;

    Uxs = 3*imfilter(Ux,Gauss_filt); % Filtrace Gausovským filtrem - transformacní pole x
    Uys = 3*imfilter(Uy,Gauss_filt); % Filtrace Gausovským filtrem - transformacní pole y

    Tx = Tx + Uxs; % Vytvoreni noveho transformacního pole
    Ty = Ty + Uys;

    M=bilint2(Pre_img,Tx,Ty); % Bilineární transformace obrazu

    % __ Vypocet podobnostních (kriteriálních) funkcí __ %

    [IAB,IAB_norm,I_xy]=MutualInformation(M,Post_img); % Vypocet hodnot MutualInformation

    if IAB~=0; % Naplnení matice MI hodnotami IAB & IAB_norm

        MI(1,m)=IAB; MI(2,m)=IAB_norm; MI(3,m)=I_xy; % Naplnení matice MI hodnotami IAB & IAB_norm & I_xy

    end

    [COR,NCC,COS,SSD]=TemplateMatching(M,Post_img); % Vypocet globalních podobnostní obrazu

    if COR~=0;

        TM(1,m)=COR; TM(2,m)=NCC; TM(3,m)=COS; TM(4,m)=SSD; % Naplnení matice TM hodnotami COR, NCC, COS, SSD

    end

    m=m+1;

end
end
```

Obrázek 30. – Kód iterační smyčky registrace obrazů při použití algoritmu „Demon“ realizovaném v programovém prostředí Matlab



Při iteračním postupu je nejprve nutné definovat počet iterací pro *for* cyklus, ve kterém je umístěno hlavní tělo programu. Poté dojde k výpočtu počáteční podobnostní funkce registrovaných obrazů pomocí normovaného korelačního koeficientu (funkce *corr2*). Následně se vypočte rozdílový obrazu *Iroz*. Jedná se o rozdíl mezi prekontrastním a postkontrastním obrazem.

Součástí iterační smyčky je volba použitého vzorce. Na základě proměnné *Mode* je možné vybrat mezi základním (*Mode==0*) nebo rozšířeným (*Mode==1*) vzorcem pro výpočet „Demonů“. V případě, že zvolíme vzorec rozšířený, dojde k výpočtu gradientu prekontrastního obrazu pomocí funkce *gradient*. Po výpočtu následuje ošetření dělení nulou a filtrace vypočteného transformačního pole ve směru *x* i *y*. Filtrace je řešena pomocí funkce *imfilter*. Jak již bylo uvedeno v jednom z předchozích odstavců, jedná se o Gaussovský filtr s výše definovanými parametry. Po filtraci dojde k vytvoření nového transformačního pole *Tx* a *Ty*.

Následuje bilineární interpolace obrazu. Jedná se o kompromis mezi rychlostí výpočtu a přesností samotné interpolace. Nejprve se pomocí funkce *ndgrid* vytvoří interpolační mřížka a poté se vypočtou souřadnice ve směru *x* a *y*. Poté je provedeno zaokrouhlení hodnot transformačního pole a výpočet interpolačních konstant. Aby byla interpolace řádně provedena, je potřeba ošetřit indexy na hranách obrazové mřížky. V závěru je pomocí smyčky *for* proveden výpočet intenzitních hodnot všech pixelů a pomocí funkce *reshape* vytvořen výsledný, interpolovaný obraz.

Po provedení bilineární interpolace dochází k výpočtu kritériálních (podobnostních) funkcí. Nejprve k výpočtu sdruženého histogramu a Mutual Information a poté k výpočtu globálních podobnostních funkcí.

```
function [IAB,IAB_norm,I_xy]=MutualInformation(I1,I2)
    I1=im2uint8(I1);
    I2=im2uint8(I2);

    %% Vypocet sdruzeneho histogramu
    hist=zeros(size(I1)); % Vytvoreni nulove matice

    row=size(I1,1);      % Zjisteni poctu sloupcu
    col=size(I2,2);      % Zjisteni poctu radku

    % Vypocet sdruzeneho 2D histogramu
    for i=1:row;
        for j=1:col;

            hist(I1(i,j)+1,I2(i,j)+1)= hist(I1(i,j)+1,I2(i,j)+1)+1;

        end
    end

    %% Vypocet MI
    % Oznaceni promennych viz Jiri Jan str.458

    [row,col] = size(hist);
    norm= hist./(row*col); % Normalizace sdruzeneho histogramu
    Nl=sum(norm');        % Aproximace normalizovanim 1-tym radkem sdruzeneho histogramu
    Nm=sum(norm);          % Aproximace normalizovanim m-tym radkem sdruzeneho histogramu
```

```

for i=1:row;    % Prochazeni jednotlivymi radky
    if( Nl(i)==0 )
    else
        HA = HA+ -(Nl(i)*(log2(Nl(i)))); % Stanoveni castecne entropie prvnio obrazu
    end
end

HB=0;
for i=1:col;    % Prochazeni jednotlivymi sloupci
    if( Nm(i)==0 )
    else
        HB= HB+ -(Nm(i)*(log2(Nm(i)))); % Stanoveni castecne entropie prvnio obrazu
    end
end

I_xy = -sum(sum(norm.*(log2(norm+(norm==0))))); % Stanoveni sdruzene entropie

IAB = (HA + HB)/I_xy;    % Normovana Mutual information

IAB_norm = HA + HB - I_xy; % Mutual information

```

Obrázek 31. – Kód výpočtu sdruženého histogramu a Mutual Information realizovaný v programovém prostředí Matlab

Z kódu je patrné, že nejprve dochází k převedení na datový formát *unit8*. Poté následuje již samotný výpočet sdruženého histogramu, dle pravidel, která byla uvedena výše. Po vytvoření sdruženého histogramu se provede jeho normalizace. Poté dojde k výpočtu částečné entropie jednotlivých obrazů. Z vypočtených částečných entropií se již (dle výše uvedeného vzorce) vypočte entropie sdružená. Posledním krokem této funkce je výpočet Mutual Information a její normované verze.

Po výpočtu Mutual Information dochází k nalezení globálních podobnostních funkcí korelačního koeficientu, normované cross-correlation, kosinového kritéria a SSD.

### 3.2.2 Struktura programu automatického registračního postupu

V této části bude názorně ukázáno, jakým způsobem byl realizován navržený program pro automatickou registraci obrazů. Na následujícím obrázku budou znázorněny definované pomocné proměnné, použité při automatické registraci.

```

%% Pomocne promenne registrace obrazu

itt=1; % Pomocna promenna

n=1; % Pomocna promenna pri StopCrit

MAX=zeros(1,20); % Pomocny vektor pri StopCrit
Roz=zeros(1,20); % Pomocny vektor pri StopCrit

cit=0; % Vytvoreni citace pri StopCrit

CC=zeros(1,itt); % Pomocna matice pri vypoctu Corr2
MI=zeros(3,itt); % Pomocna matice pri vypoctu a zobrazeni prubehu MI
TM=zeros(4,itt); % Pomocna matice pri vypoctu a zobrazeni prubehu TemplateMatching

M=Pre_img; % Pomocna promenna

```

Obrázek 32. – Definování pomocných proměnných při automatické registraci obrazů v programovém prostředí Matlab

Funkce pomocných proměnných bude patrná z následujících obrázků. Při formě automatické registrace obrazů využíváme pro výpočet „Demonů“ a bilineární interpolace stejnou funkci jako u interpolační formy registrace. Využíváme však jiný druh smyčky a to *while*. Nejvýraznější změnu zavádí vytvoření zastavovacího kritéria, které umožňuje automatickou formu registrace obrazů.

Začátek smyčky je uveden na následujícím obrázku.

```

%% Smyčka automaticke registrace dvou obrazu

while cit<=RA % Podminkou volime presnost registrace - po 5-ti krocich stanovime
% z hodnoty MI_norm maximum.Po dalsich 5-ti krocich provedeme to
% stejne atd. Je-li rozdíl hodnot IAB_norm <0.05 citac se inkrementuje
% o 1. V pripade, ze by doslo k nahlemu zmenseni hodnoty podobnostni
% fce dojde take k inkremetaci citace a tedy k zastaveni registrace

```

Obrázek 33. – Začátek smyčky automatické registrace obrazů realizované v programovém prostředí Matlab

Proměnnou RA („Registration Accuracy“) volíme přesnost registrace. Jedná se v podstatě o čítač, do kterého se zapisují rozdílové hodnoty kritériální funkce Mutual Information. V případě, že je např. RA=5, dojde k automatickému zastavení registrace a zobrazení výsledků je-li *cit*=5 (viz dále). Další proměnnou, se kterou se zde setkáváme, je W („Window“). Jedná se o plovoucí okno, které vyhodnocuje hodnoty kritériální funkce Mutual Information.

```

%___ Zastavovací kritérium dle podobnostní (kriteriální) funkce ___%

if itt>=n*W      % Čekáme W cyklu na naplnění MI(2,itt)=IAB_norm;

    c_max=n*W;    % Počet prvků, které budou naplněny z MI(2,itt) do SC

    if MI(2,c_max)~=0

        SC=zeros(1,length(W));    % Vytvoření plovoucího okna

        SC=MI(2,:);    % Naplnění matice vektoru SC hodnotami MI

        if SC(1,c_max)~=0    % Stanovení max hodnoty plovoucího okna

            MAX(1,n)=max(SC);    % Stanovení maximální hodnoty z prohledávaného vektoru

        end

        if n>=2

            Roz(1,n-1)=MAX(1,n)-MAX(1,n-1);    % Výpočet odchylky a chyby registrace

        end

        n=n+1;

        if Roz(1,n-1)<=0.005 && Roz(1,n-1)~=0 || MAX(1,n)>MAX(1,n-1)

            cit=cit+1;
        end

        if cit>=RA

            break
        end

    continue
end
end

itt=itt+1;

end

```

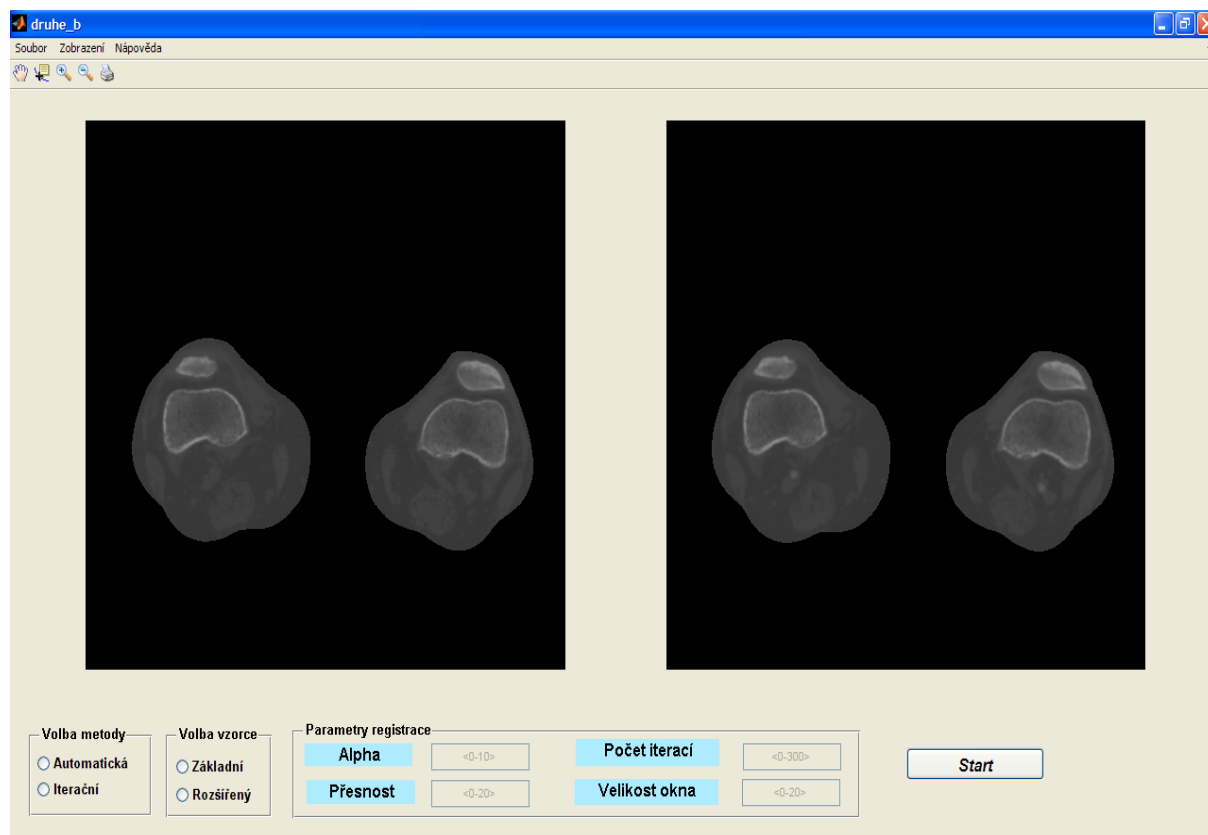
Obrázek 34. – Zastavovací kritérium automatické registrace obrazů realizované v programovém prostředí Matlab

Na začátku zastavovacího kritéria čekáme, dle zvolené hodnoty  $W$ , daný počet cyklů. Toto čekání je nutné z důvodu výpočtu hodnot kritériální funkce Mutual Information. Je-li např.  $W=4$  načtou se po čtyřech cyklech 4 hodnoty kritériální funkce MI, ze kterých se stanoví maximum. Poté se celý proces opakuje. Nyní máme dvě maximální hodnoty, které odečteme. Pokud je rozdíl menší než 0,005 dojde k inkrementaci čítače. Pokud čítač dosáhne požadované hodnoty RA dojde k přerušení registrace a zobrazení výsledků. Pokud by nastal opačný případ, kdy by maximální hodnota v dalším cyklu byla menší než předchozí, dojde také k inkrementaci čítače.

Touto formou řešíme stabilitu a přesnost registrace obrazů, tedy uvažujeme oba dva extrémy, které mohou nastat (hodnota kritériální funkce stoupá nebo naopak klesá).

### 3.2.3 Prezentace vytvořeného grafického uživatelského rozhraní GUI

V předchozí části této kapitoly byly popsány realizované části programu pro registraci obrazů. Nyní bude představeno řešení grafického uživatelského rozhraní, které bylo vytvořeno v programovém prostředí Matlab, a využívá již zmíněný kód programu pro registraci obrazů.



Obrázek 35. – Realizované grafické uživatelské rozhraní v programovém prostředí Matlab

Realizovaný program je možné využít pro registraci obrazů následujících formátů: \*.mat, \*.jpg, \*.jpeg a \*.png, přičemž je nutné, aby byla dvojice obrazů k registraci ve stejném formátu. Primárně byl program vytvořen pro provedení registrace obrazů prekontrastních a postkontrastních, získaných z CT zobrazovací modality (tato obrazová data byla dodána vedoucím diplomové práce). Z obrázku č. 35 můžeme vidět, že je grafické rozhraní rozděleno do tří částí. Jedná se o menu, zobrazovací část a část nastavení parametrů samotné registrace obrazů.

Pro provedení registrace je nutné načíst příslušná obrazová data ve vhodném formátu a zadat příslušné parametry registrace obrazů. Mezi základní parametry patří volba metody (automatická či iterační forma registrace) a vzorce (základního 2.26 či rozšířeného 2.27). Po volbě základních parametrů se zpřístupní pole, do kterých je nutné zadat odpovídající hodnoty parametrů registrace.

#### Rozsah a formát jednotlivých parametrů:

**Alpha** – je nutné zadat kladnou hodnotu v rozsahu <0,10> .

**Počet iterací** – v případě, že zvolíme iterační formu registrace obrazů, je nutné zadat celočíselnou hodnotu v rozsahu  $\langle 0, 300 \rangle$ .

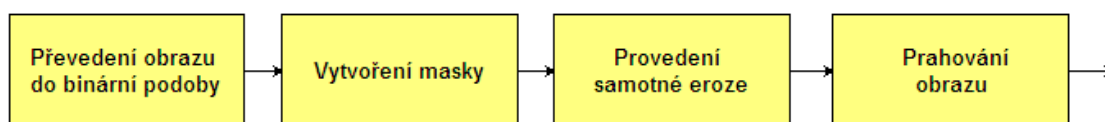
**Přesnost** – jedná se o parametr vztahující se k automatické formě registrace obrazů. Jak bylo uvedeno výše, jedná se v podstatě o čítač, který se inkrementuje v případě, že dosahujeme požadované úrovně přesnosti. Hodnoty, na základě kterých se čítač inkrementuje, jsou získány podobnostní kriteriální funkcí (v našem případě na základě Mutual Information). Aby program řádně fungoval, je nutné zadat celočíselnou hodnotu v rozsahu  $\langle 0, 20 \rangle$ , přičemž zvolená hodnota parametru musí být větší nebo rovna parametru „Velikost okna“.

**Velikost okna** – opět se jedná o parametr automatické registrace obrazů. Hodnota, která musí být v celočíselném rozsahu  $\langle 0, 20 \rangle$  ovlivňuje velikost plovoucího okna. Toto plovoucí okno načítá hodnoty kriteriální funkce, ze kterých je stanoveno maximum. Tato maxima se odečtou a pokud je vypočtený rozdíl v předem definovaném rozsahu dojde k inkrementaci čítače.

Po proběhnutí registrace obrazů se zpřístupní patřičné položky v menu *Zobrazení*. Jedná se o zobrazení průběhů podobnostních funkcí a obrazových výsledků registrace obrazů. Využité podobnostní funkce byly uvedeny výše. Do obrazových výsledků patří zobrazení rozdílového obrazu bez potlačeného rozdílového artefaktu, zobrazení rozdílového obrazu s potlačeným rozdílovým artefaktem, mapa vektorů pohybu *quiver* a prezentace pomocí šachovnice.

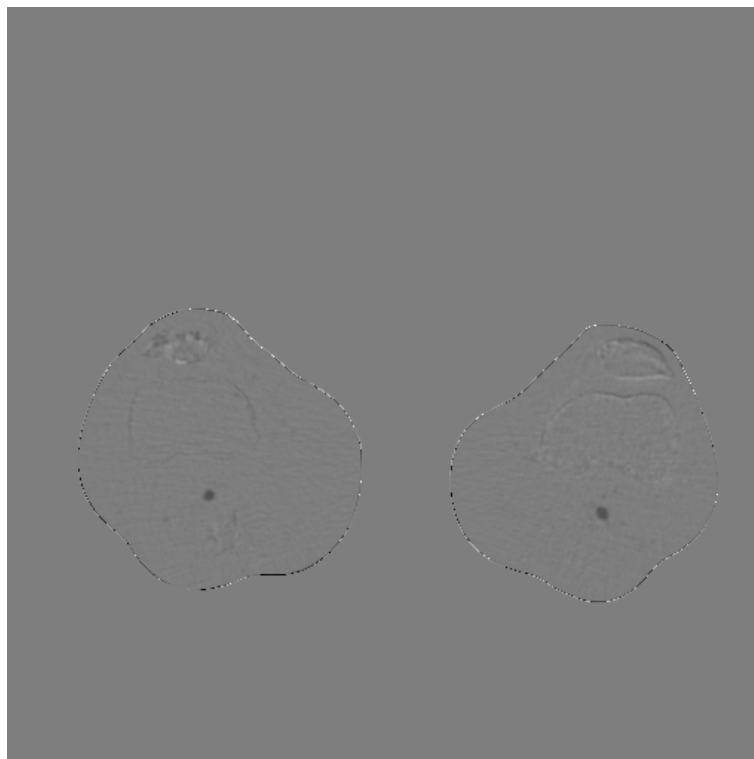
- V případě rozdílového obrazu (s potlačeným i nepotlačeným artefaktem) se jedná o rozdíl postkontrastního obrazu a výsledného registrovaného obrazu.
- V případě prezentace pomocí „šachovnice“ se jedná o zobrazení prekontrastního obrazu jako bílých polí šachovnice a rozdílového (s potlačeným rozdílovým artefaktem) obrazu jako černých polí šachovnice.
- Mapa vektorů pohybu je vytvořena pomocí funkce Matlabu s názvem *quiver*. Mapa znázorňuje výsledný registrovaný obraz, na kterém je pomocí vektorů zobrazen gradient postkontrastního obrazu.

Odstranění rozdílového artefaktu bylo provedeno pomocí základní morfologické operace s názvem eroze. Obraz byl převeden do binární podoby. Poté byla vytvořena maska, která vyhodnocuje obsah, který se pod ní vyskytuje. Odpovídá-li maska obsahu obrazu, zapíše se do příslušné pozice výsledného obrazu jednička. Účinky eroze jsou spojené se zmenšením objektu, odstraněním pruhů a hranic mezi objekty nebo k odstranění vzniklého rozdílového artefaktu.

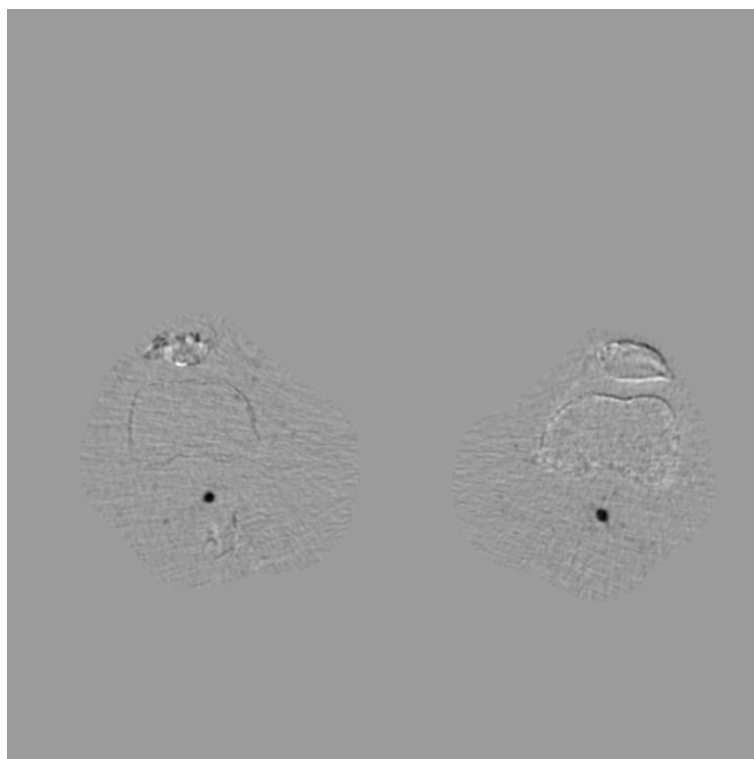


Obrázek 36. – Blokové schéma realizované eroze pro odstranění rozdílového artefaktu

Pro vytvoření masky byla použita funkce *strel*. Z důvodu tvaru registrovaných obrazů (nohou) a provedených experimentů byl morfologickým strukturujícím elementem zvolen objekt ve tvaru disku, který poskytoval nejlepší výsledky. Samotné provedení eroze bylo realizováno pomocí funkce *imerode*. Po provedení eroze byl binární obraz prahován.



*Obrázek 37. – Vzniklý rozdílový artefakt (20 iterací,  $\alpha = 1,5$ )*



*Obrázek 38. – Kompletní odstranění rozdílového artefaktu (20 iterací,  $\alpha = 1,5$ )*

Na předchozích obrázcích byly ukázány jednotlivé rozdílové obrazy (s artefaktem i bez něj). Je třeba říci, že k řádnému odstranění rozdílového artefaktu dochází, pokud jsou si výsledné obrazy co nejvíce podobné.

## 4. Prezentace a zhodnocení dosažených výsledků

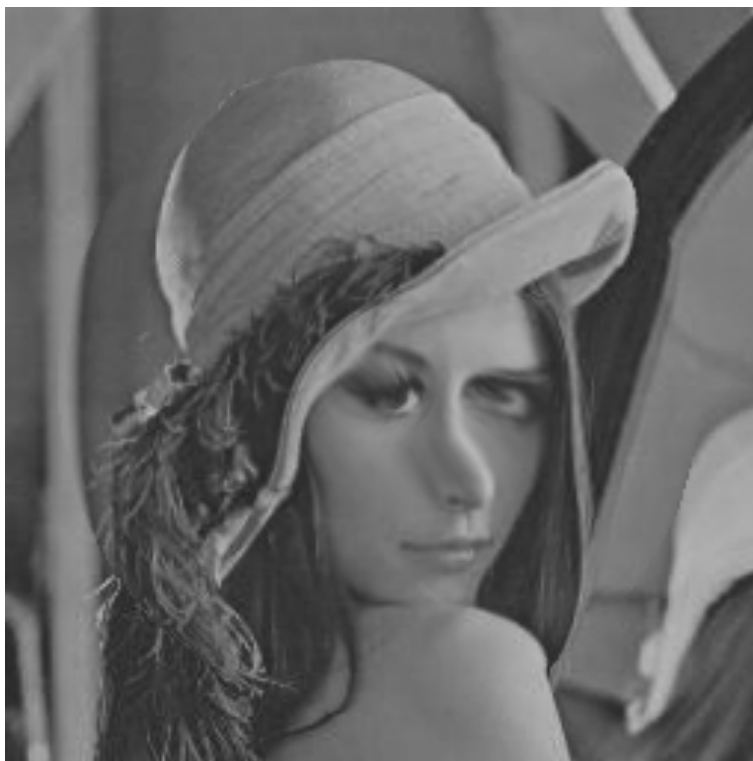
V této kapitole práce budou představeny a zhodnoceny dosažené výsledky registrace obrazů pomocí metody Optical Flow při využití algoritmu „Demon“. Zhodnocení bude provedeno na testovacích a reálných obrazech dodaných vedoucím diplomové práce. Postupně budou měněny jednotlivé hodnoty parametrů registrace a budou zobrazeny výsledné registrované obrazy, průběhy podobnostních funkcí a v neposlední řadě bude diskutován vliv jednotlivých parametrů na výsledný registrovaný obraz.

Připomeňme, že cílem této diplomové práce bylo prostudování problematiky registrace obrazů pomocí metody Optical Flow, navržení a realizování programu v prostředí Matlab a vyhodnocení výsledků na reálných obrazech, které byly dodány vedoucím práce. Realizované programové prostředí bylo uvedeno výše. V této kapitole bude otestována funkčnost realizovaného programu na klasických obrázcích („Lena“) a poté budou testována reálná medicínská data (dodaná vedoucím diplomové práce).

### *Testování iterační formy metody registrace obrazů pomocí algoritmu „Demon“ na obrazech „Lena“*

Nejprve ověříme klasickou iterační formu registrace obrazů a to pro obě varianty vzorců pro výpočet „Demonů“.

**Test 1:** Provedme nejprve test s využitím základního vzorce pro výpočet „Demonů“ a s volbou 20-ti iterací. Na následujícím obrázku č. 39 je uveden pohybový zkreslený obraz a na obrázku č. 40 je uveden statický referenční obraz.



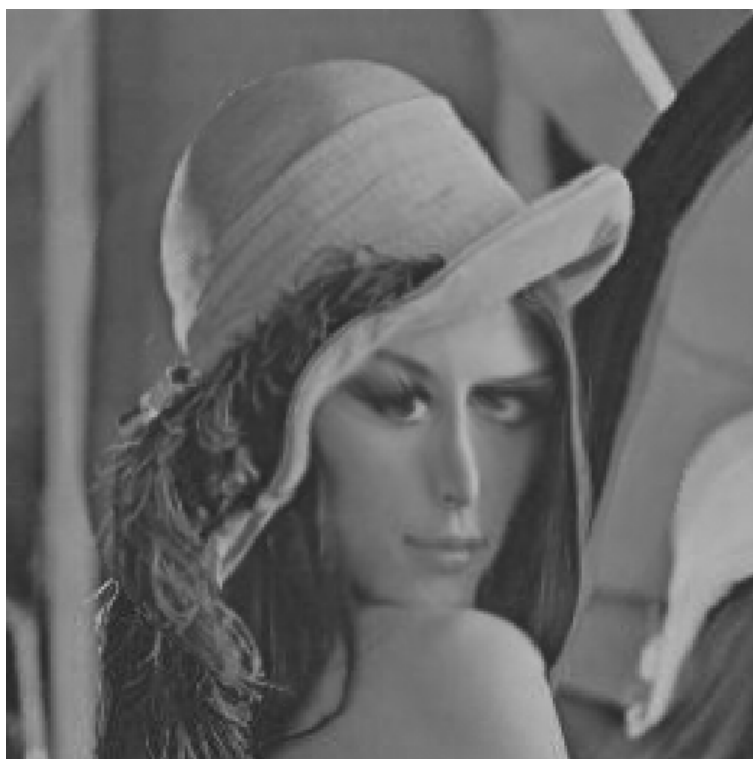
*Obrázek 39. – Pohybový testovaný obraz pro registraci*





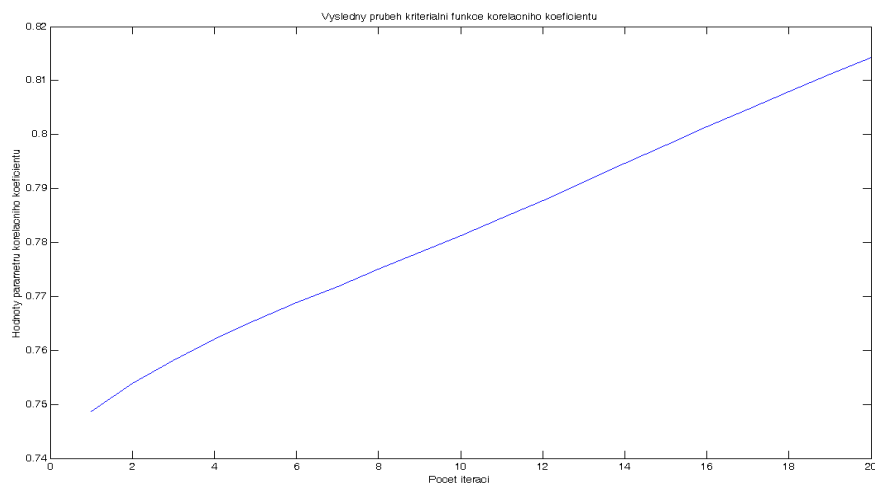
*Obrázek 40. – Referenční testovaný obraz pro registraci*

Využijeme-li základní vzorec pro výpočet „Demonů“ (2.26) a provedeme 20 iterací, získáme následující registrovaný obraz. Čas potřebný pro registraci byl přibližně 4 sekundy.

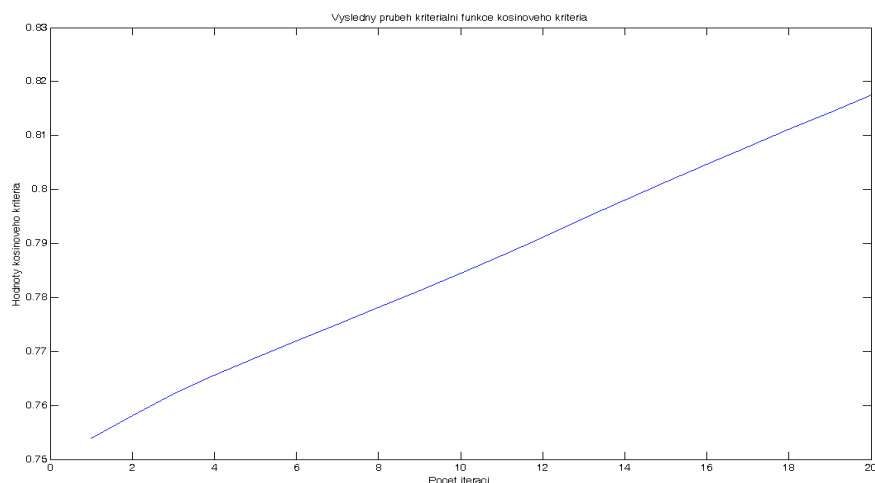


*Obrázek 41. – Výsledný registrovaný obraz (základní vzorec, 20 iterací)*

Z výsledného registrovaného obrazu je patrné, že byl počet iterací nevyhovující. V následujících grafech budou znázorněny průběhy kritériálních funkcí, které byly vypočteny během registrace obrazů.



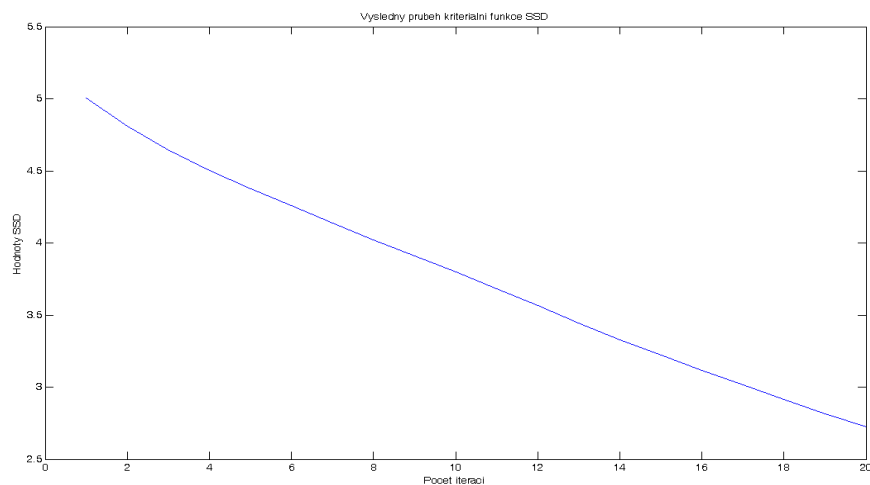
*Graf 1. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 20 iterací)*



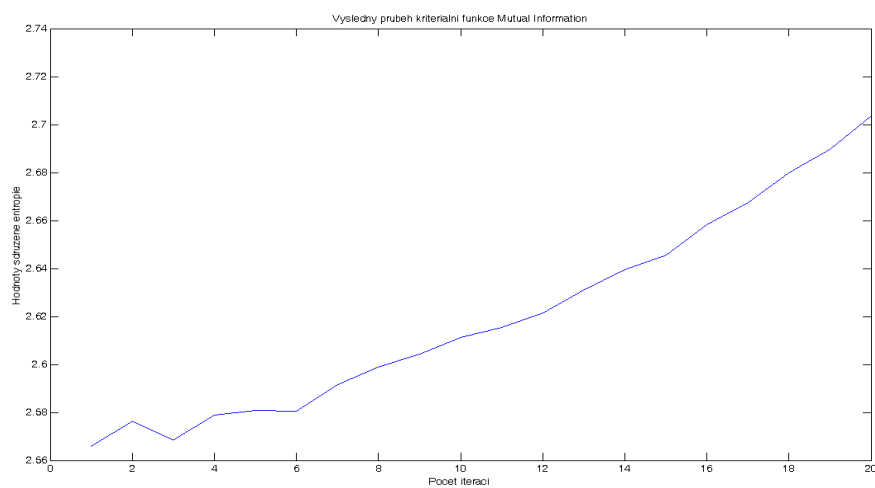
*Graf 2. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 20 iterací)*

Z průběhů je patrné, že po dokončení registrace obrazů byla, dle korelačního koeficientu, dosažená podobnost mezi referenčním a pohybovým obrazem přibližně 82%.

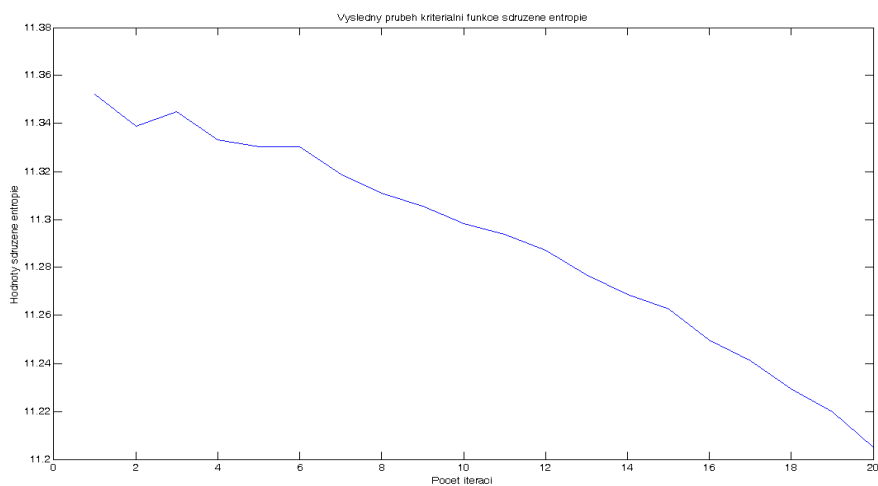
Na následujících grafech budou uvedeny zbývající průběhy realizovaných kritériálních funkcí.



Graf 3. – Průběh podobnostní funkce SSD (základní vzorec, 20 iterací)



Graf 4. – Průběh podobnostní funkce Mutual Information (základní vzorec, 20 iterací)

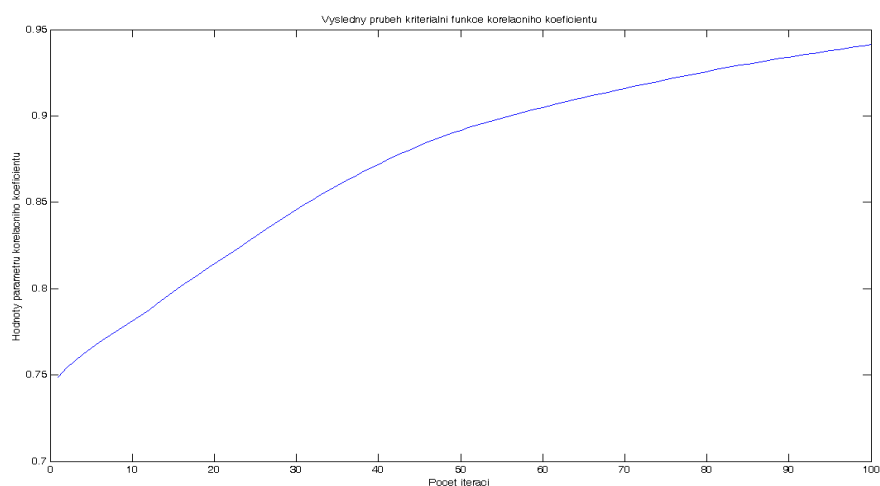


Graf 5. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 20 iterací)

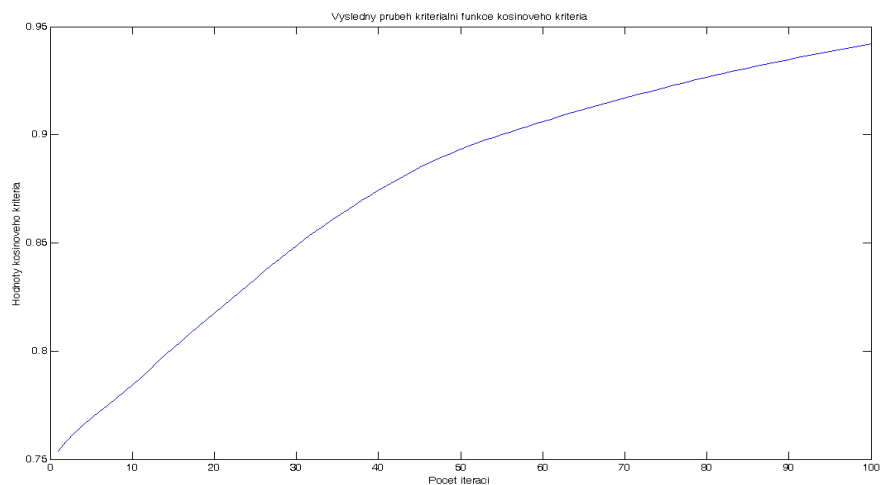
**Test 2:** Z předchozího testu je patrné, že zvolených 20 iterací bylo při tak značné deformaci pohybového obrazu nedostatečných. V tomto testu bude zvoleno 100 iterací a opět základní forma vzorce pro výpočet „Demonů“. Referenční i pohybový obraz zůstává stejný.



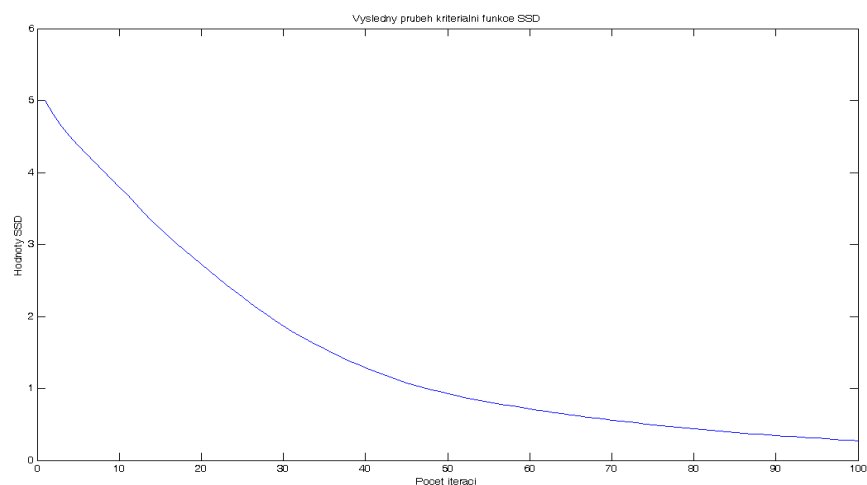
*Obrázek 42. – Výsledný registrovaný obraz (základní vzorec, 100 iterací)*



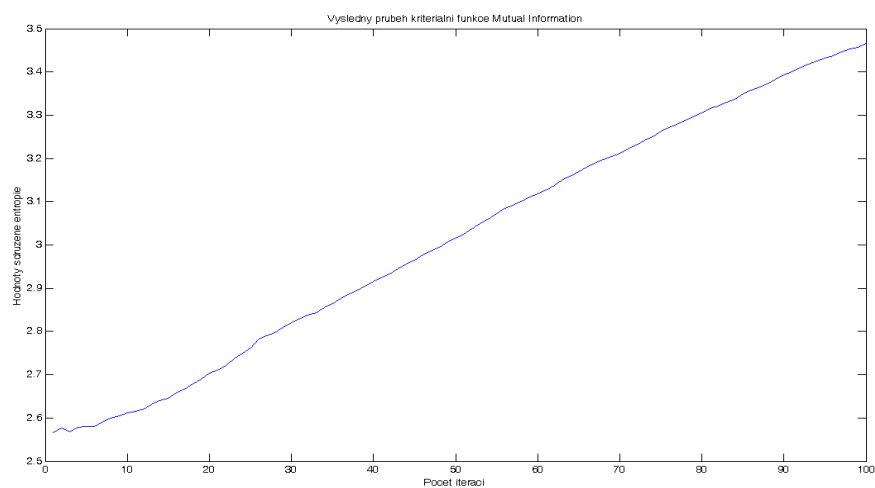
*Graf 6. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 100 iterací)*



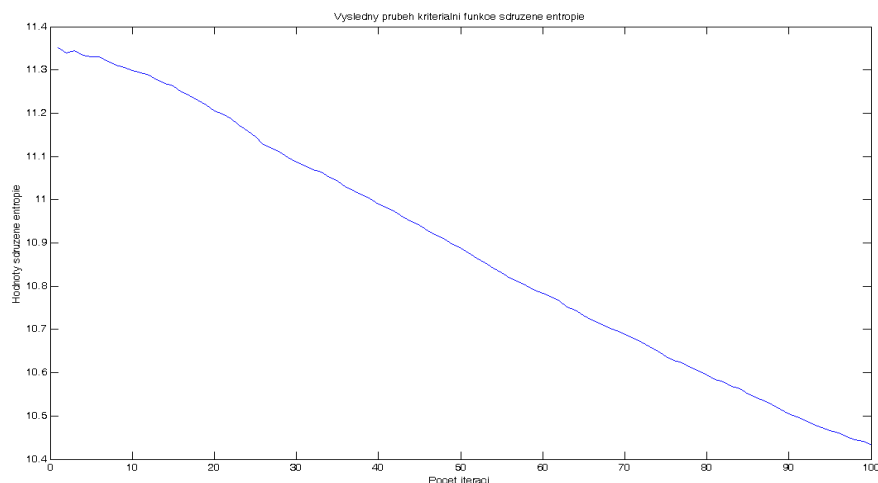
Graf 7. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 100 iterací)



Graf 8. – Průběh podobnostní funkce SSD (základní vzorec, 100 iterací)



Graf 9. – Průběh podobnostní funkce Mutual Information (základní vzorec, 100 iterací)

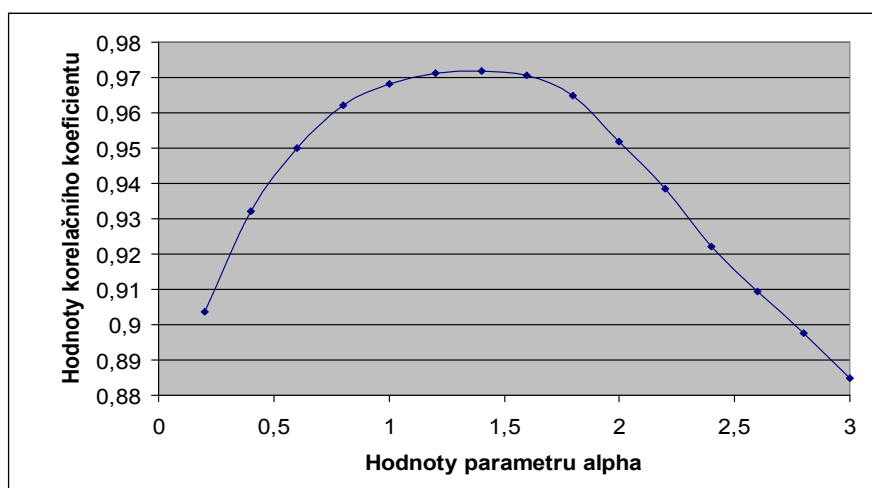


Graf 10. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 100 iterací)

Z obrázku č. 42 je patrné, že výsledný registrovaný obraz je stále odlišný od původního referenčního obrazu. Podstatně lepších výsledků registrace uvažovaných obrazů nebylo dosaženo ani při neustálém zvyšování počtu iterací. Výsledná podobnost obrazů, stanovená na základě korelačního koeficientu, byla 94,15%, přičemž samotný proces registrace trval přibližně 19 sekund.

U předchozích dvou testů byl pro výpočet „Demonů“ využit pouze základní vzorec. V následujících testech bude ukázáno, jaký vliv má parametr  $\alpha$  na výsledek registrace obrazů.

**Test 3:** V následujícím testu budeme využívat k výpočtům „Demonů“ rozšířený vzorec (2.27). Postupně bylo za parametr  $\alpha$  dosazeno celkem 15 hodnot od 0 – 3 s krokem 0,2. Při všech testech bylo použito 100 iterací.



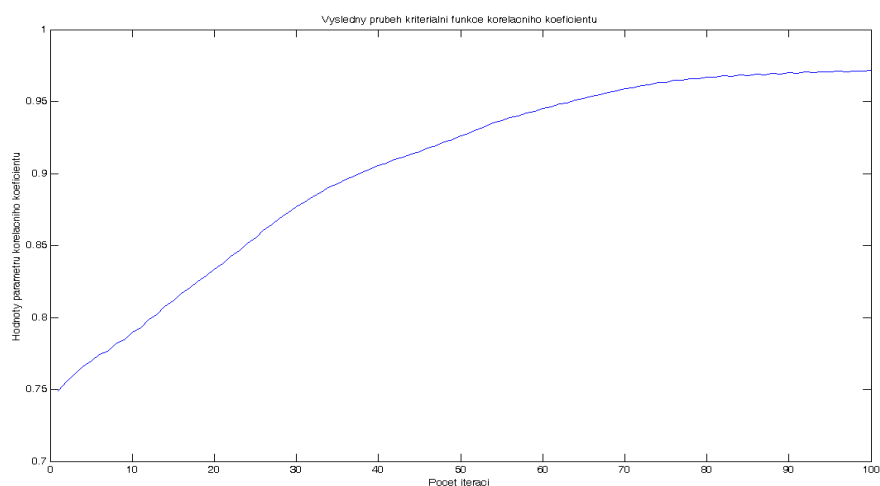
Graf 11. – Vliv velikosti parametru alpha na výslednou podobnost obrazů (100 iterací)

Z předchozího grafu je patrné, že nejlepší výsledky registrace testovaných obrazů byly dosaženy při velikosti parametru  $\alpha$  v rozsahu 1,4 – 1,6. Registrace byla provedena pomocí 100 iterací a průměrná doba potřebná pro registraci byla 21,5 s.

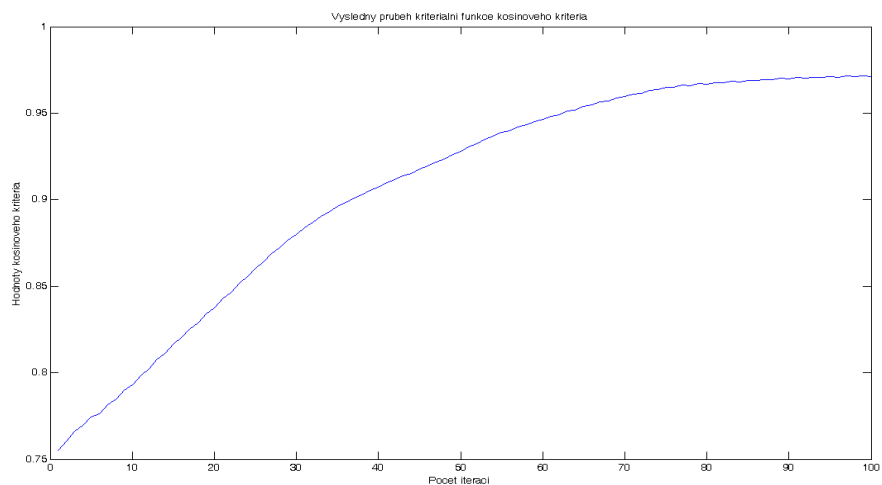
Uved'me si nyní průběhy podobnostních funkcí a výsledný registrovaný obraz pro hodnotu  $\alpha=1,5$ . Tato hodnota byla zvolena jako vhodný kompromis (viz graf 11) pro dosažení řádných výsledků registrace obrazů.



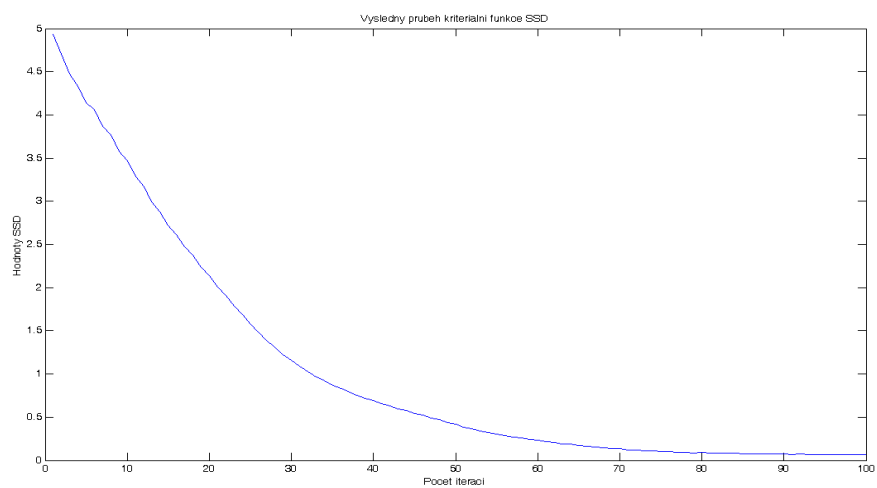
*Obrázek 43. – Výsledný registrovaný obraz (rozšířený vzorec,  $\alpha=1,5$  , 100 iterací)*



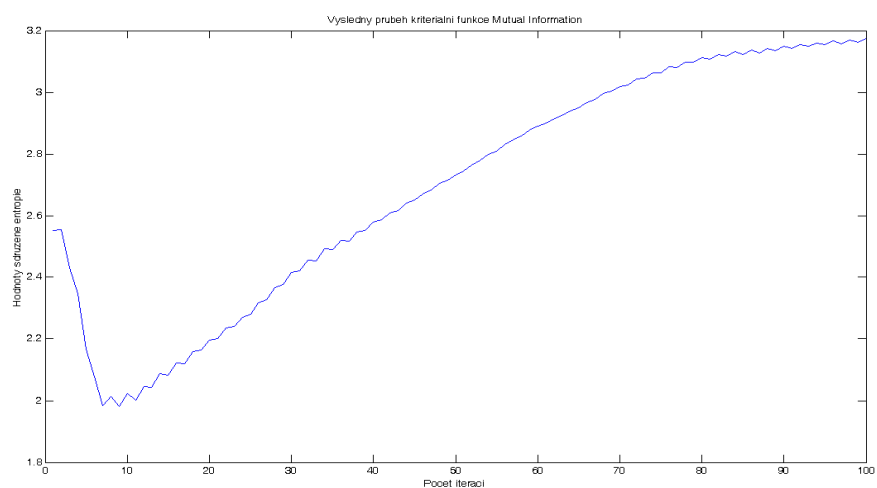
*Graf 12. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 100 iterací)*



Graf 13. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 100 iterací)



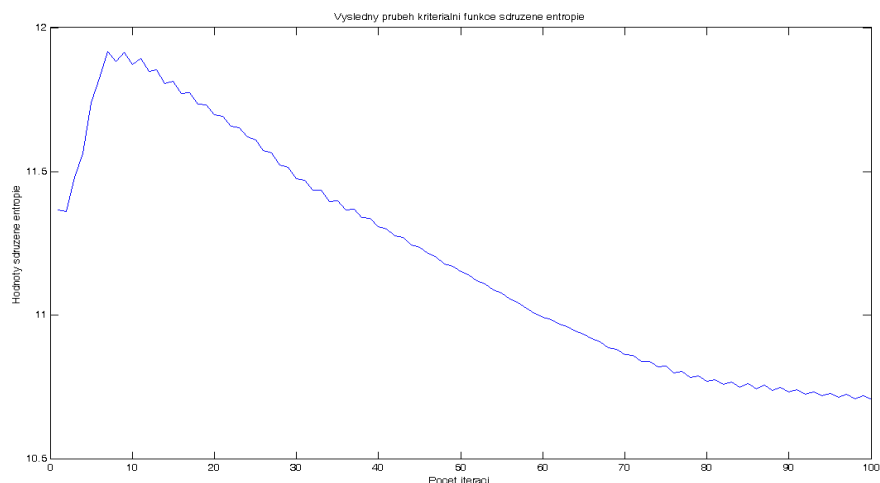
Graf 14. – Průběh podobnostní funkce SSD (základní vzorec, 100 iterací)



Graf 15. – Průběh podobnostní funkce Mutual Information (základní vzorec, 100 iterací)



Z provedeného testu jasně vyplývá, že použití rozšířeného vzorce pro výpočet „Demonů“ a vhodné nastavení parametru  $\alpha$  dává lepší výsledky než použití vzorce základního. Při volbě  $\alpha=1,5$  a 100 iteračních krocích byla získána 97,16% procentní podobnost mezi registrovanými obrazy. Vezmeme-li v potaz, že počáteční podobnost obrazů je pouze 74,87%, jedná se o dobrý výsledek. Testované obrazy však dostatečně nesplňují základní podmínku řádné registrace obrazů, která uvádí, že obrazy by si měly na začátku registrace co nejvíce odpovídat.



*Graf 16. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 100 iterací)*

V předchozích testech jsme tedy pracovali s jistým omezením (viz podmínka výše), které se vyskytovalo v podobě značně rozdílných obrazů na počátku registrace. Provedme nyní dva testy, které budou uvažovat rigidní transformaci mezi testovanými obrazy.

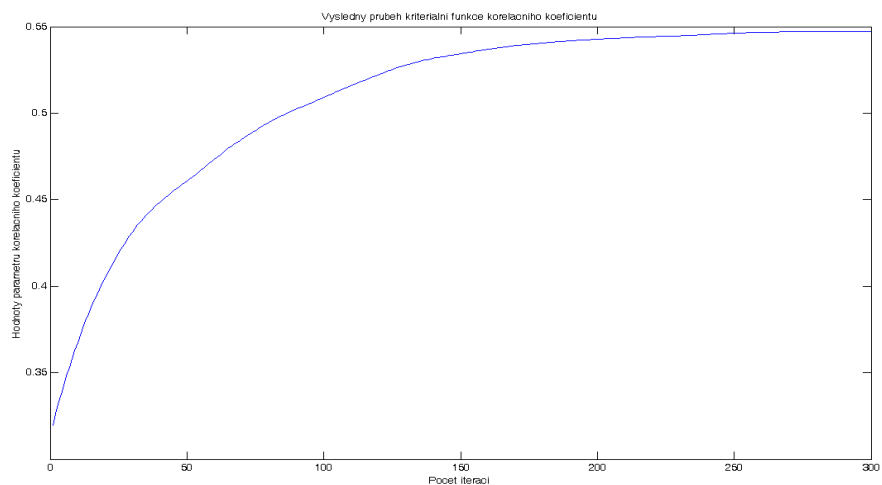
**Test 4:** V tomto testu budeme k výpočtu „Demonů“ uvažovat pouze základní vzorec. Jako referenční obraz využijeme opět obrázek 40 a jako pohybový obraz nově obrázek 44. Jednoznačným počátečním omezením je zde černá oblast na krajích pohybového obrazu a stejně tak i neostře ohraničení černých oblastí. Počáteční podobnost obrazů vypočtená pomocí korelačního koeficientu je pouze 31,97 %. V testu využijeme 300 iteračních kroků.



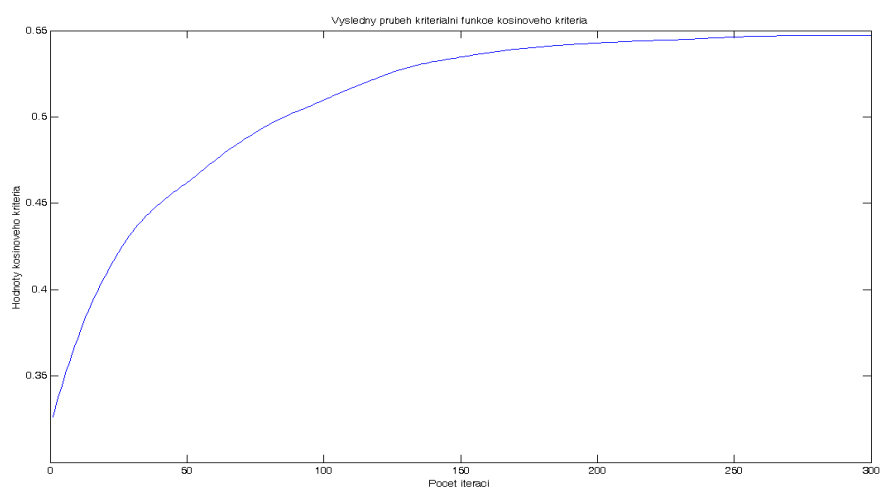
*Obrázek 44. – Pohybový testovaný obraz pro registraci*



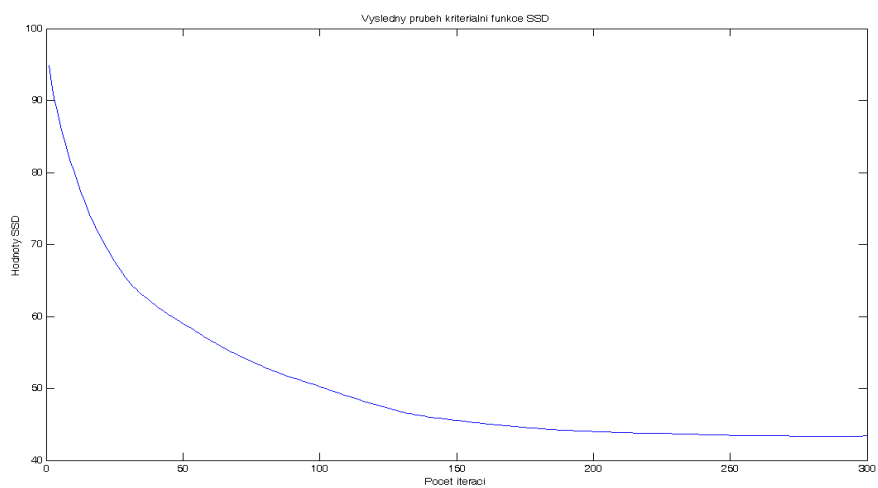
*Obrázek 45. – Výsledný registrovaný obraz (základní vzorec, 300 iterací)*



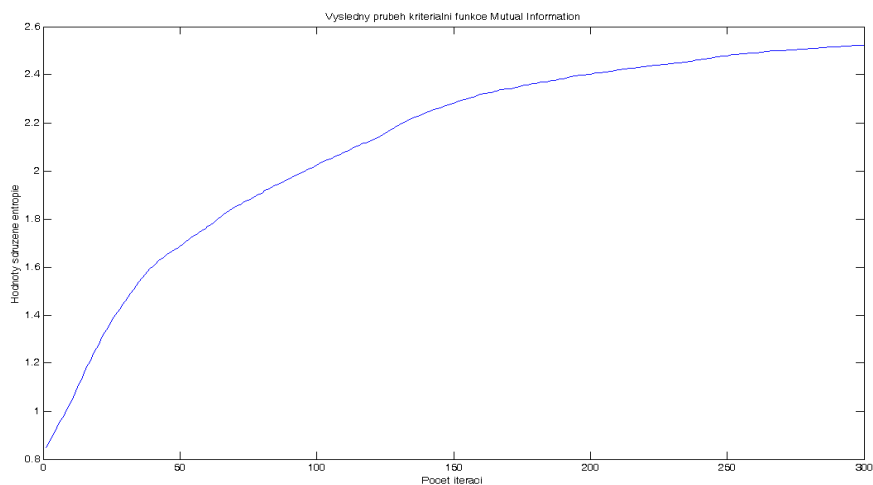
Graf 17. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 300 iterací)



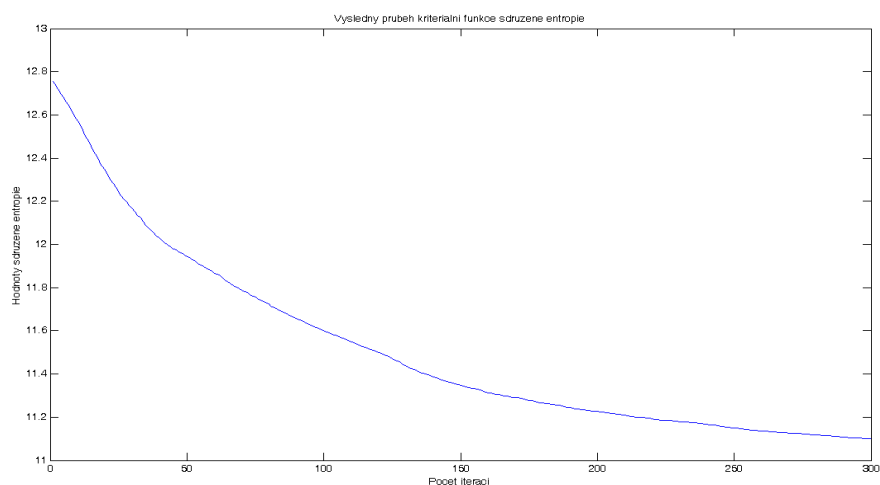
Graf 18. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 300 iterací)



Graf 19. – Průběh podobnostní funkce SSD (základní vzorec, 300 iterací)



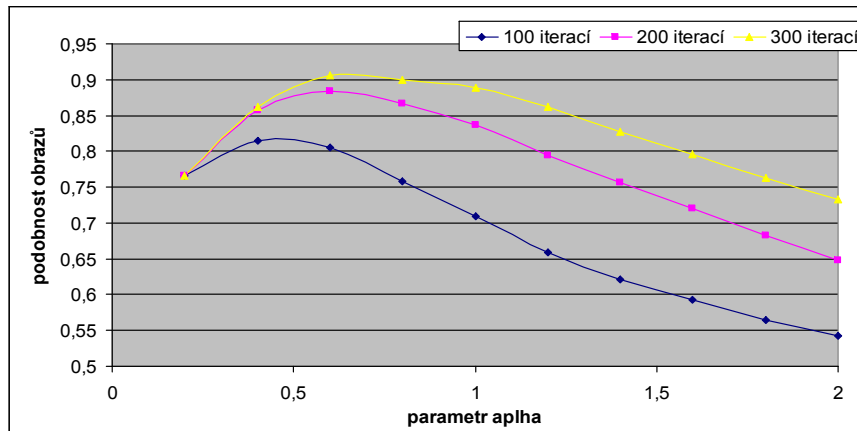
Graf 20. – Průběh podobnostní funkce Mutual Information (základní vzorec, 300 iterací)



Graf 21. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 300 iterací)

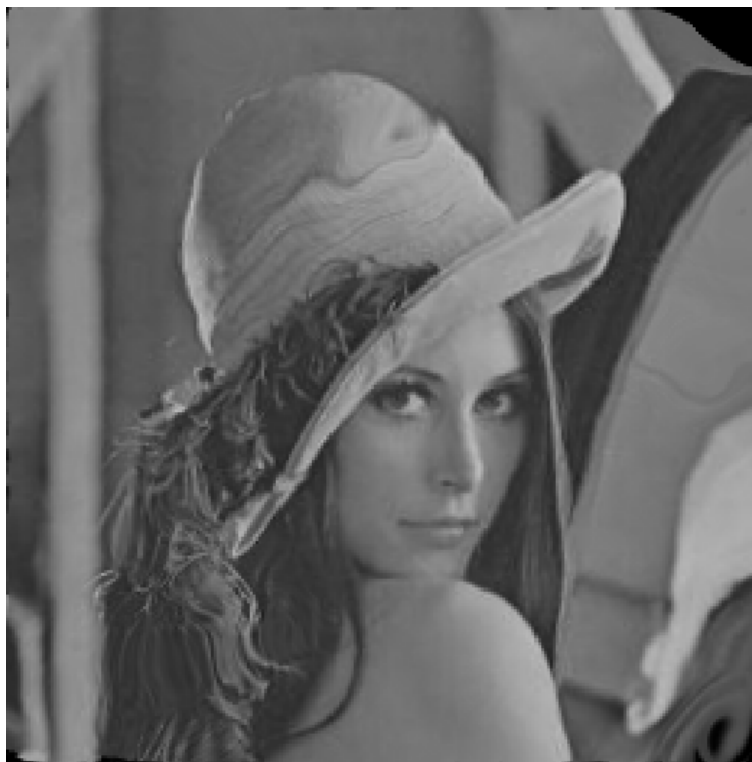
Výsledky tohoto testu nejsou uspokojivé. Využijeme-li základní vzorec pro výpočet „Demonů“, nedojde ani při více než 300 iteracích k řádné registraci obrazů. Toto omezení je způsobeno výpočtem gradientu pouze jednoho obrazu. Výsledná podobnost dle korelačního koeficientu má hodnotu 54,71%, čas registrace byl přibližně 60 s.

**Test 5:** V posledním testu s obrazy „Lena“ využijeme rozšířený vzorec pro výpočet „Demonů“. Nejprve nalezneme, stejně jako v testu č. 3, optimální hodnotu parametru  $\alpha$ , při 100, 200 a 300 iteracích.

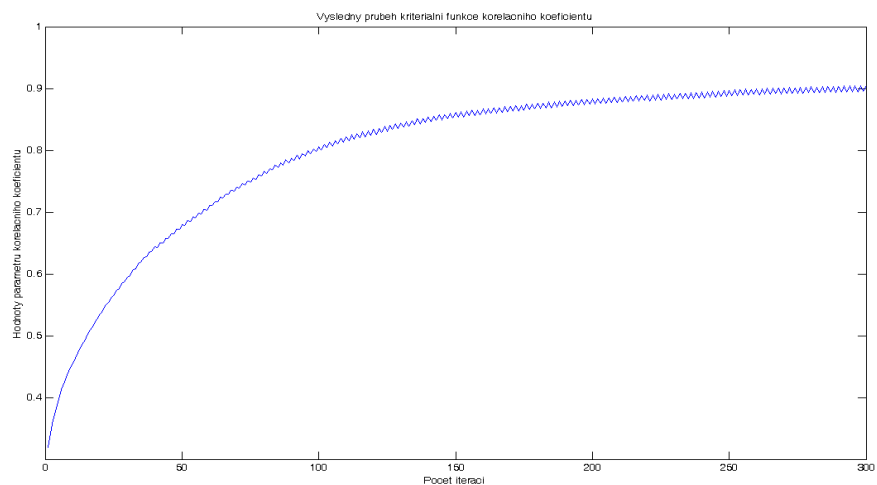


*Graf 22. – Vliv velikosti parametru alpha na výslednou podobnost obrazů (100, 200, 300 iterací)*

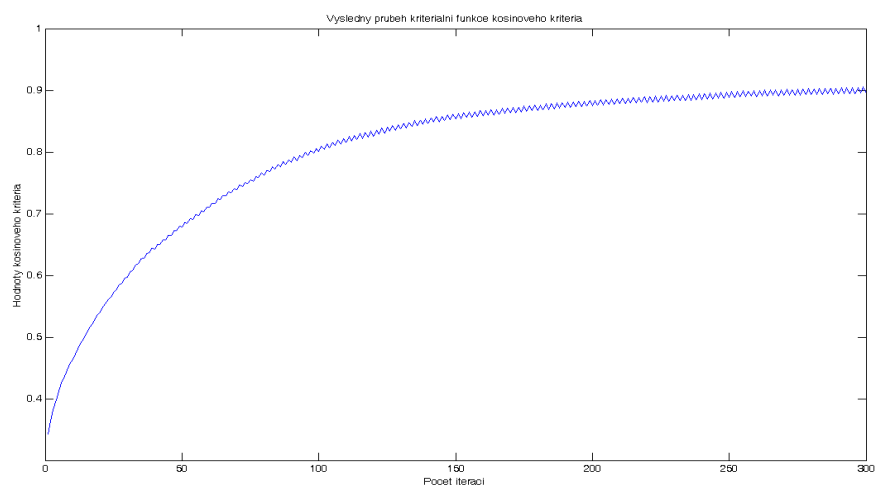
Z předchozího grafu je patrné, že nejlepší výsledky byly dosaženy při  $\alpha = 0,6$  a 300 iteracích. Postupné zvyšování počtu iteračních kroků nad 300 již nemělo žádný významný vliv na výslednou podobnost registrovaných obrazů.



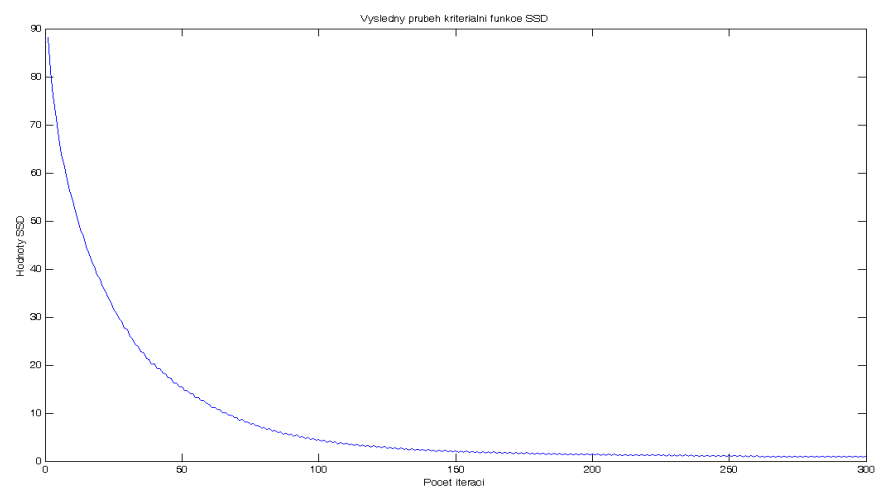
*Obrázek 46. – Výsledný registrovaný obraz (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)*



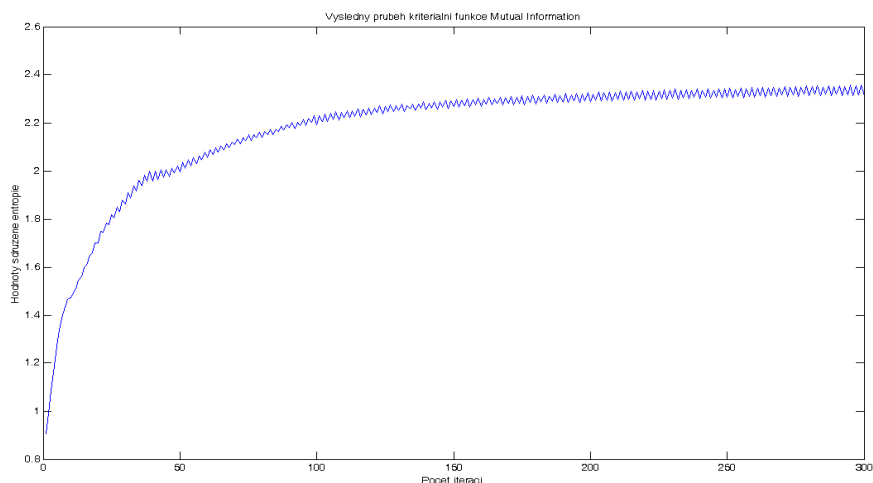
Graf 23. – Průběh podobnostní funkce korelační koeficient (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)



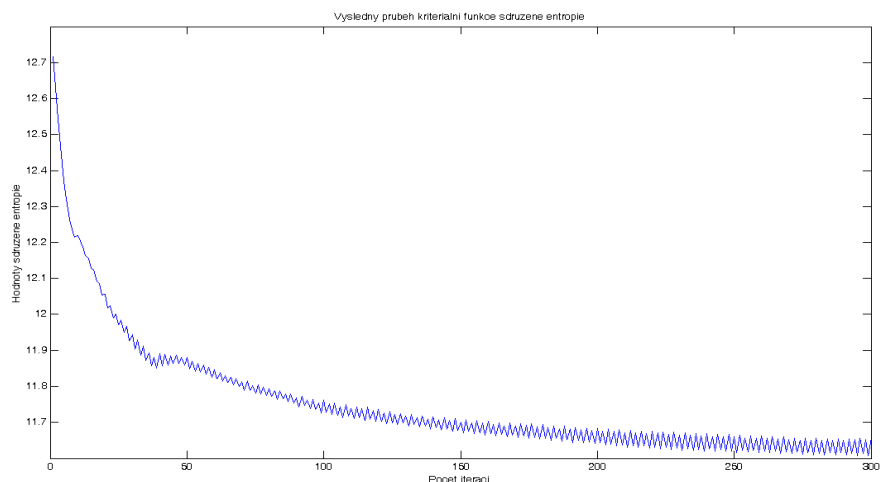
Graf 24. – Průběh podobnostní funkce kosinové kritérium (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)



Graf 25. – Průběh podobnostní funkce SSD (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)



Graf 26. – Průběh podobnostní funkce *Mutual Information* (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)



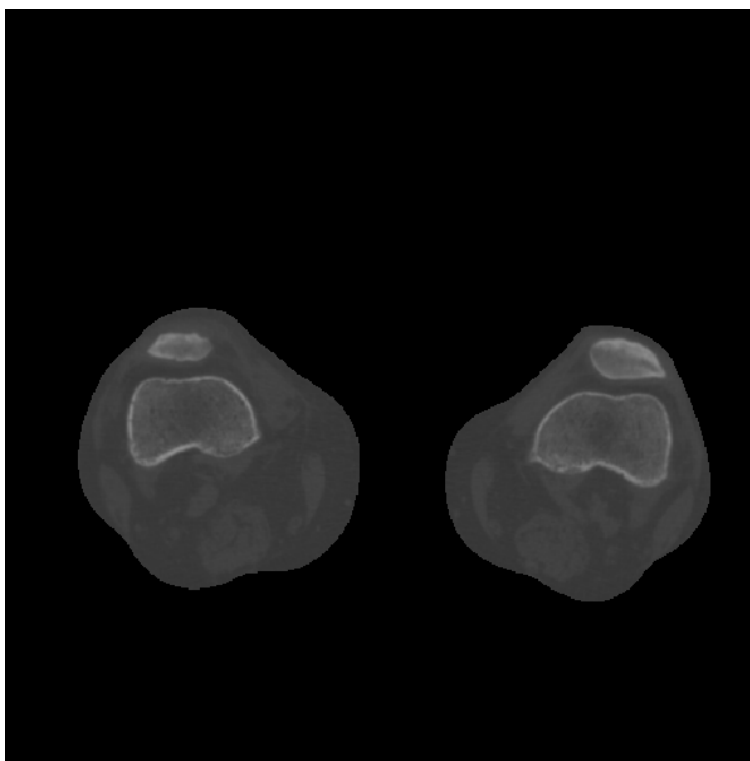
Graf 27. – Průběh podobnostní funkce sdružené entropie (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)

V předchozích grafech je vidět, že charakteristiky kritériálních funkcí nejsou zcela hladké. Toto omezení vzniká z důvodu černých ploch v pohybovém obrazu. Z výsledného obrázku č. 46 je patrné, že střední část výsledného obrazu je řádně registrovaná. Nepřesnosti a černé plochy, které vznikly na krajích výsledného obrazu se nepodařilo zcela odstranit. Využijeme-li rozšířeného vzorce k výpočtu „Demonů“ dochází (viz kapitola 2) k vzájemnému prolínání registrovaných obrazů. Při nastavení parametru  $\alpha = 0,6$  bylo dosaženo nejlepších výsledků. Menší i větší hodnoty parametru  $\alpha$  vykazovaly podstatně horší odstranění černých ploch na okrajích a poskytovaly tedy menší podobnost registrovaných obrazů. Doba registrace byla přibližně 75 sekund a dosažená výsledná podobnost mezi registrovanými obrazy byla dle korelačního koeficientu 90,52%.

### **Testování iterační formy metody registrace obrazů pomocí algoritmu „Demon“ na reálných CT obrazech**

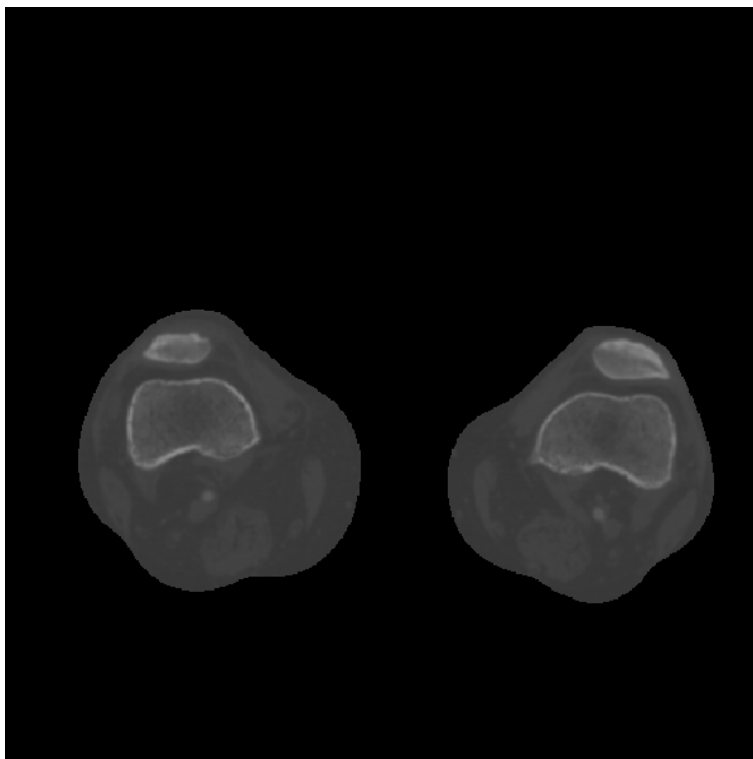
V této části práce budou představeny dosažené výsledky registrace obrazů získaných z CT zobrazovací modality. Připomeňme, že se jedná o soubor řezů nohama. První obraz je snímán před dodáním kontrastu (prekontrastní obraz), druhý obraz pak po dodání kontrastu (postkontrastní obraz). Testovaná reálná data splňují podmínku řádné registrace, neboť jejich vzájemná podobnost je před začátkem registrace (dle korelačního koeficientu) přibližně 98 %. Cílem registrace těchto obrazů je zobrazení cév. Toho dosáhneme co nejpřesnější registrací prekontrastního a postkontrastního obrazu a následným odečtením patřičných obrazů. Odečítáme výsledný registrovaný obraz od obrazu prekontrastního.

Jak bylo uvedeno výše, je toto odečtení spojeno se vznikem okrajového artefaktu, který je vhodné odstranit. V následující části práce budou provedeny dva testy ověřující funkčnost realizované metody a vytvořeného programu. Obrazy, které budou registrovány, jsou si velmi podobné, rozdíly jsou zejména u zobrazených kostních struktur a na hranách obrazů.



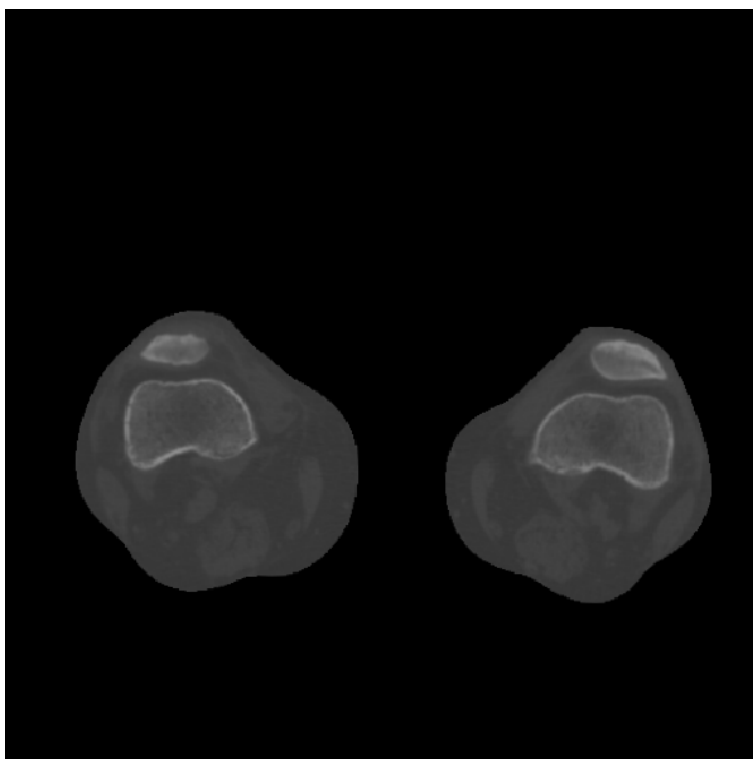
*Obrázek 47. – Prekontrastní obraz řezu nohama (soubor Inohy3DPre.mat)*



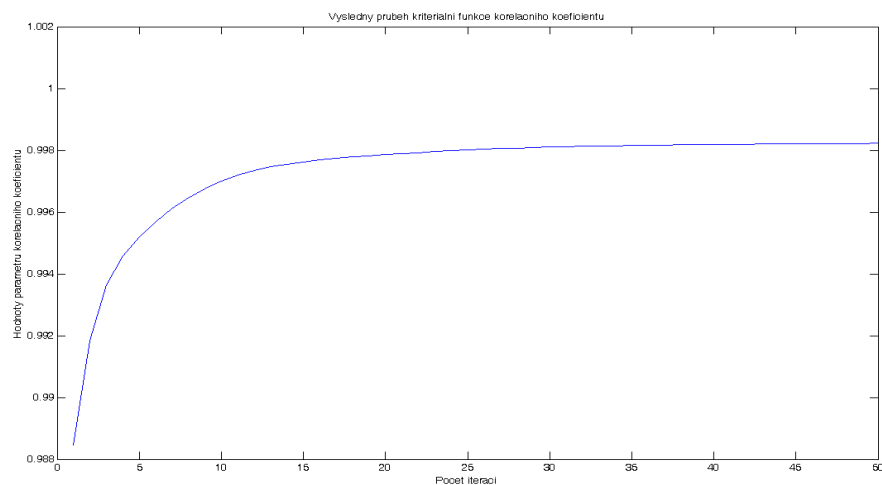


*Obrázek 48. – Postkontrastní obraz řezu nohama (soubor Inohy3DPost.mat)*

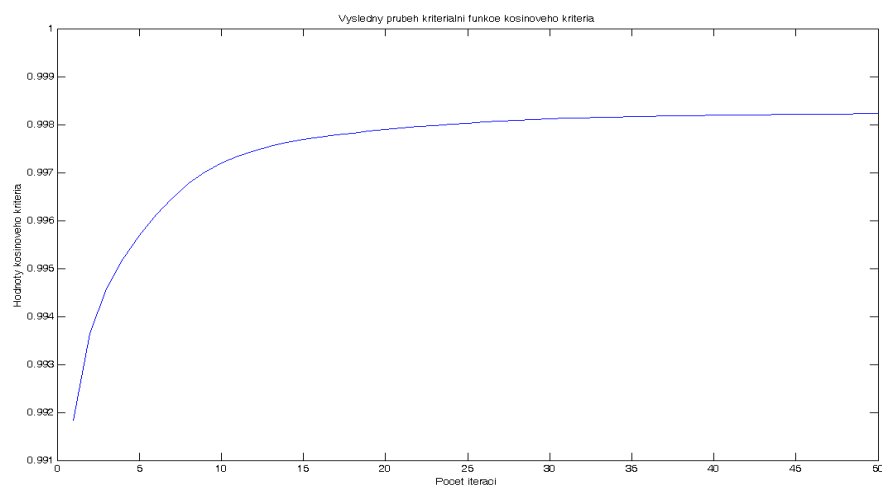
**Test 1:** Vyděme nejprve ze základního vzorce pro výpočet „Demonů“. S ohledem na počáteční shodu mezi registrovanými obrazy bylo zvoleno 50 iteračních kroků.



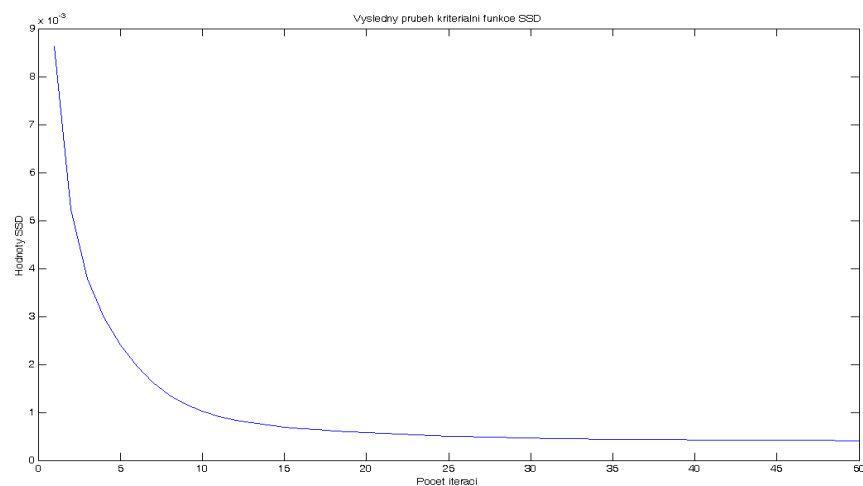
*Obrázek 49. – Výsledný registrovaný obraz (základní vzorec, 50 iterací)*



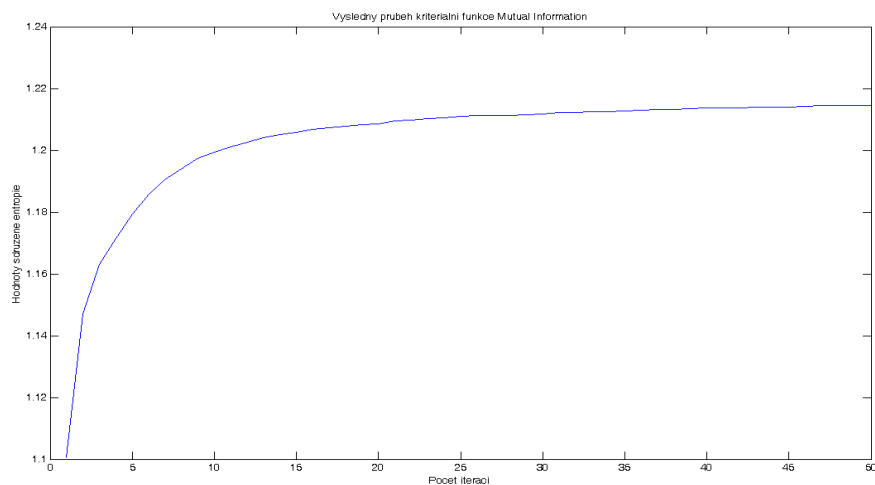
Graf 28. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 50 iterací)



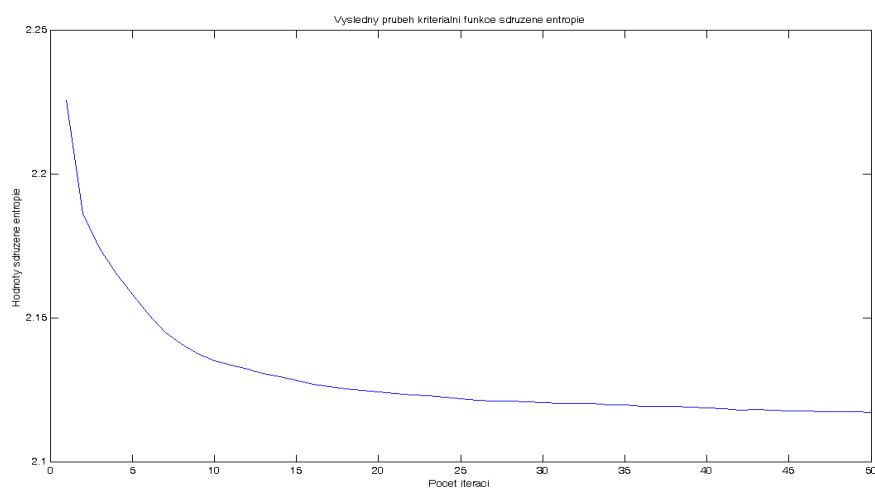
Graf 29. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 50 iterací)



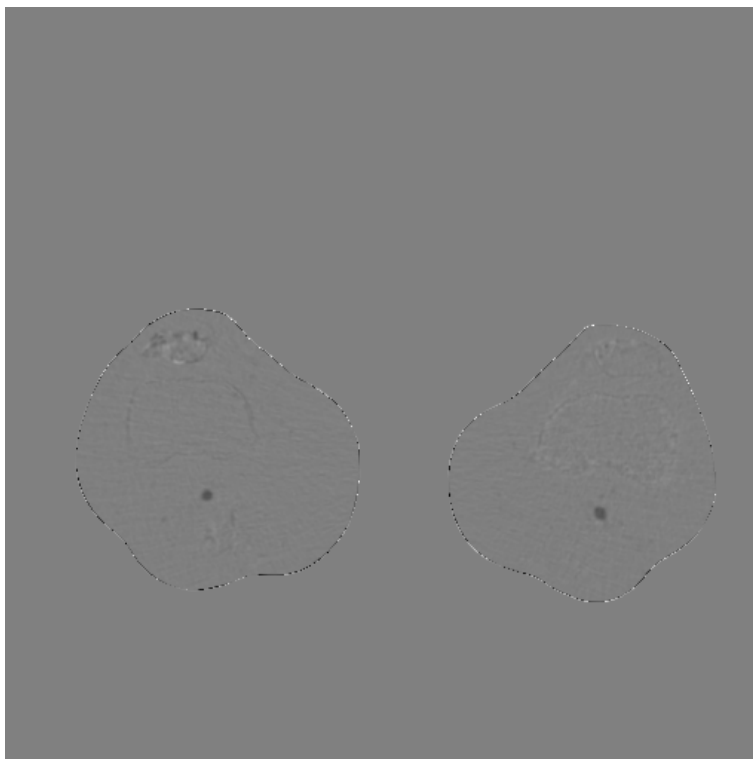
Graf 30. – Průběh podobnostní funkce SSD (základní vzorec, 50 iterací)



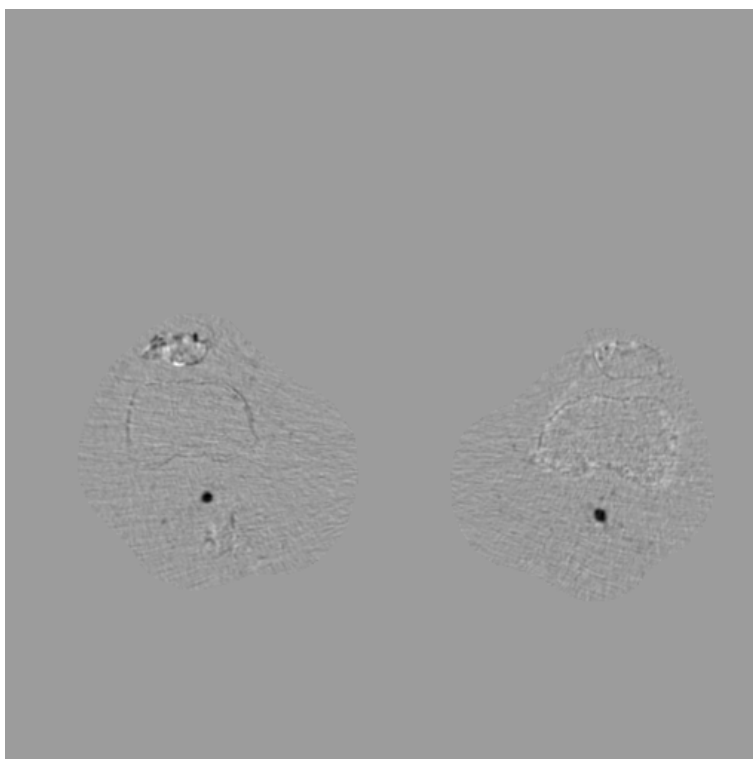
Graf 31. – Průběh podobnostní funkce Mutual Information (základní vzorec, 50 iterací)



Graf 32. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 50 iterací)



*Obrázek 50. – Výsledný rozdílový obraz bez potlačeného artefaktu (základní vzorec, 50 iterací)*

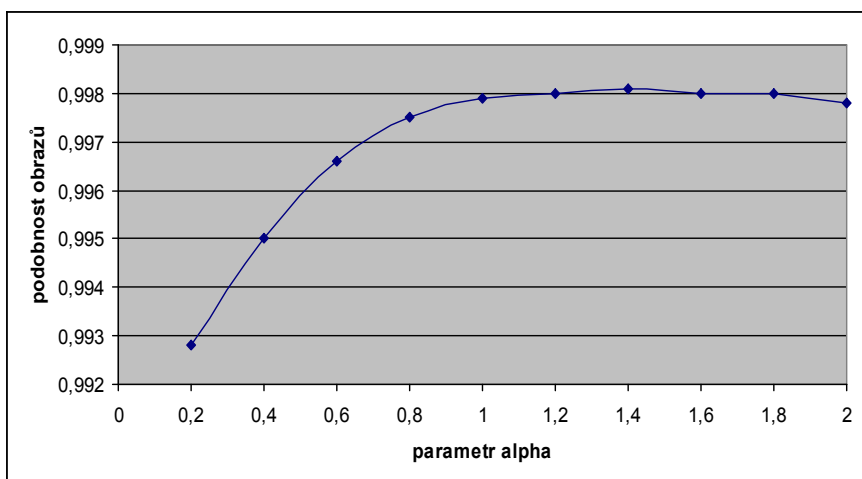


*Obrázek 51. – Výsledný rozdílový obraz s potlačeným artefaktem (základní vzorec, 50 iterací)*

Proběhlým testem byla ověřena podmínka řádné registrace obrazů. Tyto obrazy byly na začátku procesu registrace velmi podobné (viz výše). Výsledná podobnost obrazů byla na

základě korelačního koeficientu stanovena na 99,82%. Celková doba registrace byla 41,6 sekund. Na obrázku č. 50 je patrný rozdílový artefakt vznikající při odečtení výsledného obrazu od obrazu postkontrastního, který je způsoben rozdílnou velikostí odečítaných obrazů. Jeho odstranění, které je znázorněno na obrázku č. 51, může být vhodně realizováno pomocí základní morfologické operace zvané eroze (viz výše).

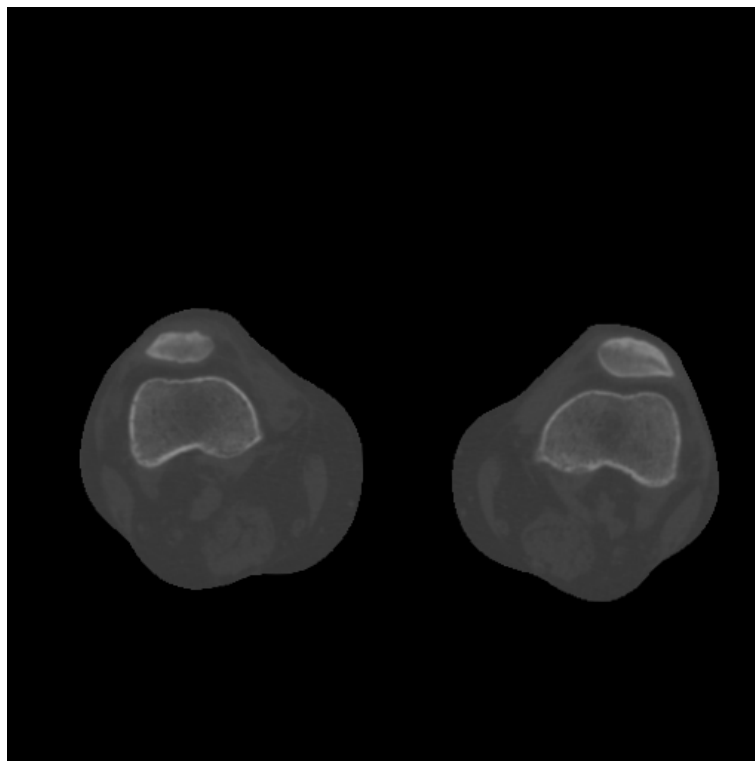
**Test 2:** V následujícím testu bude odzkoušen vliv rozšířeného vzorce na přesnost registrace obrazů. Nejprve nalezneme optimální velikost parametru  $\alpha$ , který zajistí co nejlepší výsledky registrace reálných CT obrazů.



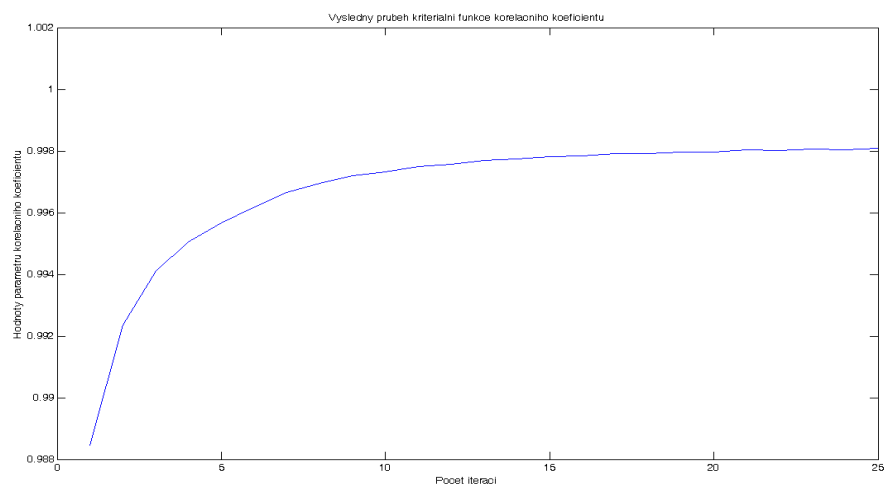
*Graf 33. – Vliv velikosti parametru alpha na výslednou podobnost obrazů (25 iterací)*

Parametr  $\alpha$  byl hledán při 20, 30 a 50 iteračních krocích. Z testu bylo zjištěno, že maximální hodnota podobnosti obrazů je dosažena již při 25 iteračních krocích. Při postupném zvyšování iteračních kroků dochází k naprosto nepatrnému zlepšení podobnosti mezi registrovanými obrazy. Závislost parametru  $\alpha$  (při 25 iteračních krocích) na výslednou podobnost obrazů je uvedena v grafu č. 33. Nejlepších výsledků bylo dosaženo při  $\alpha = 1,4$  (hodnota bude využita i při tomto testu). Během testu bylo zjištěno, že parametr  $\alpha$  významně ovlivňuje zobrazení kostní struktury v obraze. V rozsahu přibližně od 0,2 – 1 byly kostní struktury vystouplé a dobře identifikovatelné. Kolem hodnoty 1,2 dochází ke zlomu, začínají se zdůrazňovat dosud méně výrazné cévy, zatímco kostní a tkáňová struktura se stává méně výraznou. Od hodnoty 2 a více se tkáňová a kostní struktura stává téměř neidentifikovatelnou. Zůstávají zvýrazněny pouze cévy.

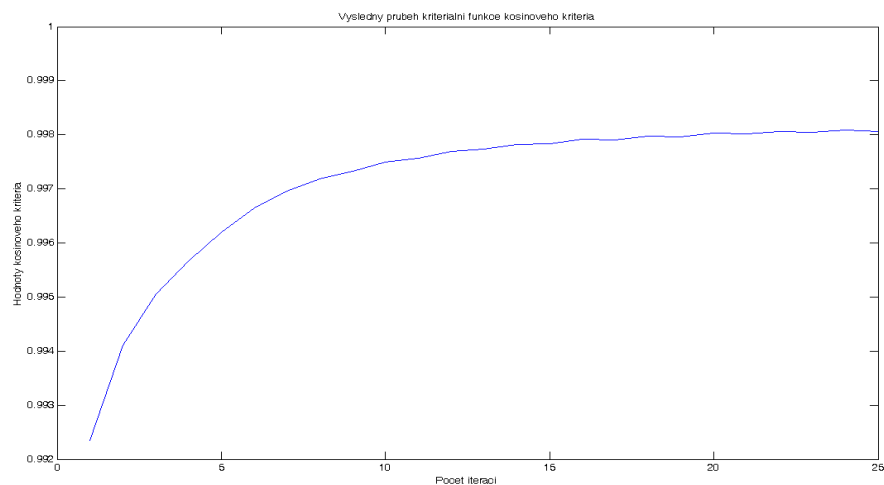
Na následujících grafech a obrázcích budou zobrazeny výsledky tohoto testu.



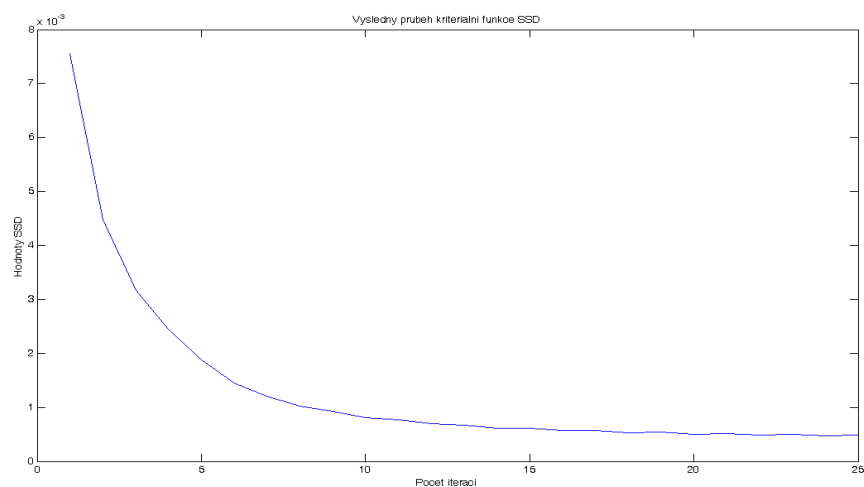
Obrázek 52. – Výsledný registrovaný obraz (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)



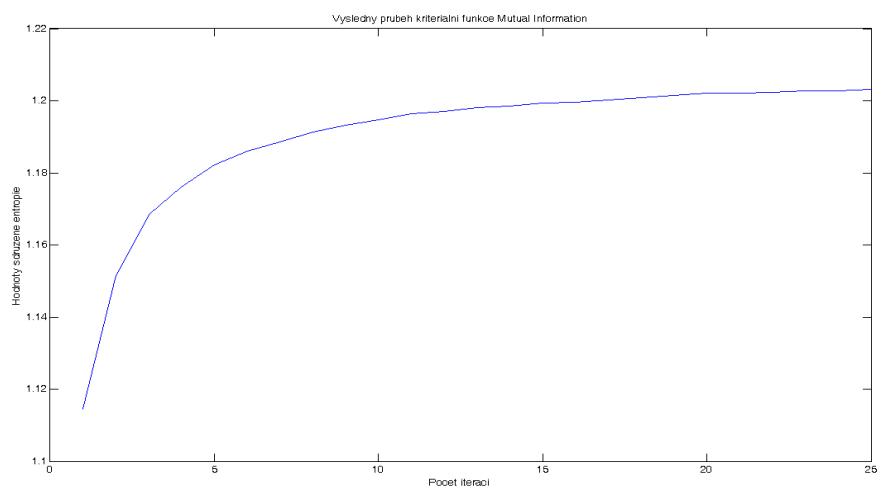
Graf 34. – Průběh podobnostní funkce korelační koeficient (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)



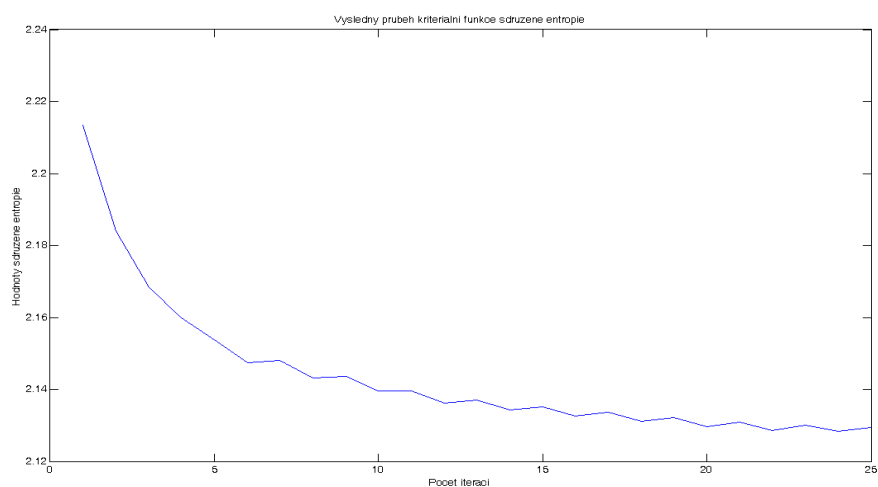
Graf 35. – Průběh podobnostní funkce kosinové kritérium (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)



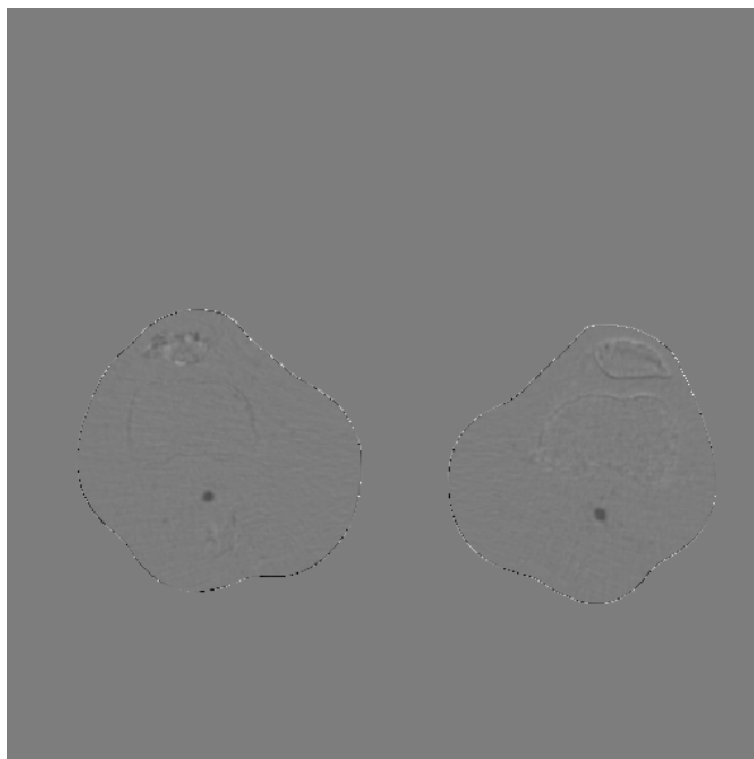
Graf 36. – Průběh podobnostní funkce SSD (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)



Graf 37. – Průběh podobnostní funkce Mutual Information (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)

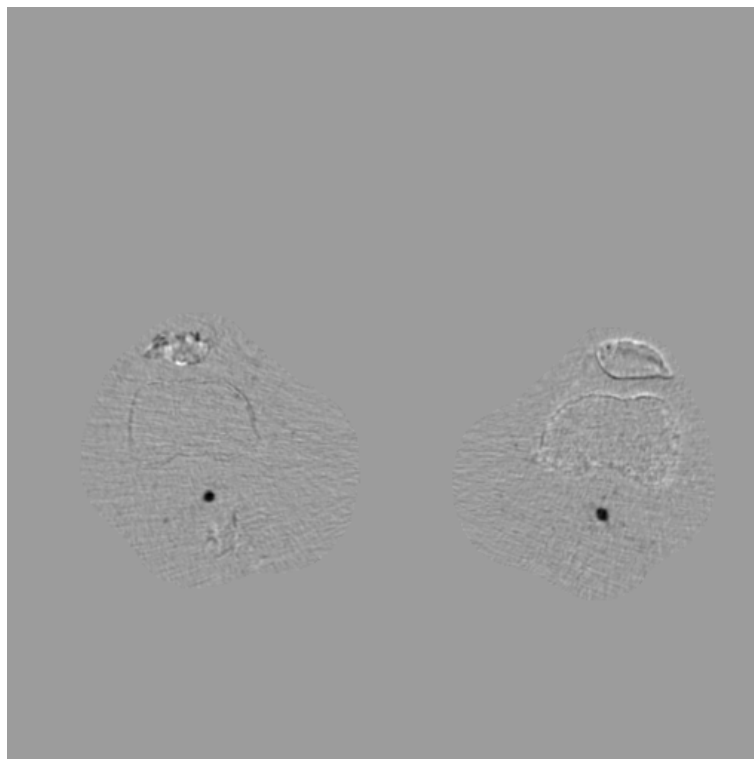


Graf 38. – Průběh podobnostní funkce sdružené entropie (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)



Obrázek 53. – Výsledný rozdílový obraz bez potlačeného artefaktu (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)

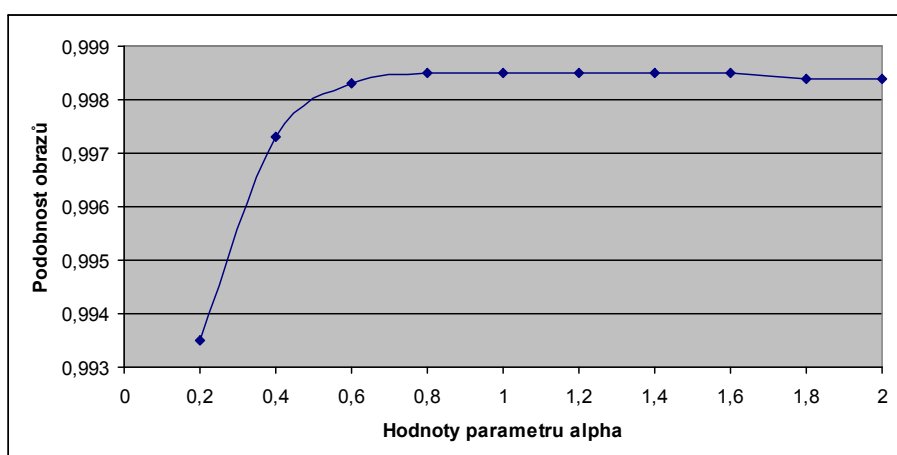




Obrázek 54. – Výsledný rozdílový obraz s potlačeným artefaktem (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)

### Testování automatické formy metody registrace obrazů pomocí algoritmu „Demon“

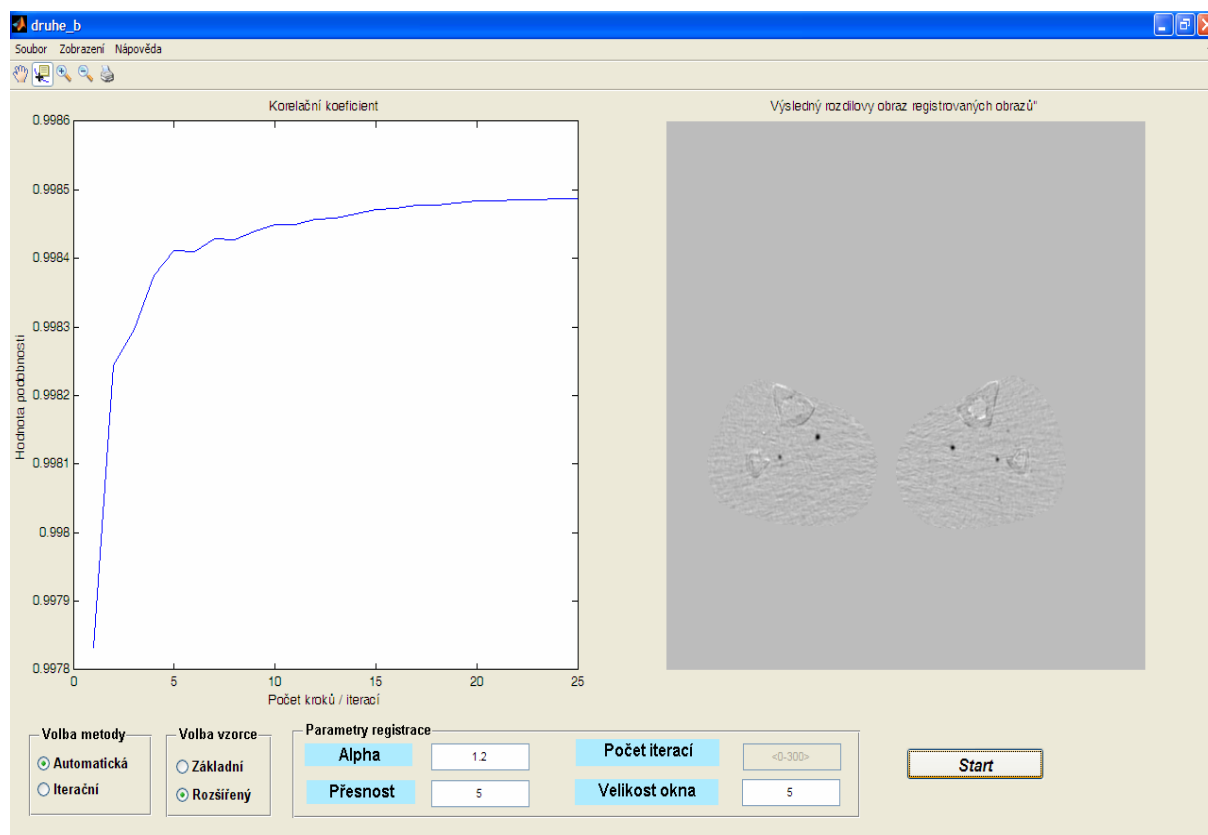
V závěru této kapitoly bude provedeno současně testování funkčnosti vytvořeného grafického rozhraní GUI a automatické formy registrace obrazů. Aby byla prezentována spolehlivost programu, byla zvolena jiná dvojice prekontrastního a postkontrastního obrazu. Popis vytvoření a funkce automatické registrace obrazů byl popsán v předchozích kapitolách. Vyjděme z rozšířené verze vzorce pro výpočet „Demonů“. Nejprve nalezneme nejvhodnější velikost parametru  $\alpha$  vzhledem k podobnosti obrazů.



Graf 39. - Vliv velikosti parametru alpha na výslednou podobnost obrazů

Z předchozího grafu je patrné, že parametr  $\alpha$  je vhodné volit v rozsahu hodnot od 0,8-1,6, neboť v tomto rozsahu dosahujeme největší podobnosti obrazů. Do hodnoty 0,8 opět dochází k většímu zvýraznění kostních struktur. Od hodnoty 0,8 ustupují kostní struktury do pozadí a jsou velmi výrazné cévy.

Při automatickém testu byly zvoleny následující hodnoty parametrů registrace obrazů: *Přesnost registrace* = 5, *Velikost okna* = 5, *alpha* = 1,2. Byly použity obrazy *20Nohy3DPre.mat* a *20Nohy3DPost.mat*.



Obrázek 55. – Výsledný rozdílový obraz s potlačeným artefaktem (automatická registrace,  $\alpha = 1,2$ , Přesnost=5, Velikost okna=5)

Z předchozího obrázku je patrné, že realizovaná automatická forma registrace obrazů je funkční, stejně jako vytvořené grafické uživatelské rozhraní GUI. Výsledná podobnost obrazů byla, dle korelačního koeficientu, cca 99,85 %. V pravé části obrázku je znázorněn rozdílový obraz s řádně potlačeným rozdílovým artefaktem a výraznými cévami.

## 5. Závěr

Cílem této diplomové práce bylo prostudovat problematiku registrace obrazů pomocí metody Optical Flow, pro danou metodu navrhnout program, otestovat jej v programovém prostředí Matlab a vyhodnotit výsledky registrace obrazů na obrazových datech dodaných vedoucím této práce.

V první kapitole bylo provedeno shrnutí základních poznatků, vztahujících se k registraci obrazů. Bylo realizováno obecné schéma registrace obrazů a byly diskutovány jeho jednotlivé bloky. Postupně byla probrána transformace, interpolace, kritériální funkce a na závěr optimalizace registrace obrazů.

Další kapitola byla věnována Optical Flow a algoritmu „Demon“. Byly probrány a diskutovány možnosti implementace algoritmu do výpočtu Optical Flow a realizace registrace obrazů.

Třetí část práce pak byla věnována realizovanému programovému řešení. Byl představen kód programu, spolu s vysvětlením jeho funkce.

Pro registraci reálných medicínských obrazů z CT, které byly dodány vedoucím diplomové práce, byla zvolena poměrně mladá a perspektivní metoda registrace obrazů založená na algoritmu „Demon“, která byla představena v roce 1996 J. P. Thirionem. Metoda byla zvolena z důvodu své flexibility a možnosti využití při registraci reálných medicínských obrazů a to i multimodálních.

Realizovaný program byl vytvořen v programovém prostředí Matlab. Byly vytvořeny celkem dvě varianty procesu registrace obrazů a to iterační nebo automatickou formou. Výsledný kód byl zpracován do přehledného a intuitivního grafického uživatelského rozhraní GUI. Toto rozhraní bylo opět vytvořeno v programovém prostředí Matlab.

V realizovaném programu byla využita bilineární interpolace, které byla zvolena jako určitý kompromis mezi početní náročností a dosaženými výsledky. Dále bylo v programu otestováno celkem sedm kritériálních funkcí. Jednalo se o korelační koeficient, kosinové kritérium, normalizovanou cross-corellation, SSD, Mutual Information, normovanou Mutual Information a sdruženou entropii. Zastavovací kritérium, v realizované automatické formě registrace obrazů, bylo vytvořeno na základě podobnostní funkce Mutual Information, která je primárně určena jako kritériální funkce při registraci multimodálních obrazů. Dodaná obrazová data jsou monomodální, avšak jistou multimodalitu zavádí do obrazů přidáný kontrast. MI byla tedy zvolena z důvodu možnosti využití programu i pro registraci multimodálních obrazů.

V poslední části této práce bylo provedeno ověření funkčnosti realizovaného programu. Pro testování bylo použit PC s procesorem Intel Celeron-M420 1,6GHz a operační paměť 1,49 GB. Nejprve byly iteračně testovány obrazy *Lena*. Poté byla funkčnost testována na reálných obrazech získaných z CT zobrazovací modality. Soubor obrazů k registraci obsahoval prekontrastní a postkontrastní obrazy. Cílem práce bylo dodané obrazy registrovat a poté (odečtením) zobrazit cévy. Jako statický obraz byl nastaven obraz postkontrastní a jako pohybový obraz byl nastaven obraz prekontrastní. Toto nastavení bylo provedeno z toho důvodu, že po odečtení obrazů chceme vidět nezkreslené cévy. V případě, že by bylo nastavení obrazů zaměněno, byly by výsledné, zobrazené cévy zkresleny.

Při odečtení registrovaného a prekontrastního obrazu došlo na hranách výsledného obrazu k vytvoření rozdílového artefaktu, který byl způsoben rozdílnou velikostí odečítaných obrazů.

Vzniklý artefakt byl odstraněn pomocí základní morfologické operace eroze. V realizovaném grafickém rozhraní byla vytvořena možnost prezentace výsledků pomocí rozdílového obrazu (s potlačeným i zachovaným rozdílovým artefaktem), šachovnice a mapy *quiver*.

Z realizovaných testů vyplývá, že je navržený program funkční a je jím možno registrovat běžné i medicínské obrazy. Z grafů je patrné, že mnohem lepší výsledky registrace obrazů poskytoval rozšířený vzorec pro výpočet „Demonů“ (2.27), který při výpočtu využívá gradientů obou registrovaným obrazů a parametr  $\alpha$ .

Z realizovaných testů vyplývá, že využití rozšířeného vzorce snižuje dobu registrace až o polovinu, přičemž dává stejné, nebo i lepší výsledky podobnosti registrovaných obrazů. Dále bylo zjištěno, že volbou parametru  $\alpha$  je možné regulovat šum v obraze. Při nižších hodnotách parametru  $\alpha$  byly zvýrazněny kostní struktury v reálných CT registrovaných obrazech, vyšší hodnoty tyto struktury potlačily a naopak zvýraznily cévy. Tohoto poznatku by bylo možné využít k lepší orientaci při diagnostice snímků.

# Seznam obrázků

- Obrázek 1. – Blokové schéma procesu registrace obrazů  
Obrázek 2. – Příklad prosté translace na jednoduchém objektu (převzato a upraveno z [3])  
Obrázek 3. – Příklad rotace na jednoduchém objektu (převzato a upraveno z [3])  
Obrázek 3. – Změna měřítka jednoduchého objektu (převzato a upraveno z [3])  
Obrázek 4. – Příklad afinní transformace na jednoduchém objektu (převzato a upraveno z [3])  
Obrázek 5. – Příklad projektivní transformace na jednoduchém objektu (převzato a upraveno z [3])  
Obrázek 6. – Příklad rozmístění řídicích bodů a vytvoření oblastí pro pice- wise transformaci (převzato a upraveno z [1])  
Obrázek 7. – Příklad vychýlení od diskrétních bodů obrazové mřížky (převzato a upraveno z [1])  
Obrázek 8. – Jednorozměrné interpolační jádro metody NN (převzato a upraveno z [1])  
Obrázek 9. – Příklad interpolace nejbližší soused (převzato a upraveno z [3])  
Obrázek 10. – Princip bilineární interpolace a její interpolační jádro (převzato z [3])  
Obrázek 11. – Interpolační jádro bikubické interpolace (převzato z [3])  
Obrázek 12. – Příklad sdruženého 2D histogramu (převzato a upraveno z [1])  
Obrázek 13. - Pohybové pole (převzato a upraveno z [7])  
Obrázek 14. - Příklad spojitého pohybového pole (převzato a upraveno z [7])  
Obrázek 15. - Srovnání polí na Barberově rotačním válci (převzato a upraveno z [7])  
Obrázek 16- Zobrazení problému apertury (převzato a upraveno z [7])  
Obrázek 17. – Okamžitá rychlost v obrazech ve statickém a pohybovém obraze (převzato a upraveno z [11])  
Obrázek 18. - Prvotní pokus s „Demony“ na semipermeabilní membráně propouštějící částice dvou různých plynů (převzato a upraveno z [11])  
Obrázek 19. - Základní předpoklad difúzního modelu založeného na „demonech“ (převzato a upraveno z [11])  
Obrázek 20. – Model přitažení mezi disky – 3 iterace (převzato z [11])  
Obrázek 21. – Model difúze mezi disky – 3 iterace (převzato z [11])  
Obrázek 22. – Model přitažení mezi disky – 6 iterací (převzato z [11])  
Obrázek 23. – Model difúze mezi disky – 6 iterací (převzato z [11])  
Obrázek 24. – Základní iterační schéma registraci obrazů na základě difúzních modelů (převzato a upraveno z [11])  
Obrázek 25. – Chování „demonů“ při technice segmentace obrazů (převzato a upraveno z [11])  
Obrázek 26. – Blokové schéma iterační registrace obrazů realizované v programovém prostředí Matlab  
Obrázek 27. – Blokové schéma automatické registrace obrazů realizované v programovém prostředí Matlab  
Obrázek 28. – Kód pro načtení obrazů realizovaný v programovém prostředí Matlab  
Obrázek 29. – Kód pro nastavení parametrů algoritmu při iteračním i automatickém schématu pro registraci obrazů realizovaném v programovém prostředí Matlab  
Obrázek 30. – Kód iterační smyčky registrace obrazů při použití algoritmu „Demon“ realizovaném v programovém prostředí Matlab  
Obrázek 31. – Kód výpočtu sdruženého histogramu a Mutual Information realizovaný v programovém prostředí Matlab  
Obrázek 32. – Definování pomocných proměnných při automatické registraci obrazů v programovém prostředí Matlab  
Obrázek 33. – Začátek smyčky automatické registrace obrazů realizované v programovém prostředí Matlab  
Obrázek 34. – Zastavovací kritérium automatické registrace obrazů realizované v programovém prostředí Matlab  
Obrázek 35. – Realizované grafické uživatelské rozhraní v programovém prostředí Matlab  
Obrázek 36. – Blokové schéma realizované eroze pro odstranění rozdílového artefaktu  
Obrázek 37. – Vzniklý rozdílový artefakt (20 iterací,  $\alpha = 1,5$ )  
Obrázek 38. – Kompletní odstranění rozdílového artefaktu (20 iterací,  $\alpha = 1,5$ )  
Obrázek 39. – Pohybový testovaný obraz pro registraci  
Obrázek 40. – Referenční testovaný obraz pro registraci  
Obrázek 41. – Výsledný registrovaný obraz (základní vzorec, 20 iterací)  
Obrázek 42. – Výsledný registrovaný obraz (základní vzorec, 100 iterací)  
Obrázek 43. – Výsledný registrovaný obraz (rozšířený vzorec,  $\alpha = 1,5$ , 100 iterací)  
Obrázek 44. – Pohybový testovaný obraz pro registraci  
Obrázek 45. – Výsledný registrovaný obraz (základní vzorec, 300 iterací)  
Obrázek 46. – Výsledný registrovaný obraz (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)  
Obrázek 47. – Prekontrastní obraz řezu nohama (soubor 1nohy3DPre.mat)  
Obrázek 48. – Postkontrastní obraz řezu nohama (soubor 1nohy3DPost.mat)  
Obrázek 49. – Výsledný registrovaný obraz (základní vzorec, 50 iterací)

Obrázek 50. – Výsledný rozdílový obraz bez potlačeného artefaktu (základní vzorec, 50 iterací)  
 Obrázek 51. – Výsledný rozdílový obraz s potlačeným artefaktem (základní vzorec, 50 iterací)  
 Obrázek 52. – Výsledný registrovaný obraz (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)  
 Obrázek 53. – Výsledný rozdílový obraz bez potlačeného artefaktu (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)  
 Obrázek 54. – Výsledný rozdílový obraz s potlačeným artefaktem (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)  
 Obrázek 55. – Výsledný rozdílový obraz s potlačeným artefaktem (automatická registrace,  $\alpha = 1,2$ , Přesnost=5, Velikost okna=5)

## Seznam grafů

Graf 1. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 20 iterací)  
 Graf 2. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 20 iterací)  
 Graf 3. – Průběh podobnostní funkce SSD (základní vzorec, 20 iterací)  
 Graf 4. – Průběh podobnostní funkce Mutual Information (základní vzorec, 20 iterací)  
 Graf 5. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 20 iterací)  
 Graf 6. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 100 iterací)  
 Graf 7. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 100 iterací)  
 Graf 8. – Průběh podobnostní funkce SSD (základní vzorec, 100 iterací)  
 Graf 9. – Průběh podobnostní funkce Mutual Information (základní vzorec, 100 iterací)  
 Graf 10. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 100 iterací)  
 Graf 11. – Vliv velikosti parametru alpha na výslednou podobnost obrazů (100 iterací)  
 Graf 12. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 100 iterací)  
 Graf 13. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 100 iterací)  
 Graf 14. – Průběh podobnostní funkce SSD (základní vzorec, 100 iterací)  
 Graf 15. – Průběh podobnostní funkce Mutual Information (základní vzorec, 100 iterací)  
 Graf 16. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 100 iterací)  
 Graf 17. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 300 iterací)  
 Graf 18. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 300 iterací)  
 Graf 19. – Průběh podobnostní funkce SSD (základní vzorec, 300 iterací)  
 Graf 20. – Průběh podobnostní funkce Mutual Information (základní vzorec, 300 iterací)  
 Graf 21. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 300 iterací)  
 Graf 22. – Vliv velikosti parametru alpha na výslednou podobnost obrazů (100, 200, 300 iterací)  
 Graf 23. – Průběh podobnostní funkce korelační koeficient (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)  
 Graf 24. – Průběh podobnostní funkce kosinové kritérium (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)  
 Graf 25. – Průběh podobnostní funkce SSD (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)  
 Graf 26. – Průběh podobnostní funkce Mutual Information (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)  
 Graf 27. – Průběh podobnostní funkce sdružené entropie (rozšířený vzorec,  $\alpha = 0,6$ , 300 iterací)  
 Graf 28. – Průběh podobnostní funkce korelační koeficient (základní vzorec, 50 iterací)  
 Graf 29. – Průběh podobnostní funkce kosinové kritérium (základní vzorec, 50 iterací)  
 Graf 30. – Průběh podobnostní funkce SSD (základní vzorec, 50 iterací)  
 Graf 31. – Průběh podobnostní funkce Mutual Information (základní vzorec, 50 iterací)  
 Graf 32. – Průběh podobnostní funkce sdružené entropie (základní vzorec, 50 iterací)  
 Graf 33. – Vliv velikosti parametru alpha na výslednou podobnost obrazů (25 iterací)  
 Graf 34. – Průběh podobnostní funkce korelační koeficient (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)  
 Graf 35. – Průběh podobnostní funkce kosinové kritérium (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)  
 Graf 36. – Průběh podobnostní funkce SSD (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)  
 Graf 37. – Průběh podobnostní funkce Mutual Information (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)  
 Graf 38. – Průběh podobnostní funkce sdružené entropie (rozšířený vzorec,  $\alpha = 1,6$ , 25 iterací)  
 Graf 39. – Vliv velikosti parametru alpha na výslednou podobnost obrazů

## Seznam zkratk

CT – počítačová tomografie  
 RTG – rentgen  
 NMR – nukleární magnetická rezonance  
 NN – (nearest neighbor) nejbližší soused  
 BL – bilineární

## Seznam příloh

Nápověda k realizovanému programu  
Vytvořené programové prostředí GUI  
Vytvořené funkce registrace obrazů

## Reference

- [1] JIŘÍ JAN: Medical Image Processing, Reconstruction and Restoration Concepts and Methods. Taylor & Francis Group, Boca Raton, 2006, 730s.
- [2] ZITOVÁ B., FLUSSER J.: Image registration methods: a survey, Image and Vision Computing 21 (2003) 977–1000, 2003, 24s.
- [3] BARTOŠ M.: Lícování obrazů lékařských zobrazovacích modalit, Brno: Diplomová práce FEKT VUTBR, 2007, 66 s.
- [4] SCHWARZ D., PROVAZNIK I., Registrace multimodálních medicínských obrazů s využitím Matlabu a ITK [online] Dostupný z WWW: [http://dsp.vscht.cz/konference\\_matlab/matlab04/schvarz.pdf](http://dsp.vscht.cz/konference_matlab/matlab04/schvarz.pdf)
- [5] KUBEČKA L., Optimization methods for image registration [online]. [2005]. Dostupný z WWW: [www.feec.vutbr.cz/EEICT/2003/.../05-kubecka\\_libor.pdf](http://www.feec.vutbr.cz/EEICT/2003/.../05-kubecka_libor.pdf)
- [6] BARRON J.L, FLEET D.J, BEAUCHEMIN S.S.: Performance of Optical Flow Techniques, [online]. [2004] IJCT 12:1 pp43-77 Dostupný z WWW: <http://www.springerlink.com>
- [7] ŘÍHA K.: Pokročilé techniky zpracování obrazu, [elektronické skriptum ]. [ 30.listopadu 2007]. Dostupný z WWW: <https://www.vutbr.cz/elearning/>
- [8] FLEET D.J., WEISS Y. Optical flow estimation [online] Dostupný z WWW: <http://www.cs.toronto.edu/~fleet/courses/cifarSchool09/flowChapter05.pdf>
- [9] BOCO I A., PELICAN E. :Optical flow [online] [červenec 2006]. Dostupný z WWW: [www.ima.ro/mmelsp\\_series/mmelsp\\_08\\_papers/bocoi\\_08.pdf](http://www.ima.ro/mmelsp_series/mmelsp_08_papers/bocoi_08.pdf)
- [10] KROON D.J., SLUMP C.H: MRI modality transformation in demon registration [online] [srpen 2009]. Dostupný z WWW: [ieeexplore.ieee.org/iel5/5174537/5192959/05193214.pdf](http://ieeexplore.ieee.org/iel5/5174537/5192959/05193214.pdf)
- [11] THIRION J.P. : Image matching as a diffusion process: an analogy with Maxwells's demons [online] [1998 Medical image analysis]. Dostupný z WWW: [www.ncbi.nlm.nih.gov/pubmed/9873902](http://www.ncbi.nlm.nih.gov/pubmed/9873902)
- [12] HE Wang, Lei Dong, Jennifer O'Daniel, Radhe Mohan,Adam S Garden, K Kian Ang, Deborah A Kuban, Mark Bonnen,Joe Y Chang a Rex Cheung: Validation of an accelerated

„demons“ algorithm for deformable image registration in radiation therapy [online] [2005 Phys. Med. Biol. 50 (2005) 2887–2905]. Dostupný z WWW: <[www.iop.org/EJ/article/0031-9155/50/12/.../pmb5\\_12\\_011.pdf](http://www.iop.org/EJ/article/0031-9155/50/12/.../pmb5_12_011.pdf)>

[13] CACHIER P., PENNEC X., AYACHE N.: Fast non rigid matching by gradient descent: Study and Improvements of the “Demons” Algorithm [online] [červen 1999 N° 3706 Rapprot de recherche]. Dostupný z WWW: <[ftp://ftp.inria.fr/INRIA/publication/dienst/RR-3706.pdf](http://ftp.inria.fr/INRIA/publication/dienst/RR-3706.pdf)>

[14] Vercauteren T., Pennec X., Perchant A., Ayache N.: Diffeomorphic Demons: Efficient Non-parametric Image Registration [online] [květen 2009 NeuroImage]. Dostupný z WWW: <<http://www-sop.inria.fr/asclepios/Publications/Tom.Vercauteren/DiffeoDemons-NeuroImage08-Vercauteren.pdf>>

[15] MANDUCHI R., TOMASI C.: Distinctiveness Maps for Image Matching [online] [1999]. Dostupný z WWW: <[citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.7711](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.7711)>

[16] HRNČÍŘ Z.: Optický tok v obrazových datech živých buněk [online] [2006 Diplomová práce]. Dostupný z WWW: <[is.muni.cz/th/60514/fi\\_m/dp.pdf](http://is.muni.cz/th/60514/fi_m/dp.pdf)>

[17] MANDUCHI R., TOMASI C.: Distinctiveness Maps for Image Matching [online] [1999]. Dostupný z WWW: <[citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.7711](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.7711)>

[18] DERPANIS K.G.: Relationship between SSD and Cross Correlation for Template Matching [online] [prosinec 2005]. Dostupný z WWW: <[http://www.cse.yorku.ca/~kosta/CompVis\\_Notes/ssd\\_cross-correlation.pdf](http://www.cse.yorku.ca/~kosta/CompVis_Notes/ssd_cross-correlation.pdf)>

[19] ŘÍHA F.: Registrace autofluorescenčních obrazů sítnice [online] [2008 Diplomová práce]. Dostupný z WWW: <<https://www.vutbr.cz/studium/zaverecne-prace>>

[20] JOHNSON H., ZHAO Y.: BRAINSDemonWarp: An Applicaton to Perform Demons Registration [online] [květen 2009]. Dostupný z WWW: <[www.nitrc.org/docman/view.php/151/.../BRAINSDemonWarp.pdf](http://www.nitrc.org/docman/view.php/151/.../BRAINSDemonWarp.pdf)>

[21] CAHILL N. D., NOBLE J.A., HAWKERS D.J.: Demons algorithms for fluid and curvature registration [online] [2009]. Dostupný z WWW: <[portal.acm.org/citation.cfm?id=1700056](http://portal.acm.org/citation.cfm?id=1700056)>

[22] YEO B.T., ET ALL: Spherical Demons: Fast Surface Registration [online] [2010]. Dostupný z WWW: <<http://www-sop.inria.fr/asclepios/Publications/Tom.Vercauteren/SphericalDemons-MICCAI08-Yeo.pdf>>

[23] BRO-NIELSEN : Medical image registration and surgery simulation [online] [1996 Doktorská práce]. Dostupný z WWW: <[www2.imm.dtu.dk/pubdb/views/edoc\\_download.php/227/.../imm227.pdf](http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/227/.../imm227.pdf)>



## Příloha 1)

### *Nápověda k realizovanému programu*

#### Registrace obrazu pomocí metody OPTICAL FLOW

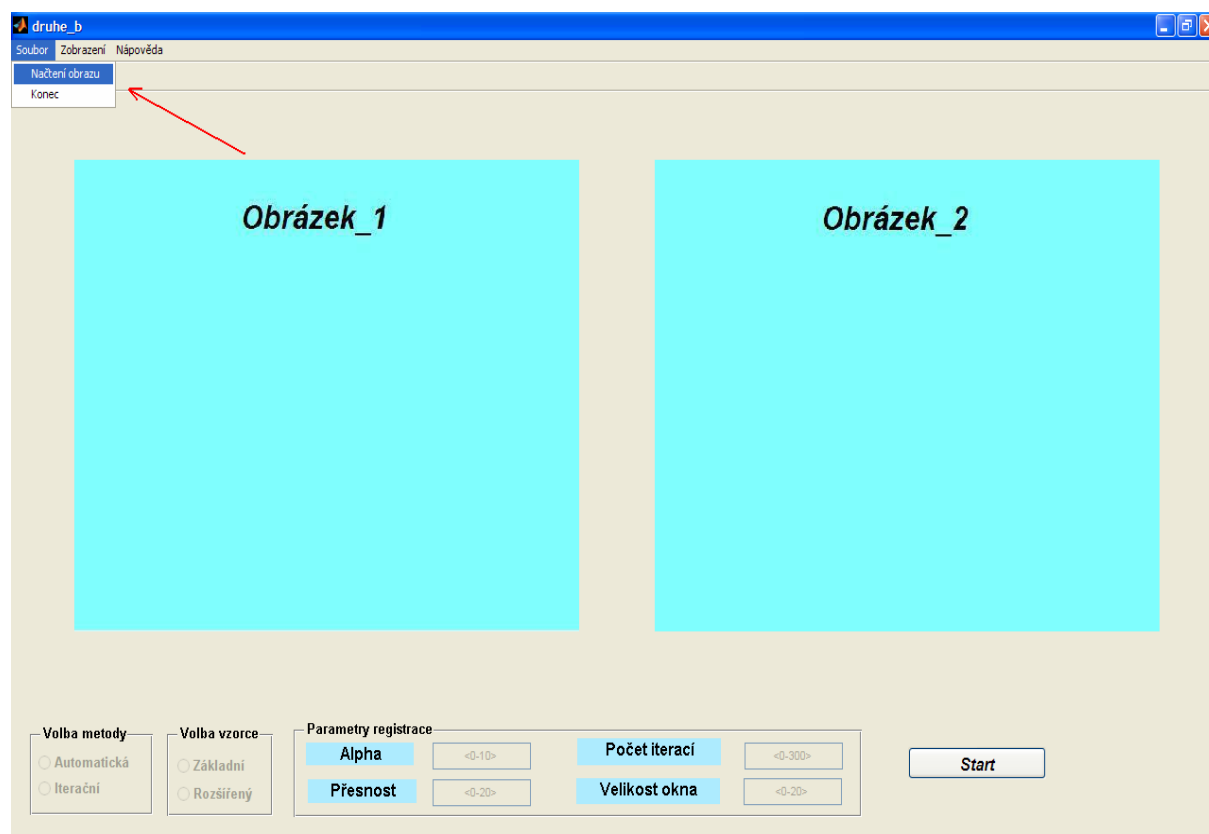
Vytvořený program GUI, pro registraci obrazů, v programovém prostředí Matlab má uživatelsky příznivé a intuitivní ovládání. V následujícím textu bude uveden pouze základní popis možností nastavení příslušných parametrů a zobrazení výsledků registrace obrazů a průběhů podobnostních kritérií.

#### 1.) Načtení obrazů

Realizovaný program umožňuje načtení obrazů následujících formátů - \*.mat, \*.jpg, \*.jpeg, \*.png. V případě, že chceme registrovat dodaná data ve formátu \*.mat, načítáme nejprve prekontrastní obraz a poté obraz postkontrastní. Realizovaný program je možné využít i pro universální registraci obrazů jiných datových formátů, potom při načítání nemluvíme o prekontrastním a postkontrastním obrazu, ale o referenčním a pohybovém obrazu.

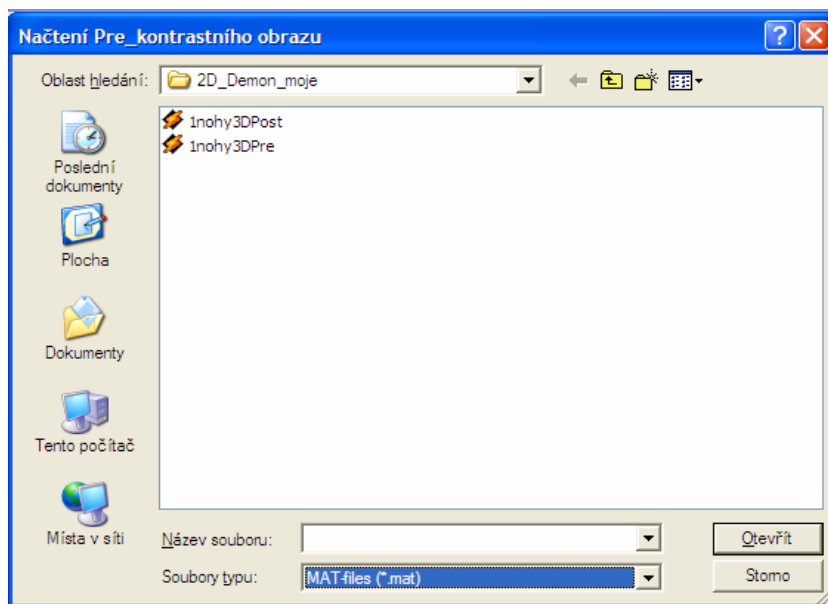
Postup načtení obrazů je následující:

**Soubor => Načtení obrazu**

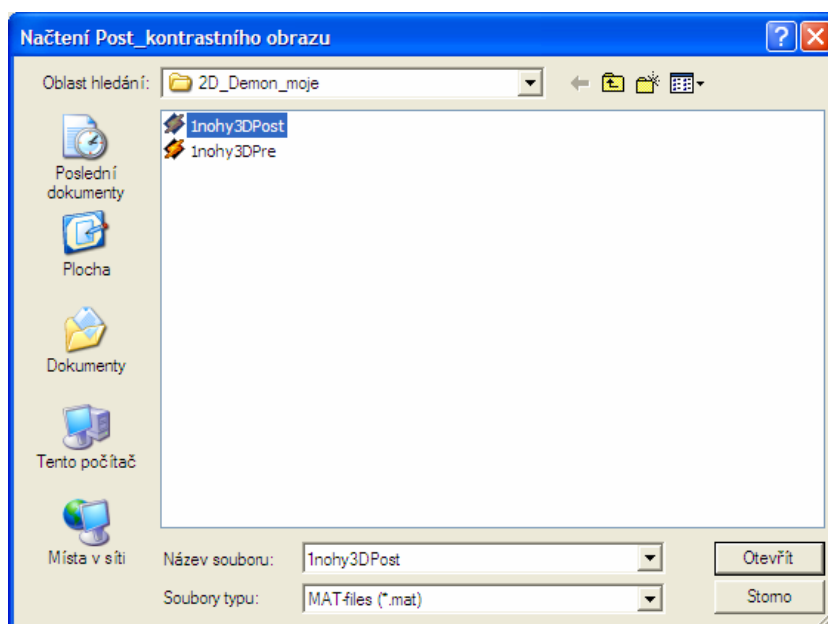


Poté se otevřou následující dialogová okna:

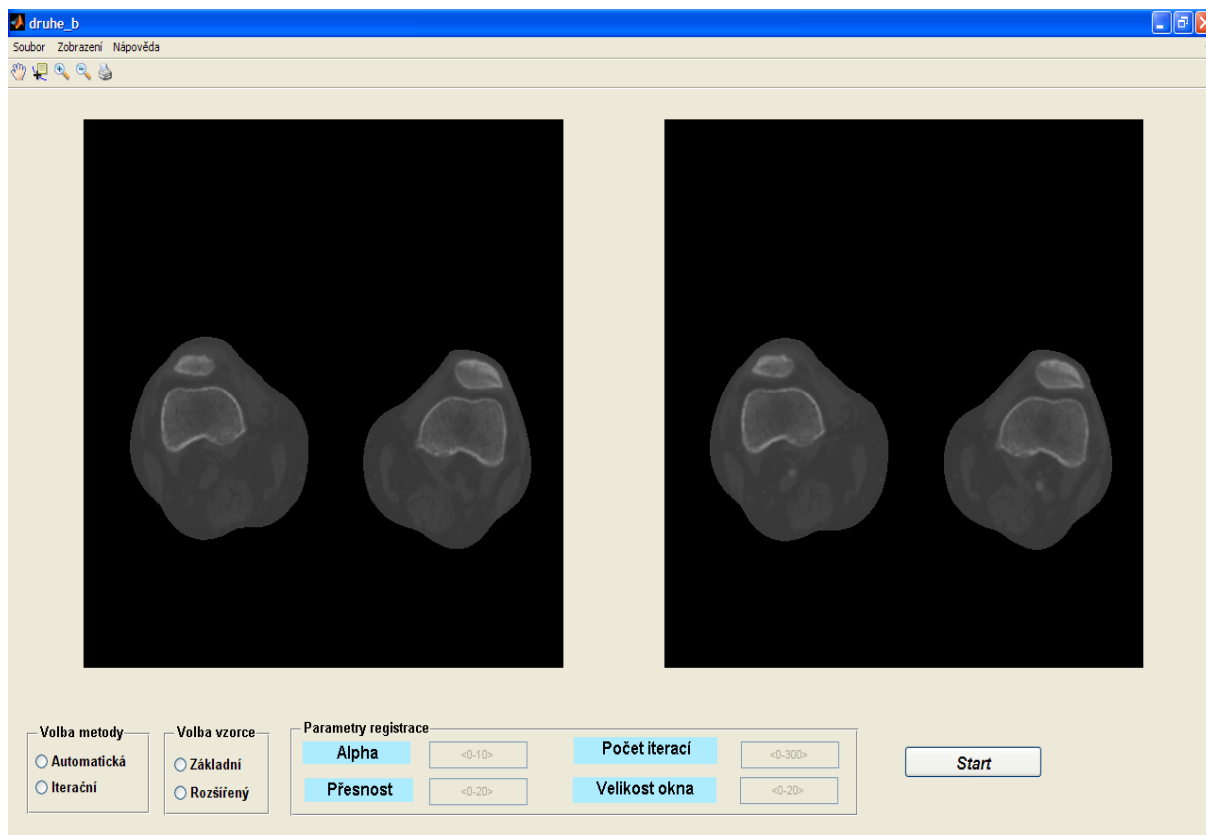
***Načtení prekontrastního obrazu => Načtení postkontrastního obrazu***



Po načtení prekontrastního obrazu se ihned objeví další dialogové okno s možností výběru načtení postkontrastního obrazu.



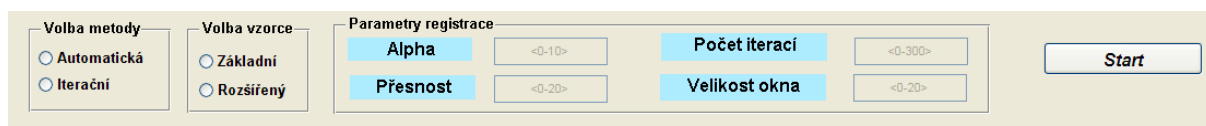
Po úspěšném načtení obrazů, které mají být registrovány, dojde k jejich automatickému zobrazení.



Obrázek nalevo zobrazuje prekontrastní obraz, obrázek napravo postkontrastní obraz.

## 2.) Nastavení parametrů registrace a její spuštění

Po úspěšném načtení obrazů, se ve spodní části okna programu zpřístupní nabídka, ve které je možné volit a zadávat parametry registrace. Z následujícího obrázku je patrné, o jaké možnosti nastavení se jedná.



**Volba metody** => Zde je možné vybrat ze dvou variant a to klasické iterační metody, případně metody automatické (viz popis v příslušné kapitole diplomové práce).

**Volba vzorce** => Proces registrace obrazů je možné provést pomocí dvou vzorců, základního a rozšířeného (viz popis v příslušné kapitole diplomové práce).

**Parametr Alpha** => Využívá se při registraci obrazů rozšířeným vzorcem. Jeho vlastnosti jsou popsány v příslušné kapitole diplomové práce. Je nutné jej zadat v rozsahu <0-10> s možností využití desetinných čísel.

**Počet iterací** => Slouží k nastavení počtu iterací při iterační metodě registrace obrazů. Je nutné jej zadat v celočíselném rozsahu <0-300>.

**Parametr Přesnost** => Jedná se o parametr, kterým se definuje velikost čítače (zastavovací kritérium) při automatické metodě registrace obrazů. Hodnoty, s kterými zde pracujeme, jsou v celočíselném rozsahu **<0-20>**. Parametr **Přesnost** musí být z důvodu funkčnosti realizovaného automatického zastavení registrace obrazů větší nebo roven parametru **Velikost okna**.

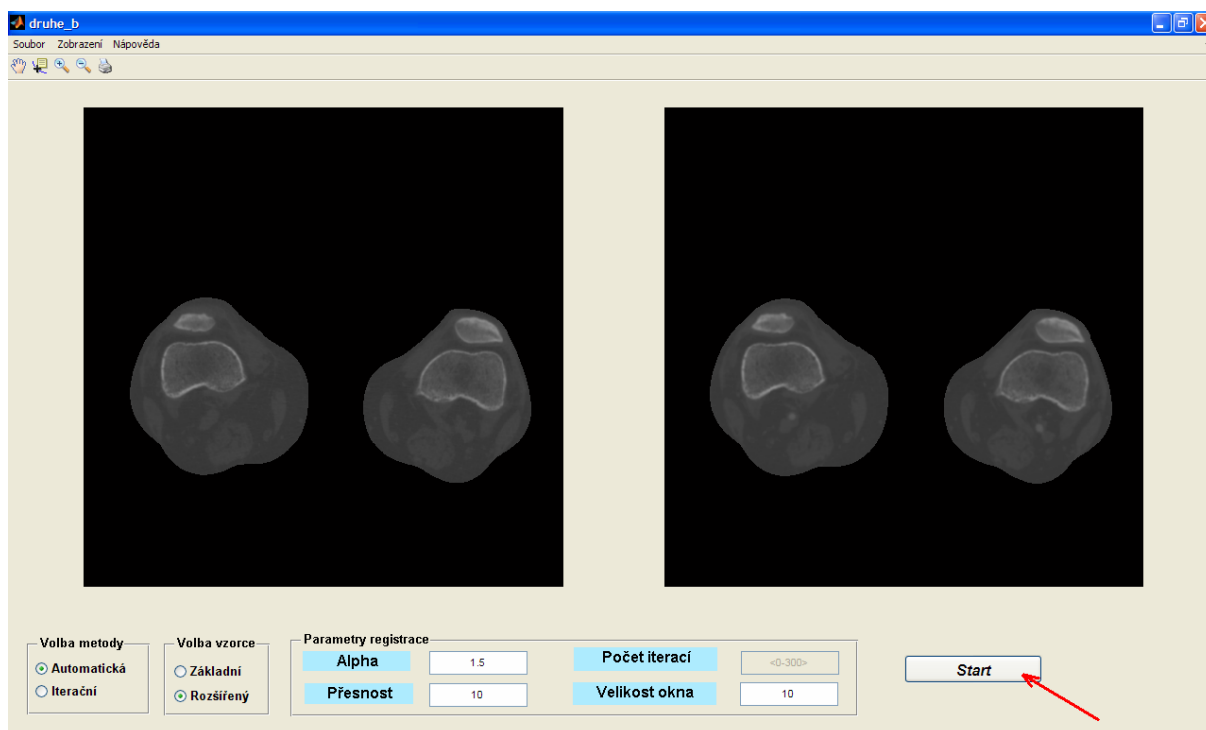
**Parametr Velikost okna** => Jedná se opět o parametr vztahující se k realizované automatické metodě registrace obrazů. Jedná se o plovoucí okno, ze kterého se vyhodnocují maxima průběhu podobnostní funkce (příslušná kapitola diplomové práce). Je-li hodnota parametru např. 5, bude se maximum vyhodnocovat z pěti vzorků průběhu podobnostní funkce. Parametr musí být menší nebo roven parametru **Přesnost** a to v celočíselném rozsahu **<0-20>**.

Po volbě požadované metody a vzorce se zpřístupní příslušná pole, do kterých je nutné zadat hodnoty v odpovídajícím rozsahu a formátu. Příklad je uveden na následujícím obrázku

V případě, že uživatel zadá hodnoty mimo požadovaný rozsah, případně ve špatném formátu, zobrazí se dialogové okno s varováním.

Pokud nejsou k zadány všechny požadované hodnoty a dojde ke kliknutí na tlačítko **Start**, zobrazí se dialogové okno, které uživateli řekne, které hodnoty je třeba doplnit, aby bylo možné spustit registraci obrazů.

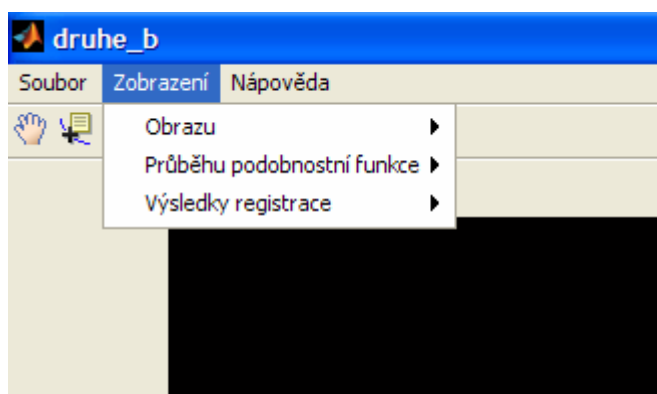
Po zadání všech parametrů registrace (ve vhodném rozsahu a formátu) je možné spustit registraci obrazů, viz následující obrázek.



Po spuštění procesu registrace se na obrazovce objeví dialogové okno, informující o probíhající registraci. Po skončení registrace toto okno zmizí.

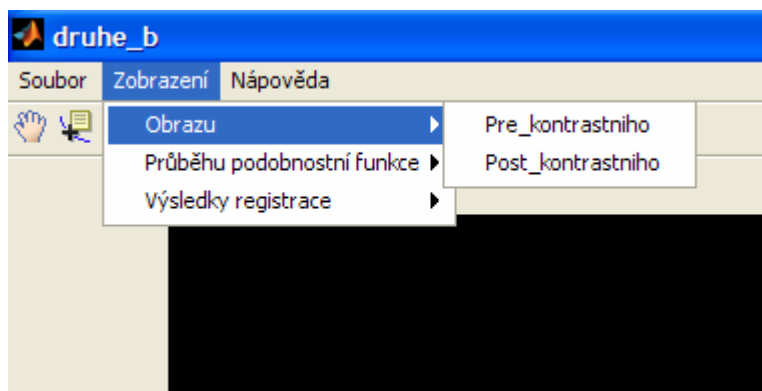
### 3.) Zobrazení průběhů podobnostních funkcí a výsledků registrace

Nyní si ukažme jakým způsobem je možné zobrazit průběhy podobnostních funkcí mezi registrovanými obrazy a obrazové výsledky samotné registrace, případně jak znovu zobrazit původně načtené obrazy.

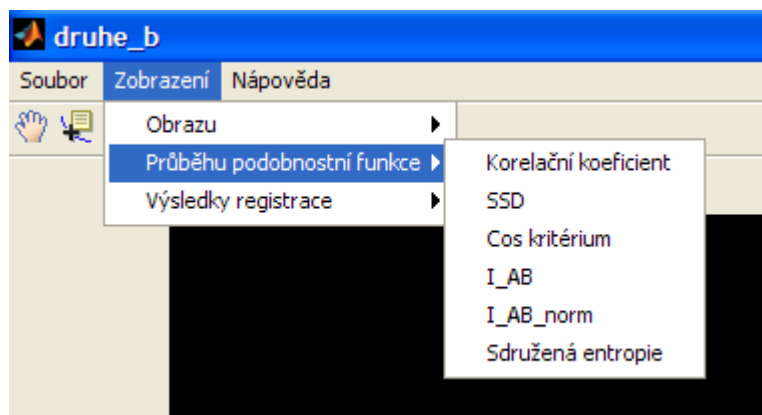


Na obrázku můžeme vidět výše popsané možnosti zobrazení.

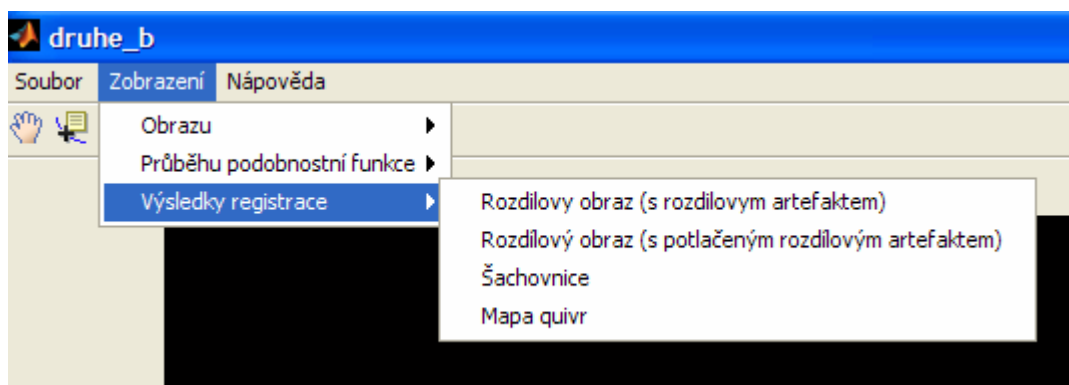
Zobrazení původně načtených obrazů:



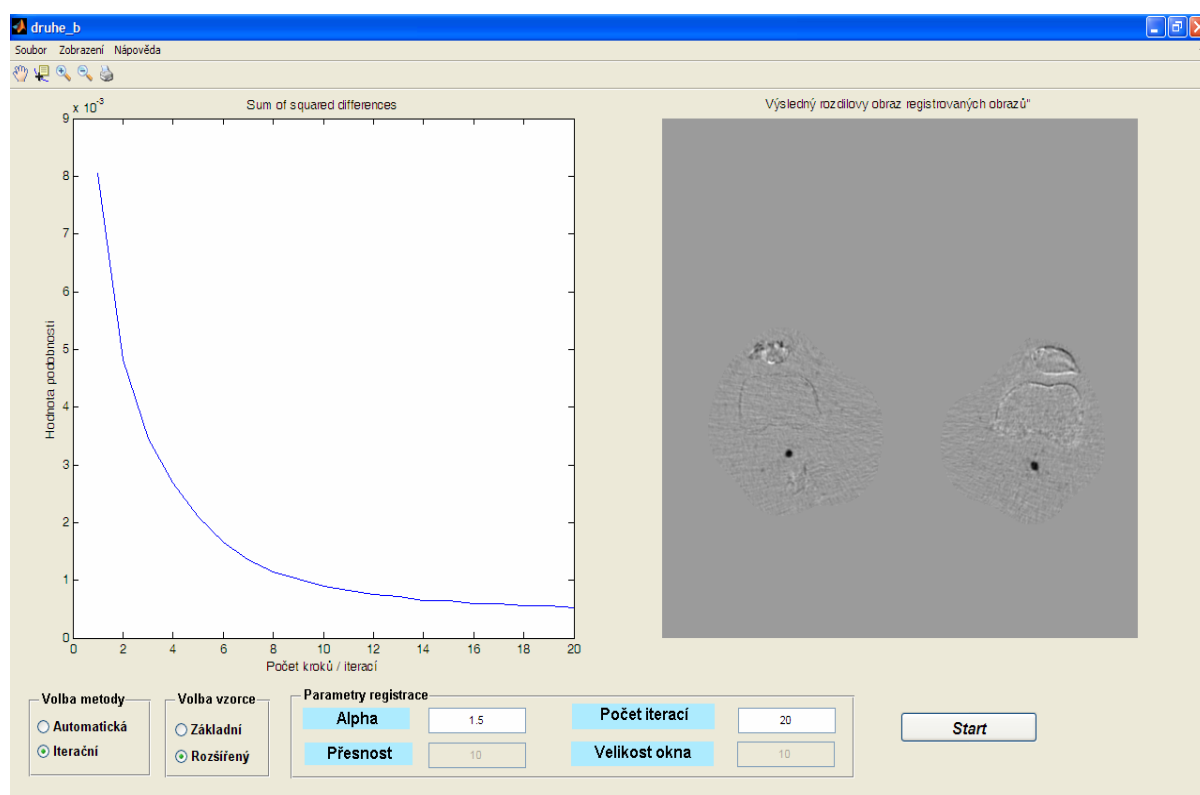
Zobrazení průběhů podobnostních funkcí registrovaných obrazů:



Zobrazení výsledků registrovaných obrazů:



Na následujícím obrázku je zobrazen příklad prezentace výsledků registrace obrazů. Levý obrázek představuje průběh podobnostní funkce SSD a pravý obrázek znázorňuje výsledný obraz po potlačení rozdílového artefaktu.



#### 4.) Nápořěda

Položka v menu s názvem *Nápořěda* slouží k zobrazení tohoto souboru.

#### 5.) Reset programu

Při každém načtení nových obrazů se program se nastaví do defaultních hodnot.