



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ
Katedra biomedicínské techniky

**Zpracování MRI snímků v kardiologii (MRI tagging)
pomocí metod využívajících Fourierovu transformaci**

**MRI image data processing in cardiology (MRI
tagging) via methods based on the Fourier transform**

Bakalářská práce

Studijní program: Biomedicínská a klinická technika

Studijní obor: Biomedicínský technik

Autor bakalářské práce: Stanislav Štrba

Vedoucí bakalářské práce: doc. Ing. Jiří Hozman, Ph.D.

Kladno 2017

Katedra biomedicínské techniky

Akademický rok: 2016/2017

Z a d á n í b a k a l á ř s k é p r á c e

Student: **Stanislav Štrba**
Obor: Biomedicínský technik
Téma: **Zpracování MRI snímků v kardiologii (MRI tagging) pomocí metod využívajících Fourierovu transformaci**
Téma anglicky: MRI image data processing in cardiology (MRI tagging) via methods based on the Fourier transform

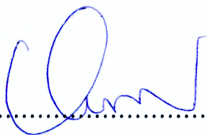
Z á s a d y p r o v y p r a c o v á n í :

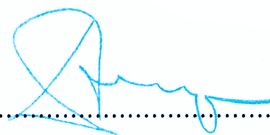
Pro zadané sekvence MRI obrazových dat (MRI tagging) navrhnete a realizujete načtení těchto dat ve formátu DICOM. Dále navrhnete a realizujete extrakci pohybového vektoru v rámci celého obrazu, resp. v oblasti zájmu (ROI) a též extrakci obrysů levé komory. Jako výsledek uceleného řešení zobrazte grafy tzv. příčného a torzního namáhání srdečního svalu. Vše realizujte na MRI snímcích s použitím přídatného gradientu v podobě mřížky a zaměřte se na metody využívající Fourierovu transformaci (algoritmus HARP a jeho zdokonalené verze).

Seznam odborné literatury:

- [1] El-Sayed H Ibrahim, Myocardial tagging by Cardiovascular Magnetic, *Journal of Cardiovascular Magnetic Resonance*, ročník 13, číslo 36, 2011, Červenec, 1-40 s., <http://www.jcmr-online.com/content/13/1/36>, doi:10.1186/1532-429X-13-36
- [2] Amir A. Amini, Jerry L. Prince (eds.), *Measurement of Cardiac Deformations from MRI: Physical and Mathematical Models*, Kluwer Academic Publishers, 2001, 320 s., ISBN 1-4020-0222-X
- [3] Pleva M., Ouředníček P., *MRI srdce - Praktické využití z pohledu kardiologa*, ed. 1, Grada, 2012, 144 s., ISBN 978-80-247-3931-1
- [4] NEMA et al., *The DICOM Standard*, 30.3.2005, [Revidováno 31.12.2015], [Citováno 8.1.2017], <http://dicom.nema.org/standard.html>

Zadání platné do: 11.09.2018
Vedoucí: doc. Ing. Jiří Hozman, Ph.D.
Konzultant: MUDr. Theodor Adla, KZM FN Motol


.....
vedoucí katedry / pracoviště


.....
děkan

V Kladně dne 20.02.2017

PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci s názvem „*Zpracování MRI snímků v kardiologii (MRI tagging) pomocí metod využívajících Fourierovu transformaci*“ vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně 19. 5. 2017

.....

Stanislav Štrba

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu této práce, doc. Ing. Jiřímu Hozmanovi, Ph.D., za výborné vedení a trpělivost během celého procesu vzniku mé bakalářské práce. Dále bych rád poděkoval Klinice zobrazovacích metod 2. LF UK a FN Motol, zejména MUDr. Theodoru Adlovi za poskytnutí snímků a výsledků z projektu AZV ČR, č. 15-31398A s názvem Charakteristiky elektromechanické dyssynchronie predikující efekt srdeční resynchronizační terapie, který vznikl z iniciativy doc. MUDr. Lucie Riedelbachové, Ph.D. z Kardiologické kliniky 2. LF UK a FN Motol, bez které by tento projekt vůbec nevznikl a proto také jí patří moje díky.

ABSTRAKT

Zpracování MRI snímků v kardiologii (MRI tagging) pomocí metod využívajících Fourierovu transformaci

Ve své bakalářské práci se zabývám zpracováním kardiovaskulárních snímků z magnetické rezonance s přídavným gradientem, takzvaným MRI tagging. Práce je součástí studie Kardiologické kliniky 2. LF UK a FN Motol, která se zabývá léčbou srdeční dyssynchronie. Ta se projevuje nerovnoměrnou aktivitou částí levé komory. Cílem je tvorba programu, který by byl nápomocný v rozhodovacím procesu, zdali je pacient vhodným kandidátem pro podstoupení srdeční resynchronizační terapie, CRT. Program v MatLabu je schopen načíst data ve formátu DICOM. Je schopen zároveň pracovat s transversálními i longitudinálními snímky. Pomocí Fourierovy transformace program dokáže zrekonstruovat pohyblivou mřížku a na základě její analýzy může exportovat pohybový vektor, počítat příčnou deformaci, tzv. radial strain a zároveň torzní deformaci, tj. circumferential strain. Vypočítaná data jsou reprezentována v podobě grafů mechanické deformace levé komory, ze kterých lze odečíst rozdíly v deformaci segmentů levé komory a tak diagnostikovat poruchu převodního systému srdečního. Bylo zjištěno, že pro tuto metodu nejsou vhodné longitudinální řezy.

Klíčová slova

Kardiovaskulární zobrazování magnetickou rezonancí, MRI tagging, Fourierova transformace, segmentace levé komory srdce, srdeční dyssynchronie

ABSTRACT

MRI image data processing in cardiology (MRI tagging) via methods based on the Fourier transform

Bachelor's thesis deals with the processing of cardiovascular images from magnetic resonance with an additional gradient, so-called MRI tagging. Thesis is part of project currently running at Cardiology Clinic of the 2nd Faculty of Medicine of the Charles University and Motol University Hospital, which deals with the treatment of cardiac dyssynchrony. This is manifested by the uneven activity of parts of the left ventricle. The aim is to create a program that would assist in the decision making process whether the patient is a suitable candidate for cardiac resynchronization therapy, CRT. The MatLab program is able to load data in DICOM format. It is also capable of working with both transversal and longitudinal sequences. Using the Fourier Transform, the program can reconstruct the motion. Based on its analysis, it can export the motion vector, compute the transversal deformation, the so-called radial strain, and the circumferential strain. The calculated data are represented in the graphs of left ventricular mechanical deformation from which the differences in deformation of the left ventricular segments it is possible to diagnose cardiac conduction system disorder. In conclusion, this method is not suitable for longitudinal sections computation.

Keywords

Cardiovascular magnetic resonance imaging, MRI tagging, Fourier transform, segmentation of the left ventricle of the heart, heart dyssynchrony

Obsah

Seznam symbolů a zkratk.....	9
1 Úvod.....	12
1.1 Přehled současného stavu.....	12
1.1.1 CRT – Srdeční resynchronizační léčba, srdeční dyssynchronie.....	12
1.1.2 Segmentace levé komory podle modelu AHA.....	13
1.1.3 Magnetická rezonance s přídavným gradientem.....	15
1.1.4 Současné používaný software v dané problematice.....	16
1.1.4.1 HARP.....	16
1.1.4.2 Medviso Segment.....	16
1.1.4.3 OsiriX / Harmonic Phase Flow plugin.....	16
1.1.4.4 cvi42.....	17
1.2 Cíle práce.....	17
2 Metody.....	18
2.1 Tvorba syntetických dat.....	18
2.2 Načtení snímků magnetické rezonance.....	19
2.2.1 Formát DICOM.....	19
2.2.2 Úprava kontrastu.....	21
2.2.3 Načtení metadat.....	22
2.3 Výběr ROI.....	22
2.3.1 ROI transversálních řezů.....	22
2.3.2 ROI longitudinálních řezů.....	24
2.3.3 Zvětšení a oříznutí snímků.....	24
2.3.4 Výběr os a bodů longitudinálních řezů.....	25
2.4 Fourierova transformace a návrh filtrů.....	26
2.4.1 2D Fourierova transformace a amplitudové spektrum.....	26
2.4.2 Tvorba filtrů pro rekonstrukci mřížky.....	27
2.4.3 Další filtry pro následující operace.....	28
2.4.4 Zpětná Fourierova transformace.....	28
2.5 Extrakce obrysů levé komory.....	29
2.5.1 Extrakce obrysů transversálních řezů.....	29

2.5.2	Extrakce obrysů longitudinálních řezů	31
2.6	Segmentace podle modelu AHA	32
2.6.1	Segmentace transversálních masek	32
2.6.2	Segmentace longitudinálních masek	34
2.7	Rekonstrukce mřížky.....	36
2.8	Průniky mřížky	38
2.8.1	Souřadnice průniků mřížky	38
2.8.2	Seskupení odpovídajících průniků	39
2.8.3	Odstranění prázdných pozic v matici	39
2.9	Extrakce pohybového vektoru.....	40
2.10	Deformace levé komory	40
2.10.1	Graf příčné deformace	41
2.10.2	Graf torzní deformace.....	41
3	Výsledky	44
3.1	Výsledky analýzy syntetických dat	44
3.2	Pohybový vektor.....	49
3.3	Porovnání grafů zdravého a nemocného	50
4	Diskuse.....	54
5	Závěr	55
	Seznam použité literatury	56
	Příloha A: Syntetická data k analýze	58
	Příloha B: Obsah přiloženého CD.....	61

Seznam symbolů a zkratek

Seznam zkratek

Zkratka	Význam
AZV ČR	Agentury pro zdravotnický výzkum ČR
MRI	Zobrazení magnetickou rezonancí (<i>Magnetic resonance imaging</i>)
EKG	Elektrokardiografie
CMR	Kardiovaskulární zobrazení magnetickou rezonancí (<i>Cardiovascular magnetic resonance imaging</i>)
AHA	American Heart Association
SA	Krátká osa (Short axis)
LA	Dlouhá osa (Long axis)
2CH	Řez protínající 2 dutiny myokardu (Two chamber)
3CH	Řez protínající 3 dutiny myokardu (Three chamber)
4CH	Řez protínající 4 dutiny myokardu (Four chamber)
SPAMM	Spatial Modulation of Magnetization
HARP	Harmonic Phase
DICOM	Digital Imaging and Communications in Medicine
ROI	Oblast zájmu (<i>Region of interest</i>)
FFT	Rychlá Fourierova transformace (Fast Fourier Transform)
KZM	Klinika zobrazovacích metod 2. LF UK a FN Motol

Seznam obrázků

Obr. 1.1: Segmentace levé komory podle modelu AHA, prezentace řezů [4].....	18
Obr. 1.2: Schéma všech segmentů levé komory [3]	18
Obr. 1.3: Postupné snižování viditelnosti mřížky v průběhu srdečního. [6].....	18
Obr. 2.1: Syntetické snímky.....	18
Obr. 2.2: Dialogové okno pro kód pacienta.....	18
Obr. 2.3: Porovnání výběru složky na obou operačních systémech	20
Obr. 2.4: porovnání původního a upraveného snímku s histogramy	22
Obr. 2.5: Výběr ROI a středu levé komory transversálního řezu	18
Obr. 2.6: Výběr ROI longitudinálního řezu 4CH	24
Obr. 2.7: Určení osy a bodů levé komory longitudinálního řezu 2CH.....	25
Obr. 2.8: Porovnání amplitudového spektra původního a zvětšeného ROI.....	26
Obr. 2.9: Vyznačené pozice pro filtraci frekvenční složky amplitudového spektra, výsledný filtr a amplitudové spektrum po aplikaci filtru.....	27
Obr. 2.10: Výsledky zpětné Fourierovy transformace.....	18
Obr. 2.11: Waitbar	29
Obr. 2.12: Vyznačení obrysů levé komory s naznačenou maskou vnitřního obrysu.....	30
Obr. 2.13: Výběr polygonu naznačujícího obrys levé komory a transparentní maska ...	31
Obr. 2.14: Definice lateroseptální osy	32
Obr. 2.15: Masky všech apikálních segmentů	33
Obr. 2.16: Segmentace podle modelu AHA	35
Obr. 2.17: Detekce hran a prahování	36
Obr. 2.18: Zrekonstruovaná mřížka v jenom směru	36
Obr. 2.19: Zrekonstruovaná mřížka prezentovaná na původním snímku.....	37
Obr. 2.20: Průniky mřížky po aplikaci masky.....	38
Obr. 2.21: Ideální podoba pohybového vektoru [19].....	40
Obr. 2.22: Uspořádání kolmic pro pozorování torzní deformace.....	42
Obr. 3.1: Graf relativní vzdálenosti od středu pro bazální řez.....	44
Obr. 3.2: Graf relativní vzdálenosti od středu pro střeň řez	45
Obr. 3.3: Graf relativní vzdálenosti od středu pro apikální řez	45
Obr. 3.4: Graf příčné deformace, radial strain bazálního řezu	46

Obr. 3.5: Graf příčné deformace, radial strain středního řezu	46
Obr. 3.6: Graf příčné deformace, radial strain apikálního řezu	47
Obr. 3.7: Graf torzní deformace, circumferential strain bazálního řezu	47
Obr. 3.8: Graf torzní deformace, circumferential strain středního řezu	48
Obr. 3.9: Graf torzní deformace, circumferential strain apikálního řezu	48
Obr. 3.10: Pohybový vektor nemocného v bazálním řezu	49
Obr. 3.11: Pohybový vektor nemocného v středním řezu	49
Obr. 3.12: Pohybový vektor nemocného v apikálním řezu	50
Obr. 3.13: Grafy relativní vzdálenosti od středu pro bazální řez, vlevo zdravý, vpravo nemocný	50
Obr. 3.14: Grafy příčné deformace pro bazální řez, vlevo zdravý, vpravo nemocný	51
Obr. 3.15: Grafy torzní deformace pro bazální řez, vlevo zdravý, vpravo nemocný	51
Obr. 3.16: Grafy relativní vzdálenosti od středu pro střední řez, vlevo zdravý, vpravo nemocný	52
Obr. 3.17: Grafy příčné deformace pro střední řez, vlevo zdravý, vpravo nemocný	52
Obr. 3.18: Grafy torzní deformace pro střední řez, vlevo zdravý, vpravo nemocný	52
Obr. 3.19: Grafy relativní vzdálenosti od středu pro apikální řez, vlevo zdravý, vpravo nemocný	53
Obr. 3.20: Grafy příčné deformace pro apikální řez, vlevo zdravý, vpravo nemocný	53
Obr. 3.21: Grafy torzní deformace pro apikální řez, vlevo zdravý, vpravo nemocný	53
Obr. A.1: Sekvence syntetických snímků v bazálním řezu	58
Obr. A.2: Sekvence syntetických snímků v středním řezu	58
Obr. A.3: Sekvence syntetických snímků v apikálním řezu	59
Obr. A.4: Sekvence syntetických snímků v dvoukomorovém řezu	59
Obr. A.5: Sekvence syntetických snímků v tříkomorovém řezu	60
Obr. A.6: Sekvence syntetických snímků v čtyřkomorovém řezu	60

1 Úvod

Tato bakalářská práce se zabývá zpracováním a analýzou dat z magnetické rezonance v kardiologii. Námět této práce vznikl jako součást projektu AZV ČR, č. 15-31398A s názvem Charakteristiky elektromechanické dyssynchronie predikující efekt srdeční resynchronizační terapie probíhajícího na Kardiologické klinice 2. LF UK a FN Motol pod vedením doc. MUDr. Lucie Riedelbachové, Ph.D. Zabývá se diagnostikou u pacientů s chronickým srdečním selháním. Tuto nemoc je možné léčit pomocí srdeční resynchronizační terapie. Právě za tímto účelem je potřeba vytvořit program, který by dokázal usnadnit radiologům a kardiologům rozhodování, zdali je pacient vhodný pro podstoupení této léčby. Program analyzuje data s přídatným gradientem, mřížkou která se deformuje v závislosti na pohybujícím se srdečním svaly. Jelikož se jedná o velmi pravidelnou mřížku, je k její analýze vhodné použít Fourierovu transformaci. Takto získaná data je pak vhodné prezentovat ve srovnávacích grafech, kde lze pozorovat mechanickou deformaci každého segmentu myokardu v čase. Právě rozdíly mezi konkrétními segmenty můžou napomoci určení správné volby terapie pro pacienta.

1.1 Přehled současného stavu

V současné době je diagnostickým standardem monitorování elektrické dyssynchronie a to zejména pomocí EKG. Analýza mechanické deformace může vést k přesnějším výsledkům v oblasti mechanické dyssynchronie.

1.1.1 CRT – Srdeční resynchronizační léčba, srdeční dyssynchronie

Srdeční resynchronizační léčba (cardiac resynchronization therapy, CRT) je jedna z možností léčby srdeční dyssynchronie, poruchy koordinované činnosti srdečních komor, u pacientů s chronickým srdečním selháním, u kterých není účinná běžná farmakologická léčba. Dyssynchronie se může projevovat jako porucha koordinace stahů mezi levou a pravou komorou, a/nebo mezi jednotlivými stěnami jedné z komor. Elektrická dyssynchronie může být způsobena zpomalením elektrické aktivace komory, která pak vyústí v dyssynchronii mechanickou.

V současné době se k diagnostice využívá zejména EKG vyšetření, konkrétně doba trvání QRS komplexu a celkový tvar EKG křivky. Toto vyšetření je velmi dobře použitelné u pacientů s blokáci levého Tawarova raménka. U jiných typů poruch ale nemusí být rozdíl v čase aktivace komor dobře patrný. Šířka QRS komplexu sice ukazuje celkovou dobu trvání aktivace komor, ale již nevypovídá nic o způsobech deformací a jejich časových charakteristikách. Zejména u pacientů s velmi úzkým QRS komplexem je zobrazení velmi individuální a nepřesné. Přesnější výsledky mohou být získány za pomoci elektro-anatomického CARTO mapování, které je ale velmi časově náročné,

invazivní a nemůže být použito u každého pacienta. Z toho důvodu se k vyšetřování používá i přes svou nepřesnost EKG, které je neinvazivní a časově nenáročné.

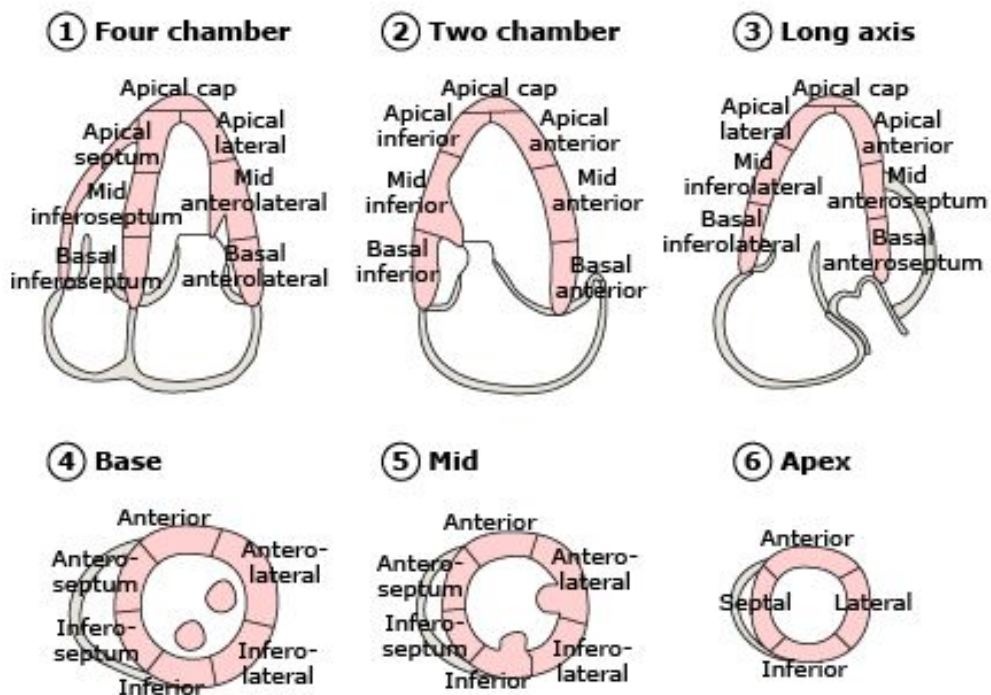
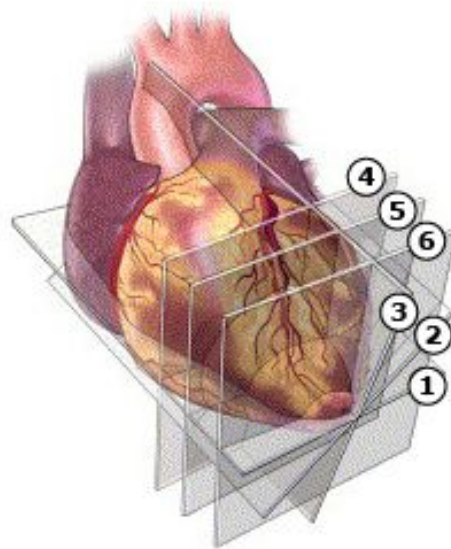
Další možností zjišťování mechanické dyssynchronie je vyšetření pomocí MRI. Na sekvencích snímků lze dobře pozorovat změny na stěnách srdečních komor v průběhu času. Nevýhodou této metody je její časová a finanční náročnost a zejména to, že ji nelze praktikovat u pacientů s již implantovanými kardiostimulátory či defibrilátory. V současné době také bohužel neexistují studie, které by zaznamenávaly mechanickou i elektrickou dyssynchronii zároveň. Ve výše uvedeném projektu je snaha současně zobrazit výsledky jak ze systému Carto, tak i z CMR [1].

Srdeční resynchronizační léčba je indikována pacientům se srdečním selháním, systolickou dysfunkcí levé komory (projevuje se ejekční frakcí nižší než 35 %) , anebo u pacientů s poruchou nitrokomorového vedení vzruchu (projevem je QRS komplex delší než 130 ms). Provádí se za pomoci specializovaného kardiostimulátoru, jehož jedna elektroda se implantuje do pravé komory. Druhá se poté implantuje na povrch komory levé, a to žilním systémem. Alternativně lze elektrodu na povrch levé komory našít. V některých případech lze přidat ještě třetí elektrodu do pravé síně, která pomáhá k součinnosti síní a komor. Kardiostimulátor má za úkol obnovit správnou funkci srdečního svalu tím, že se elektrický signál šíří ze dvou (tří) různých míst a tím je stimulace efektivnější. Dochází k významnému zúžení QRS komplexu. Léčba je účinná zhruba u 70 % pacientů. Procento pacientů, kterým tato léčba nepomáhá je vysoké právě kvůli velmi individuálním projevům poruch na křivce EKG a tím ztížené diagnostice [2].

1.1.2 Segmentace levé komory podle modelu AHA

V roce 2002 vydala American Heart Association, zkráceně AHA, článek ve kterém jeho autoři standardizovali nomenklaturu a segmentaci tomografických řezů myokardu. Model segmentace levé komory je velmi podstatný pro tuto práci, protože výsledný program pracuje pro každý segment podle modelu AHA samostatně.

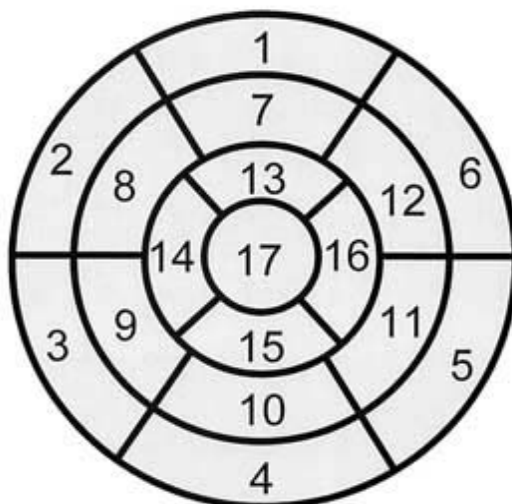
Myokard byl rozdělen do 6 řezů. Tři transversálních, tzv. short-axis – SA (krátká osa) a tři longitudinálních řezů, tzv. long-axis – LA (dlouhá osa). Transversální řezy se označují podle pozice, ve které byl daný řez, resp. Snímek, pořízen. Nejbazálnější řez se označuje Base (na Obr. 1.1 označen číslem 4), střední řez se označuje Mid (na Obr. 1.1 označen číslem 5) a nejapikálnější řez se označuje Apex (na Obr. 1.1 označen číslem 6). Longitudinální řezy se označují podle počtu dutin, které daný řez protíná. Řez procházející všemi čtyřmi dutinami myokardu se označuje 4CH, neboli four chamber (na Obr. 1.1 označen číslem 1), řez procházející dvěma dutinami se označuje 2CH, nebo two chamber (na Obr. 1.1 označen číslem 2). Řez, který protíná 3 dutiny se označuje LA, neboli long axis (dlouhá osa), nicméně se můžeme zejména v různých softwarech setkat i s označením 3CH, čili three chamber (na Obr. 1.1 označen číslem 3) [3].



Obr. 1.1: Segmentace levé komory podle modelu AHA, prezentace řezů [4].

Tyto řezy se následně segmentují na více částí podle pozice oblasti, kterou v rámci levé komory reprezentují. První 2 transversální řezy jsou segmentovány na 6 částí, apikální řez je rozdělen pouze na 4 části. Každý longitudinální řez je segmentován na 7 částí. Celkový počet segmentů je 17 a můžeme je pozorovat na schématu v Obr. 1.2. Každý segment je pojmenován podle anatomické pozice ve které se nachází [3]. Nejlépe se orientuje podle septa myokardu, čili hranice mezi levou a pravou komorou, která je zpravidla otočená k sternu. Pro další aplikace je tedy lateroseptální osa dost podstatná.

Left Ventricular Segmentation



- | | | |
|------------------------|-----------------------|---------------------|
| 1. basal anterior | 7. mid anterior | 13. apical anterior |
| 2. basal anteroseptal | 8. mid anteroseptal | 14. apical septal |
| 3. basal inferoseptal | 9. mid inferoseptal | 15. apical inferior |
| 4. basal inferior | 10. mid inferior | 16. apical lateral |
| 5. basal inferolateral | 11. mid inferolateral | 17. apex |
| 6. basal anterolateral | 12. mid anterolateral | |

Obr. 1.2: Schéma všech segmentů levé komory [3].

1.1.3 Magnetická rezonance s přidavným gradientem

Pro posouzení lokální kontraktility myokardu lze použít i tzv. mřížkovou sekvenci (REST GRID či SPAMM). Jedná se o řadu saturujících pulzů (režim MZ zobrazování IR), které vytvoří obraz mřížky v rovině sběru včetně lokálního označení myokardu. Z deformací původních čtverců mřížky, tj. i vlastní tkáně, lze pak sledovat lokální pohyb myokardu v rámci srdečního cyklu, posuzovat tedy jeho kontrakci a relaxaci. [5]



Obr. 1.3: Postupné snižování viditelnosti mřížky v průběhu srdečního [6].

Na Obr. 1.3 lze pozorovat, jak v průběhu srdečního cyklu klesá intenzita mřížky. Ve většině případů je ke konci sekvence mřížka téměř neviditelná, z čehož vyvstávají nejzávažnější problémy při zpracování snímků s přídavným gradientem.

1.1.4 Současně používaný software v dané problematice

Diagnostice mechanické dyssynchronie se věnuje řada klinicky používaných softwarů. Většina z nich však nepracuje s přídavným gradientem ani Fourierovou transformací. Následuje stručný výčet nejčastěji využívaných programů.

1.1.4.1 HARP

HARP je na MatLabu založený program vyvinutý IACL (Image analysis and communications lab) na Johns Hopkins Universty. Je určen ke zpracování pohybů analýzou tagged snímků z magnetické rezonance. Vychází z Harmonic Phase Algorithm, který vytvořili v roce 1999 Nael F. Osman a Jerry L. Prince. Pomocí HARPu lze kromě snímků srdce zpracovávat také snímky jazyka.

Ke zpracování dat se používají spektrální maxima Fourierova prostoru tagged snímku z MRI. Spektrální maxima se získávají pásmovou propustí. Z nich se získávají fázové snímky pomocí inverzní Fourierovy transformace. Odtud také pochází název programu – Harmonic phase = HARP images. Mapuje pohyb bodů v čase a počítá s tím, že nejbližší bod na dalším snímku je bodem ze snímku předchozího. Zaznamená pohyby ve vertikálním i horizontálním směru a zobrazuje je ve formě pole 2D vektorů, zobrazujících pohyb každého bodu v daném snímku. Běžná chyba zobrazení je 1 pixel [7].

1.1.4.2 Medviso Segment

Segment je obsáhlý software na analýzu medicínských obrazových dat. Vyvinuli jej společně firma Medviso AB a Lund Cardiac MR Group na Lund University ve Švédsku. Je určen pro výzkumné účely, a pro nekomerční účely je za podmínky citace dostupný zdarma. Lze jej využít nejen pro zpracování dat kardiiovaskulární MRI, ale také pro srdeční CT, SPECT průtoku krve myokardem a pro široký okruh dalších radiologických a kardiologických snímků. Je dostupný v samostatné verzi, i ve verzi pro spuštění v MatLabu, a je možné jej upravovat [8].

1.1.4.3 OsiriX / Harmonic Phase Flow plugin

Harmonic Phase Flow je software vyvinutý pouze pro výzkumné účely, není schválen pro klinické využití. Počítá odhady pohybu na snímcích MRI a deformaci stěny myokardu. Při výpočtu zpracovává tagged MRI snímky za pomoci Gabor filtrace. Nejedná se o samotný program, nýbrž o doplňkový plugin k programu OsiriX, který

je dostupný pro Mac. Osirix je vyvíjen švýcarskou společností Pixmeo. Harmonic Phase Flow software byl vyvinut španělskou skupinou Computer Vision Center pro nemocnici Sant Pau v Barceloně [9].

1.1.4.4 cvi42

Tento program je profesionálním produktem od kanadské společnosti Circle Cardiovascular Imaging. Kromě snímků z magnetické rezonance zpracovává také snímky z CT. Narozdíl od předchozích software zpracovává snímky bez přídavného gradientu. Umí analyzovat deformace srdeční stěny spolu s analýzami žívej a otoků, a kromě toho má i mnoho dalších funkcí, například výpočty průtoků kontrastní tekutiny, analýza T1 a T2 relaxačních časů, zobrazení zúžení v tepnách a detekce kornatění tepen [10].

1.2 Cíle práce

Cílem této práce je tvorba programu v programovém prostředí MatLab, který dokáže zpracovat a vyhodnotit snímky z magnetické rezonance ve formátu DICOM. Je potřebné realizovat načtení dat do programu v tomto formátu, tak aby bylo se snímky dále možné pracovat.

Podstatnou součástí řešení je extrakce obrysů levé komory. Tento úkol je nutné realizovat manuálně, čili umožnit obsluze vyznačit okraje levé komory a ty následně využít při další segmentaci snímků.

Jelikož se program zpracovává snímky s přídavným gradientem, pro rekonstrukci mřížky je potřeba využití Fourierovy transformace. Z takto extrahované mřížky je potřeba získat pozici průniků, které odpovídají mechanické deformaci v daném segmentu.

Jako prezentace by měl sloužit graf příčné a torzní deformace myokardu. Tyto grafy by měly odlišit deformaci různých segmentů navzájem a tak odhalit případnou mechanickou dyssynchronii.

Program je potřeba vyzkoušet na syntetických a zároveň reálných datech. Podstatou je porovnat výsledky zdravého a nemocného subjektu. K tomu by měla sloužit syntetická data a následné ověření na sérii reálných dat.

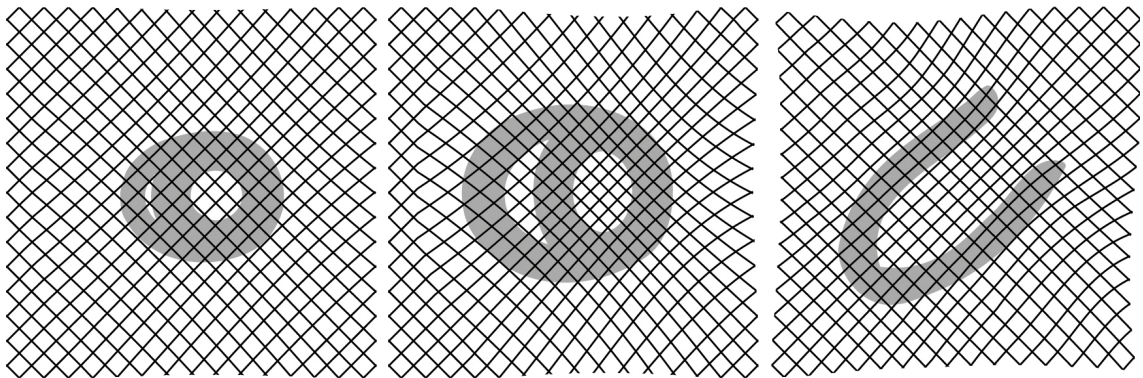
2 Metody

Pro zpracování kardiologických MRI snímků byl v programovém prostředí MatLab vytvořen skript, který je schopen najednou zpracovat celou sérii snímků vztaženou k jednomu pacientovi. Program je vytvořen tak, aby všechny výsledky zůstaly načteny v programovém prostředí MatLab pro další zpracování, porovnání výsledků různých subjektů a zpětnou prezentaci. Nicméně během celého procesu výpočtu je vytvořeno velké množství proměnných a dílčích výsledků, které musí být před zpracováním další série dat smazány.

Tento program pracuje jak pro syntetická, tak pro reálná data z magnetické rezonance. Jedinou podmínkou je, že snímky musí obsahovat přídavný gradient, čili MRI tagged snímky.

2.1 Tvorba syntetických dat

Za účelem testování funkčnosti programu bylo potřeba vytvořit sérii snímků, které by simulovali stah myokardu s přídavným gradientem. Od reálných dat se tyto snímky liší zejména v tom, že nedochází ke ztrátě kontrastu mřížky. Nedochází tak k ztrátě pozic průníků rekonstruované mřížky jak je tomu u dat reálných. Dalším rozdílem, je nepřítomnost torzního pohybu, co by se mělo projevit minimální, nebo ideálně nulovou torzní deformací. Pro ilustraci je na Obr. 2.1 uvedeno několik příkladů syntetických snímků.



Obr. 2.1: Syntetické snímky.

Snímky byly vytvořeny v programu Adobe Illustrator. Díky vektorové povaze obrázku mřížka zůstává kvalitní i po reformaci. Do formátu DICOM byly snímky převedeny v programu Adobe Photoshop.

2.2 Načtení snímků magnetické rezonance

První část skriptu se věnuje jejich úpravě. Z důvodu schopnosti programu zpracovat více sérií patientských snímků je jako první potřeba vymazat proměnné, které mohly být, a pravděpodobně jsou, uloženy v programovém prostředí MatLab z předešlého zpracování. Tomu odpovídají následné příkazy:

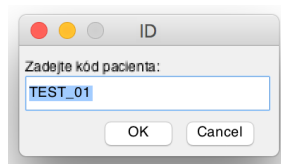
```
% smazání dočasných proměnných
clear obsah_slozky;
clear TG_SA1_cesty TG_SA1_metadata TG_SA1_snimky_EDIT TG_SA1_snimky_RAW;
clear TG_SA2_cesty TG_SA2_metadata TG_SA2_snimky_EDIT TG_SA2_snimky_RAW;
clear TG_SA3_cesty TG_SA3_metadata TG_SA3_snimky_EDIT TG_SA3_snimky_RAW;
clear TG_2CH_cesty TG_2CH_metadata TG_2CH_snimky_EDIT TG_2CH_snimky_RAW;
clear TG_3CH_cesty TG_3CH_metadata TG_3CH_snimky_EDIT TG_3CH_snimky_RAW;
clear TG_4CH_cesty TG_4CH_metadata TG_4CH_snimky_EDIT TG_4CH_snimky_RAW;
```

Příkaz `clear` zajistí odstranění všech dočasných proměnných potřebných pro načtení a úpravy další série obrazových dat. Odstraněné proměnné a to, čemu odpovídají, bude vysvětleno dále v textu.

Pro označení série snímku odpovídajících jednomu pacientovi je potřeba vložit kód pacienta. Tomu odpovídá dialogové okno, které vyzve obsluhu programu k zadání kódu pacienta:

```
% Zadání kódu pacienta
dlg_title = 'ID';
patient_ID = inputdlg('Zadejte kód pacienta:',dlg_title,1,{ 'TEST_01' });
```

Proměnná `dlg_title` odpovídá nápisu v liště dialogového okna. Defaultně je do dialogového okna vepsáno `TEST_01` a je úlohou obsluhy, aby nápis změnila na jméno, resp. kód pacienta. Označení nesmí obsahovat mezery a interpunkci, protože ním budou označeny výsledné matice uložené v programovém prostředí MatLabu. Výzva k vložení kódu pacienta je ilustrována na Obr. 2.2. Kód pacienta je následně uložen jako proměnná `patient_ID`.



Obr. 2.2: Dialogové okno pro kód pacienta.

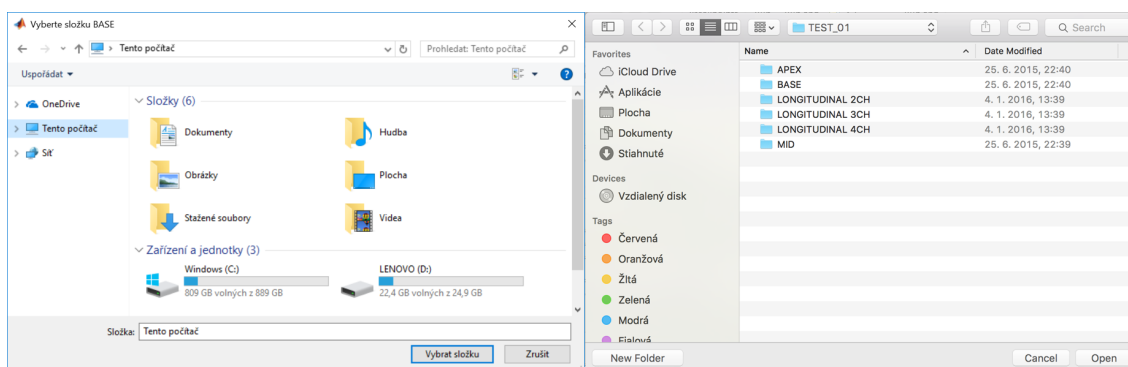
2.2.1 Formát DICOM

DICOM, nebo Digital Imaging and Communications in Medicine je datový formát, který je využíván pro komunikaci a management informací zobrazovacích technik v medicíně a tomu odpovídajících dat [11].

Další částí skriptu je tedy samotné načtení dat ve formátu DICOM. Snímky jednoho pacienta obsahují 6 sekvencí odpovídajících různým řežům srdce. Konkrétně řežům BASE, MID, APEX, 2CH, 3CH a 4CH. Tyto řezy byly podrobněji popsány

v kapitole 1.1.2 Segmentace myokardu podle modelu AHA. Každé sekvenci odpovídá složka obsahující konkrétní na sebe navazující snímky. Jelikož celá série snímků jednoho pacienta obsahuje také snímky, které neobsahují přídatný gradient a jsou označeny pouze náhodně generovanými čísly, bylo potřeba tyto snímky vytřídit a adekvátně pojmenovat pro další zpracování. Tento výběr byl prováděn pomocí programu OsiriX, který má nejvhodnější programové prostředí s ideálním popisem každého snímku a možností dohledání dané složky v počítači. V konečném důsledku tedy složka jednoho pacienta obsahuje 6 podsložek pojmenovaných BASE, MID, APEX, LONGITUDINAL 2CH, LONGITUDINAL 3CH a LONGITUDINAL 4CH.

Pro výběr složky obsahující konkrétní sekvenci byl použit příkaz `uigetdir`, který otevře okno s výběrem dané složky. V liště tohoto okna je instrukce, která složka je aktuálně požadována. K výběru všech sekvencí jednoho pacienta dochází hned po sobě. Zde se objevuje problém při obsluze programu v MatLabu na operačním systému Mac OS, kde chybí lišta s instrukcí, pro kterou složku je výběr požadován. Porovnání výběru složky na operačních systémech Windows a Mac OS můžeme pozorovat na Obr. 2.3.



Obr. 2.3: Porovnání výběru složky na obou operačních systémech.

Proto je důležité dbát na to, že výběr složek po sobě následuje v pořadí BASE, MID, APEX, LONGITUDINAL 2CH, LONGITUDINAL 3CH a LONGITUDINAL 4CH. Tento problém v Mac OS by šlo například řešit využitím GUI (Graphic User Interfaces).

Celému načtení a úpravě jedné sekvence odpovídá následující část skriptu:

```
% Výběr složek, uložení snímků do matice a jejich úprava, načtení metadat
dlg_title = 'Vyberte složku BASE';
cesta = uigetdir('',dlg_title);
obsah_slozky = dir(strcat(cesta, '/'));
obsah_slozky = obsah_slozky(~strncmpi('.', {obsah_slozky.name}, 1));
TG_SA1_cesty = strcat(strcat(cesta, '/'),transpose({obsah_slozky.name}
for i=1:length(TG_SA1_cesty)
    TG_SA1_snimky_RAW{i}=dicomread(char(TG_SA2_cesty(i)));
    % uložení původního snímku do matice
    TG_SA2_snimky_EDIT{i}=imadjust(TG_SA1_snimky_RAW{i},stretchlim(TG_SA1_snim
    ky_RAW{i}),[]); % úprava kontrastu
    TG_SA1_metadata{i}=dicominfo(char(TG_SA1_cesty(i))); % načtení metadat
end
```

Po výběru složky se vytvoří seznam všech souborů, které se v dané složce nacházejí pomocí příkazu `dir`. Jedná se o výčet celého obsahu složky včetně skrytých souborů a systémových složek. Tyto soubory jsou nežádoucí a způsobovaly problémy při načtení dat. Tento problém byl ošetřen odstraněním všech „dotfiles“ neboli souborů ve tvaru „.“ nebo „.“ pomocí příkazu `~strncmpi`. Následně muselo být otestováno, jestli skript funguje pro soubory, které jsou ve tvaru „Název.dcm“ které odpovídají koncovce formátu DICOM. U nich se problém neobjevil. Většina snímků poskytnutých KZM je bez koncovky, takže zde se problém neobjevil vůbec.

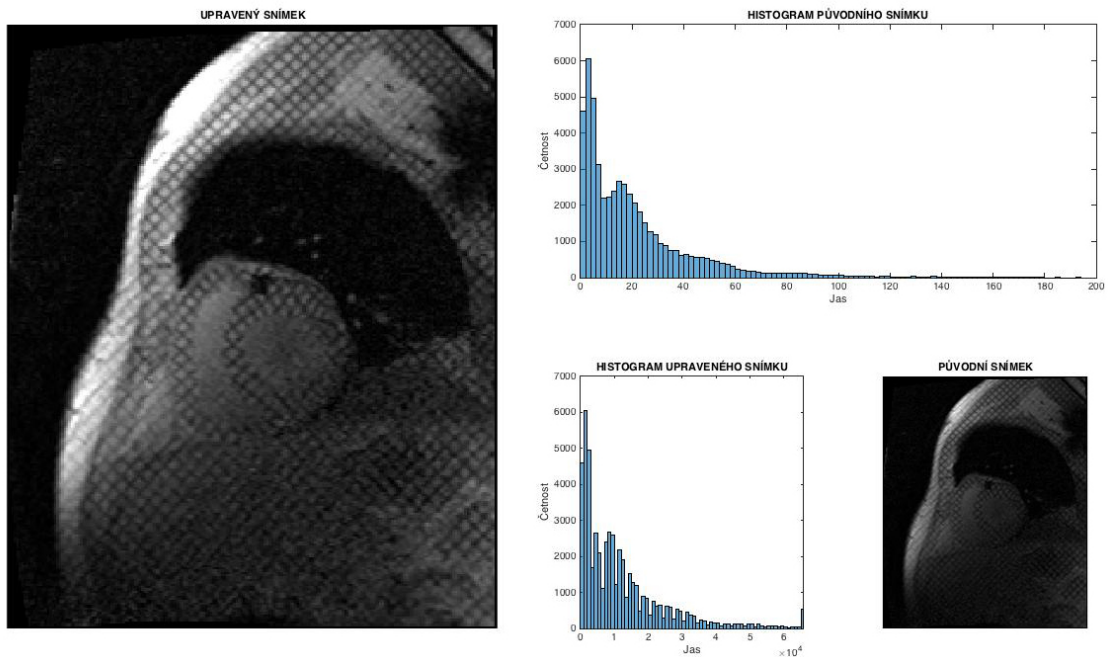
Proměnná `TG_SA1_cesty` obsahuje adresy všech konkrétních snímků dané sekvence v počítači a pomocí příkazu `dicomread` je všechny načte do programu pod názvem `TG_SA1_snimky_RAW` s odpovídajícím číslem pořadí v sekvenci.

Názvy těchto matic odpovídají: TG – jedná se o tagged MRI snímky, SA1 – snímek je transversální řez v pozici BASE a RAW – jedná se o neupravené snímky.

Proces se obdobně opakuje pro zbylých 5 složek se sekvencemi pacienta.

2.2.2 Úprava kontrastu

Z důvodu potřebného zásahu obsluhy do snímků je nevyhnutné upravit kontrast všech snímků tak, aby byly struktury levé komory viditelné a lehce rozeznatelné. Tato úprava je realizována nástroji z Image Processing Toolbox™, konkrétně příkazy `imadjust` a `stretchlim`, který provádí takzvaný „Histogram stretching“. Tato funkce přidělí 1% nejsvětlejších a 1% nejtmavších bodů hodnoty `maxim` a `minim` jasu a hodnoty mezi nimi adekvátně rozdělí, při čemž dojde k zvýšení kontrastu daného snímku [12]. Zároveň dojde k převodu škály 256 odstínů šedé na snímku na škálu 65536 odstínů šedé na snímku. Porovnání původního a upraveného snímku je uvedeno na Obr. 2.4. Pro lepší vizualizaci jsou připojeny histogramy původního a upraveného snímku, kde je zjevné, že nejsvětlejším bodům upraveného snímku je přidělena stejná hodnota jasu.



Obr. 2.4: porovnání původního a upraveného snímku s histogramy.

2.2.3 Načtení metadat

Metadata jsou informace o daném snímku uloženy ve formátu DICOM. Nejsou to obrazová data, nicméně jsou obsažena ve stejném souboru jako konkrétní obrázek. Jedná se o množství standardizovaných dat, které obsahují informace o přístroji, na kterém byl snímek pořízen, stejně jako informace vztahy k celkové studii, pacientovi, sekvenci a konkrétnímu snímku [13]. Pro další potřebu byla metadata každého snímku uloženy pomocí příkazu `dicominfo`.

2.3 Výběr ROI

ROI nebo Region of interest, doslovně přeloženo oblast zájmu, je výseč snímku, která je pro další výpočet důležitá. Důvodem pro zpracovávání ROI místo celého snímku je, že zpracování celého snímku by bylo časově a výkonově velice náročné a na výsledek by nemělo žádný vliv. Proto je definice ROI jako oblasti vymezující výhradně oblast myokardu nezbytná.

2.3.1 ROI transversálních řezů

Všechny transversální řezy u jednoho pacienta mají podobné vlastnosti. Zejména stejné přiblížení a natočení os. Z toho důvodu je u nich vhodné využít stejných parametrů. Jedním z těchto společných parametrů je velikost ROI. Je evidentní, že největší rozměr bude mít levá komora u bazální sekvence, proto se velikost ROI bude odvíjet od tohoto rozměru. Dalším důležitým parametrem pro všechny 3 řezy je společný střed, konkrétně střed levé komory. Volbě těchto parametrů odpovídají následujících částech skriptu:

```

% Rozměry ROI transversálních řezů
TG_SA1_ROI=TG_SA1_snimky_EDIT;
TG_SA2_ROI=TG_SA2_snimky_EDIT;
TG_SA3_ROI=TG_SA3_snimky_EDIT;
fig=imshow(TG_SA1_snimky_EDIT{1}, []);
set(gcf, 'units','normalized','outerposition',[0 0 1 1], 'Name','Výběr ROI', 'Number Title','off');
title('Vymezte okraje levé komory a výběr ROI potvrďte dvojklikem');
[TG_SA1_ROI{1,1}, coordROI_SA]=imcrop(fig);
close all;
ROIradius=ceil(max([coordROI_SA(3) coordROI_SA(4)])*0.7);

```

Pro výpočet požadovaného rozměru je využita funkce `imcrop`, jejímž výstupem jsou počáteční souřadnice a následně šířka a výška daného výběru. Právě z těchto dvou hodnot je dopočítán výsledný rozměr ROI. Cílem je, aby byl ROI čtverec a jeho rozměr byl 1,4krát větší, než větší z dvojice rozměrů vymezujících přesně levou komoru. Proto byl použit příkaz `max`, a po následném výběru a vynásobení koeficientem uložen parametr `ROIradius`, který odpovídá poloměru ROI. Pozice středů levé komory u všech tří transversálních řezů musí být zadána ručně a to konkrétně v prvním snímku z každé série.

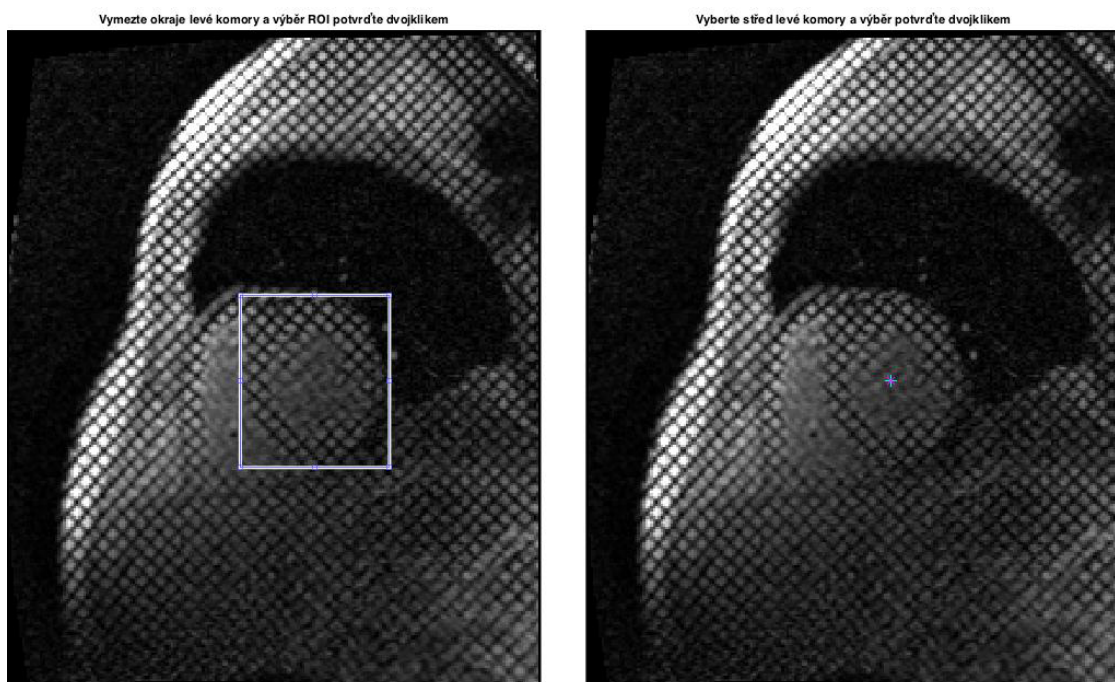
```

% Výběr středů levé komory transversálních řezů
fig=imshow(TG_SA1_snimky_EDIT{1}, []);
set(gcf, 'units','normalized','outerposition',[0 0 1 1], 'Name','Výběr středu LK', 'Number Title','off');
title('Vyberte střed levé komory a výběr potvrďte dvojklikem');
[TG_SA1_ROI_X,TG_SA1_ROI_Y]=getpts;
close all;

```

Tento proces je následně zopakován pro zbylé 2 řezy. Pro zadání pozice středu byl použit příkaz `getpts` a jak vertikální tak horizontální hodnota byla uložena do proměnných `TG_SA1_ROI_X` a `TG_SA1_ROI_Y`.

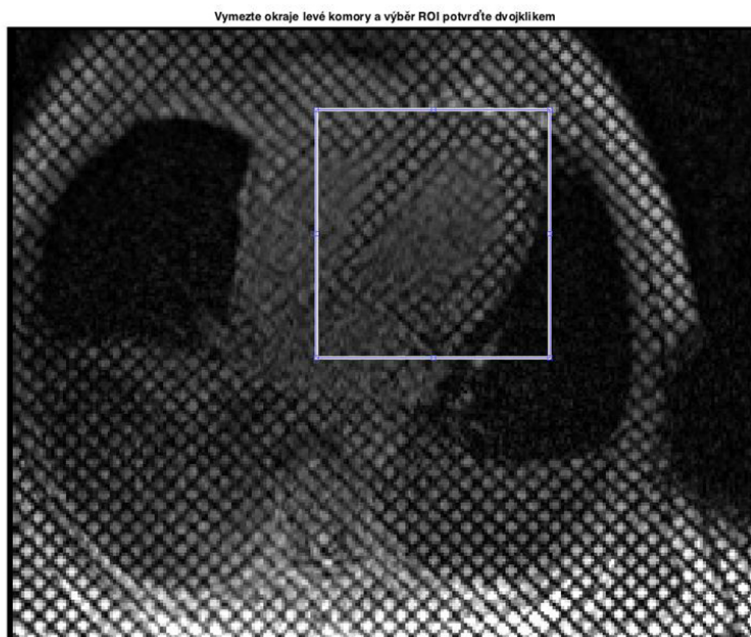
Vymezení okrajů levé komory a výběr středu u bazálního řezu můžeme pozorovat na Obr. 2.5.



Obr. 2.5: Výběr ROI a středu levé komory transversálního řezu.

2.3.2 ROI longitudinálních řezů

Longitudinální řezy se na rozdíl od těch transversálních od sebe liší rotací, rozměry a přiblížením daného snímku. Z toho důvodu je potřebné ROI vybírat pro každý řez zvlášť. U longitudinálních řezů není důvod pro čtvercový ROI, proto se žádný poloměr nepočítá. Pro vymezení rozměrů levé komory se používá stejné metody jako u transversálních řezů, čili příkazu `imcrop`. Na Obr. 2.6 je znázorněn výběr ROI u sekvence 4CH.



Obr. 2.6: Výběr ROI longitudinálního řezu 4CH.

2.3.3 Zvětšení a oříznutí snímků

Vzhledem k tomu, že MRI snímky, zejména ty s přídatným gradientem, jsou velmi malé a nedosahují ani rozměrů 300 x 300 pixelů, je nutné vymezené ROI zvětšit. Pomocí funkce `imcrop` a parametrů získaných při výběru ROI jsou snímky oříznuty. Longitudinální řezy jsou, stejně jako řezy transversální, oříznuty tak, aby rozměr ROI byl 1,4násobkem rozměru levé komory.

```
% Ořezání původních snímků podle ROI a zvětšení snímků ROI
resize=10; % poměr zvětšení snímků ROI
for i=1:length(TG_SA1_ROI)
    TG_SA1_ROI{1,i}=imcrop(TG_SA1_snimky_EDIT{1,i}, [(round(TG_SA1_ROI_X)-ROI
radius) (round(TG_SA1_ROI_Y)-ROIradius) (2*ROIradius) (2*ROIradius)]);
    TG_SA1_ROI{i}=imresize(TG_SA1_ROI{i},resize);
end
for i=1:length(TG_4CH_ROI)
    TG_4CH_ROI{1,i}=imcrop(TG_4CH_snimky_EDIT{1,i}, [(round(coordROI_4CH(1))-
(coordROI_4CH(3)*0.2)) (round(coordROI_4CH(2))-(coordROI_4CH(4)*0.2)) (co
ordROI_4CH(3)*1.4) (coordROI_4CH(4)*1.4)]);
    TG_4CH_ROI{i}=imresize(TG_4CH_ROI{i},resize);
end
```


Následně po oříznutí jsou příkazem `imresize` snímky zvětšeny. Testováním bylo zjištěno, že ideální je desetinásobné zvětšení. V části skriptu je ukázáno oříznutí a zvětšení snímků SA1 a 4CH, obdobně jsou však zvětšeny i stávající snímky.

2.3.4 Výběr os a bodů longitudinálních řezů

Po vytvoření ROI je u longitudinálních řezů nutno vyznačit osy levé komory, obdobně jako u transversálních řezů středové body levé komory.

```
% Výběr os levé komory longitudinálních řezů a výběr bodů pro segmentaci
fig=imshow(TG_2CH_ROI{1}, []);
set(gcf, 'units','normalized','outerposition',[0 0 1 1], 'Name','Výběr osy LK',
'NumberTitle','off');
title('Prvním bodem označte vrchol (apex) levé komory a tahem určete osu LK,
dvojklikem potvrďte');
imf = imline;
osa_TG_2CH_ROI=wait(imf);
title('Vyberte nejapikálnější bod UVNITŘ LK, dvojklikem potvrďte');
[TG_2CH_APIin_X,TG_2CH_APIin_Y]=getpts;
title('Vyberte nejbazálnější bod LK, dvojklikem potvrďte');
[TG_2CH_BAout_X,TG_2CH_BAout_Y]=getpts;
close all;
```

Osa levé komory je definována pomocí příkazu `imline` tak, že první bod určuje nejapikálnější bod levé komory a druhý bod určuje osu. Délka této úsečky není podstatná. Ideální však je, pokud její délka přesahuje délku myokardu. Výhodou funkce `imline` je, že se vyznačené body dají upravovat až do doby, kdy je jejich správnost potvrzena.

Kromě osy je u longitudinálních snímků potřeba vyznačit 2 další body. Jedním z nich je nejapikálnější bod uvnitř dutiny levé komory a dalším bodem je nejbazálnější bod myokardu. Ideálně jsou tyto body vyznačeny v blízkosti osy levé komory. Význam těchto bodů je objasněn v podkapitole 2.6 Segmentace podle modelu AHA. Na Obr. 2.7 je naznačen výběr osy levé komory spolu s dvěma body.



Obr. 2.7: Určení osy a bodů levé komory longitudinálního řezu 2CH.

2.4 Fourierova transformace a návrh filtrů

Snímky s přídavným gradientem obsahují výrazné mřížky v dvou na sebe kolmých směrech. Po transformaci obsahuje spektrum výrazné frekvenční složky. Z toho důvodu je vhodné použít metodu Rychlé Fourierovy transformace, neboli FFT [14].

2.4.1 2D Fourierova transformace a amplitudové spektrum

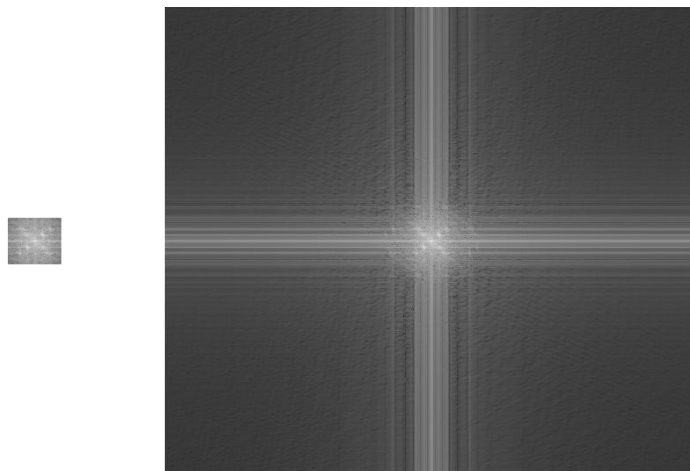
Část skriptu, která se zabývá transformací je nutno začít odstraněním dočasných proměnných, které v programu pravděpodobně zůstaly po předešlém výpočtu. Z důvodu, že transformace je náročnější na výkon počítače a zároveň i na čas, bylo pro ilustraci přidáno okno vyjadřující stav procesu, takzvaný „waitbar“.

```
clear F_TG_SA1_ROI iF1_TG_SA1_ROI_magnitude iF1_TG_SA1_ROI;
clear F_TG_SA2_ROI iF1_TG_SA2_ROI_magnitude iF1_TG_SA2_ROI;
clear F_TG_SA3_ROI iF1_TG_SA3_ROI_magnitude iF1_TG_SA3_ROI;
clear F_TG_2CH_ROI iF1_TG_2CH_ROI_magnitude iF1_TG_2CH_ROI;
clear F_TG_3CH_ROI iF1_TG_3CH_ROI_magnitude iF1_TG_3CH_ROI;
clear F_TG_4CH_ROI iF1_TG_4CH_ROI_magnitude iF1_TG_4CH_ROI;
WB = waitbar(0, 'Spouštění...');
set(WB, 'Name', 'Probíhá výpočet...');
Perc=0; PercMax=18;
```

Samotná 2D Rychlá Fourierova transformace byla provedena příkazem `fft2`, ze které je potřeba získat amplitudové spektrum pomocí příkazu `fftshift` a dopočtením rovnice.

```
% FFT
for i=1:length(TG_SA1_ROI)
    F_TG_SA1_ROI{i}=fft2(TG_SA1_ROI{i}); % 2D fourierova transformace
    F_TG_SA1_ROI_magnitude{i}=log(1+abs(fftshift(F_TG_SA1_ROI{i})));
    % Amplitudové spektrum
end; Perc=Perc+1;
waitbar(Perc/PercMax, WB, sprintf('%d%%', ceil(Perc/PercMax*100)));
```

Amplitudová spektra jsou následně dopočtena pro všechny snímky všech 6 sekvencí obdobným způsobem jako v uvedeném skriptu. Výsledné amplitudové spektrum je ilustrováno na Obr. 2.8. Pro porovnání je uvedeno spektrum původního a zvětšeného ROI. Z tohoto porovnání je patrné, že zvětšení nemá na vlastnosti mřížky vliv.



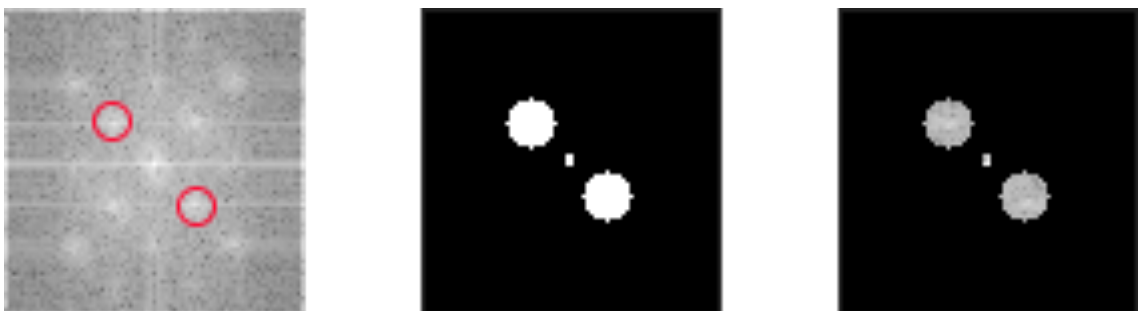
Obr. 2.8: Porovnání amplitudového spektra původního a zvětšeného ROI.

2.4.2 Tvorba filtrů pro rekonstrukci mřížky

Cílem filtrace amplitudového spektra je postupně odfiltrovat vše kromě frekvenční a stejnosměrné složky. Frekvenční složka spektra reprezentuje tu část původního snímku, která se opakuje s určitou frekvencí. Opakující se čáry jsou v spektru reprezentovány jedním bodem. Jelikož je mřížka složená z rady na sebe kolmých čar, je také spektrum souměrné. Z praktických důvodů je filtrace provedena pro každý směr mřížky zvlášť. Body které reprezentují frekvenční složku spektra se nacházejí v pozici maxim na diagonále symetricky podle středu co nejbliž k centru amplitudového spektra. Stejnosměrná složka odpovídá pixelu umístěnému přesně ve středu amplitudového spektra [15]. Výsledný filtr má tedy tvar dvou středově souměrných kruhů a jednoho pixelu uprostřed jejich spojnice. Pro mřížku v opačném směru je filtr horizontálně převrácen.

```
polomer_filtru=5; % poloměr filtru pro filtraci amplitudového spektra
hodnota_peaku_SA1=max(max(F_TG_SA1_ROI_magnitude{1}([1:(0.498*(sizeROI_SA(1))
],[1:(0.498*(sizeROI_SA(2)))])));
[y1,x1]=find(F_TG_SA1_ROI_magnitude{1}([1:(0.498*(sizeROI_SA(1))],[1:(0.498*(
sizeROI_SA(2)))]))==hodnota_peaku_SA1);
x2=sizeROI_SA(2)-x1; y2=sizeROI_SA(1)-y1;
[y3,x3]=find((F_TG_SA1_ROI_magnitude{1})==(max(max(F_TG_SA1_ROI_magnitude{1}
))));
[X1,Y1]=meshgrid(-(x1-1):(sizeROI_SA(2)-x1),-(y1-1):(sizeROI_SA(1)-y1));
[X2,Y2]=meshgrid(-(x2-1):(sizeROI_SA(2)-x2),-(y2-1):(sizeROI_SA(1)-y2));
filter1_SA1=((X1.^2+Y1.^2)<=polomer_filtru^2)+((X2.^2+Y2.^2)<=polomer_filtru^2
);
filter2_SA1=fliplr(filter1_SA1);
filter1_SA1(y3,x3)=1; % filtr amplitudového spektra pro první směr SA1 ROI
filter2_SA1(y3,x3)=1; % filtr amplitudového spektra pro druhý směr SA1 ROI
Perc=Perc+1; waitbar(Perc/PercMax,WB,sprintf('%d%%',ceil(Perc/PercMax*100)));
```

Experimentálně bylo zjištěno, že nevhodnější poloměr filtru pro frekvenční složku je 5 pixelů. Pozice umístění středů těchto filtrů je automaticky dopočtena pomocí maxima amplitudového spektra v dané oblasti. Pro každou sekvenci je pak vytvořen vlastní filtr stejným způsobem. Na Obr. 2.9 jsou vyznačeny pozice pro filtraci frekvenční složky amplitudového spektra, výsledný filtr a amplitudové spektrum po aplikaci filtru.



Obr. 2.9: Vyznačené pozice pro filtraci frekvenční složky amplitudového spektra, výsledný filtr a amplitudové spektrum po aplikaci filtru.

2.4.3 Další filtry pro následující operace

Během celé analýzy dat je potřebné větší množství univerzálních filtrů a operátorů, které jsou dále vícekrát během procesu použity.

```
filter_mrizka_SA=zeros(sizeROI_SA(1),sizeROI_SA(2));
filter_mrizka_SA([1:5:end],:)=0.6; % filtr pro rekonstrukci mřížky
transversálních řezů
filter_mrizka_2CH=zeros(sizeROI_2CH(1),sizeROI_2CH(2));
filter_mrizka_2CH([1:5:end],:)=0.6; % filtr pro rekonstrukci mřížky 2CH
filter_mrizka_3CH=zeros(sizeROI_3CH(1),sizeROI_3CH(2));
filter_mrizka_3CH([1:5:end],:)=0.6; % filtr pro rekonstrukci mřížky 3CH
filter_mrizka_4CH=zeros(sizeROI_4CH(1),sizeROI_4CH(2));
filter_mrizka_4CH([1:5:end],:)=0.6; % filtr pro rekonstrukci mřížky 4CH
prewitt1=[ 1 1 0; 1 0 -1; 0 -1 -1]; % prewittův operátor pro první směr
prewitt2=[0 1 1; -1 0 1; -1 -1 0]; % prewittův operátor pro druhý směr
expand=strel('disk',50); % parametr pro hledání okolí průsečníků
```

Filtry jsou použity především pro eliminaci příliš velkého počtu bodů rekonstruované mřížky. Zejména z důvodu zmenšení výpočetní doby a zároveň proto, že se jedná o zvětšená data a frekvence mřížky je i po aplikaci filtru 2x větší než před zvětšením.

Prewittovy operátory jsou používány při detekci hran. Mají tvar:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \text{ a } \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

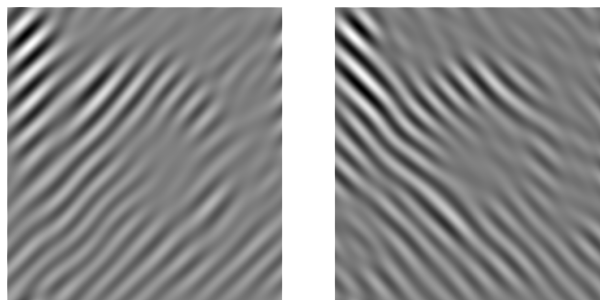
a jsou využívány v dalších částech skriptu pro rekonstrukci mřížky.

2.4.4 Zpětná Fourierova transformace

Cílem zpětné Fourierovy transformace je z každého původního snímku získat 2 nové snímky které obsahují pouze mřížku v jednom směru.

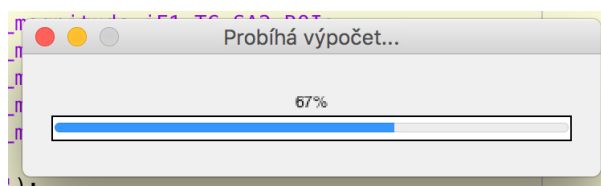
```
Zpětná FFT odfiltrovaných snímků
for i=1:length(TG_SA1_ROI)
    iF1_TG_SA1_ROI{i}=fftshift(F_TG_SA1_ROI{i}).*filter1_SA1;
    iF2_TG_SA1_ROI{i}=fftshift(F_TG_SA1_ROI{i}).*filter2_SA1;
    iF1_TG_SA1_ROI{i}=ifft2(ifftshift(iF1_TG_SA1_ROI{i}));
    iF2_TG_SA1_ROI{i}=ifft2(ifftshift(iF2_TG_SA1_ROI{i}));
end; Perc=Perc+1;
waitbar(Perc/PercMax,WB,sprintf('%d%%',ceil(Perc/PercMax*100)));
delete(WB);
```

Odfiltrované amplitudové spektrum je za použití příkazů `fftshift` a `ifft2`, který odpovídá zpětné 2D Fourierově transformaci, převedeno zpět na snímky obsahující pouze mřížku v jednom směru. Výsledky jedné filtrace jsou zobrazeny na Obr. 2.10.



Obr. 2.10: Výsledky zpětné Fourierovy transformace.

Po dokončení filtrace automaticky zmizí i „waitbar“ když nabyde 100%. Ukázka na Obr. 2.11.



Obr. 2.11: Waitbar.

2.5 Extrakce obrysů levé komory

Obrysy levé komory jsou důležité pro určení plochy levé komory a tím vytvoření masky pozice svalů v každém okamžiku sekvence. Tyto masky jsou následně využity pro rekonstrukci a výpočet průniků mřížky. Tento proces však nemůže být automatizován. Je zapotřebí manuálně vyznačit obrys levé komory v každém snímku sekvence. Tento proces je velmi zdoluhavý a náročný na obsluhu. Některé snímky, zejména longitudinálních řezů, jsou špatně rozlišitelné a určit přesný okraj je velmi náročné.

2.5.1 Extrakce obrysů transversálních řezů

```
clear maska_SA1 maska_SA2 maska_SA3 maska_2CH maska_3CH maska_4CH;
```

Tato část skriptu začíná odstraněním dočasných proměnných z předešlých výpočtů.

```
% Segmentace transversálních snímků
clear pos_poly_x_in pos_poly_x_out pos_poly_in pos_poly_out;
fig = figure('units','normalized','outerposition',[0 0 1 1],'Name','Segmentace
LK','NumberTitle','off');
for i=1:length(TG_SA1_ROI)
    if ~ishandle(fig)
        break
    else
        fig, imshow((TG_SA1_ROI{i}), []);
        for j=1:2
            if j==1
                title('Vyznačte VNITŘNÍ obrys levé komory');
                if i==1
                    imf = impoly;
                else
                    imf = impoly(gca, pos_poly_x_in{i-1});
                end
                pos_poly_x_in{i} = wait(imf);
                pos_poly_in{i}=transpose(pos_poly_x_in{i});
                pos_poly_in{i}(:, length(pos_poly_in{i})+1)
                =pos_poly_in{i}(:, 1);
                delete(imf);
                hold on
                in=fnplt(cscvn(pos_poly_in{i}));
                in_fill=fill(in(1,:),in(2,:),[1 0 0]);
                set(in_fill,'FaceAlpha',.33,'EdgeAlpha', 0);
                drawnow
            else
                title('Vyznačte VNĚJŠÍ obrys levé komory');
                if i==1
                    imf = impoly;
                else

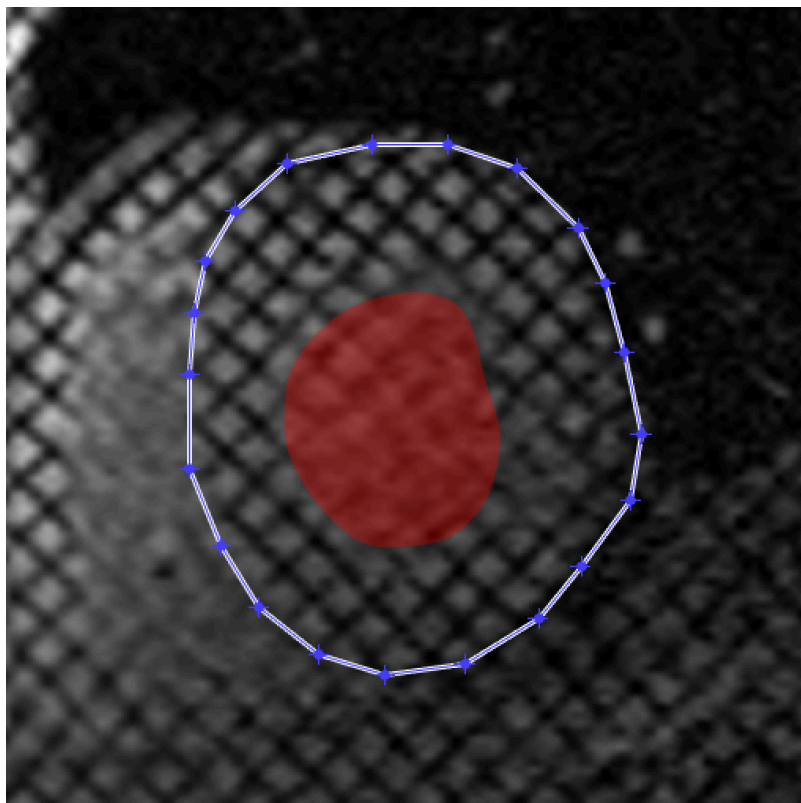
```

```

        imf = impoly(gca, pos_poly_x_out{i-1});
    end
    pos_poly_x_out{i} = wait(imf);
    pos_poly_out{i}=transpose(pos_poly_x_out{i});
    pos_poly_out{i}(:, length(pos_poly_out{i})+1)
    =pos_poly_out{i}(:, 1);
    delete(imf);
    hold on
    out=fnplt(cscvn(pos_poly_out{i}));
    out_fill=fill(out(1,:),out(2,:),[1 0.27 0]);
    set(out_fill, 'FaceAlpha', .33, 'EdgeAlpha', 0);
    drawnow
    pause (0.5)
end
end
end
end
maska_out=poly2mask(out(1,:),out(2,:),sizeROI_SA(1),sizeROI_SA(2));
maska_in=poly2mask(in(1,:),in(2,:),sizeROI_SA(1),sizeROI_SA(2));
maska_in=-maska_in;
maska_SA1{i}=maska_out.*maska_in;
end
close(fig);

```

Okraj levé komory je naznačen postupně. Nejdřív vnitřní okraj, následně vnější. Obrys je naznačen sérií bodů prostřednictvím funkce `impoly`. Tento obrys je následně interpolací proložen uzavřenou křivkou a převeden na masku. Interpolaci zabezpečují funkce, které jsou součástí Curve Fitting Toolbox™ [16]. Maska je prezentována transparentní vrstvou přímo na zpracovávaném snímku během segmentace vnějšího okraje, vizte Obr. 2.12. Pro vznik hotové masky levé komory je potřebné od sebe odečíst vnější a vnitřní masku.



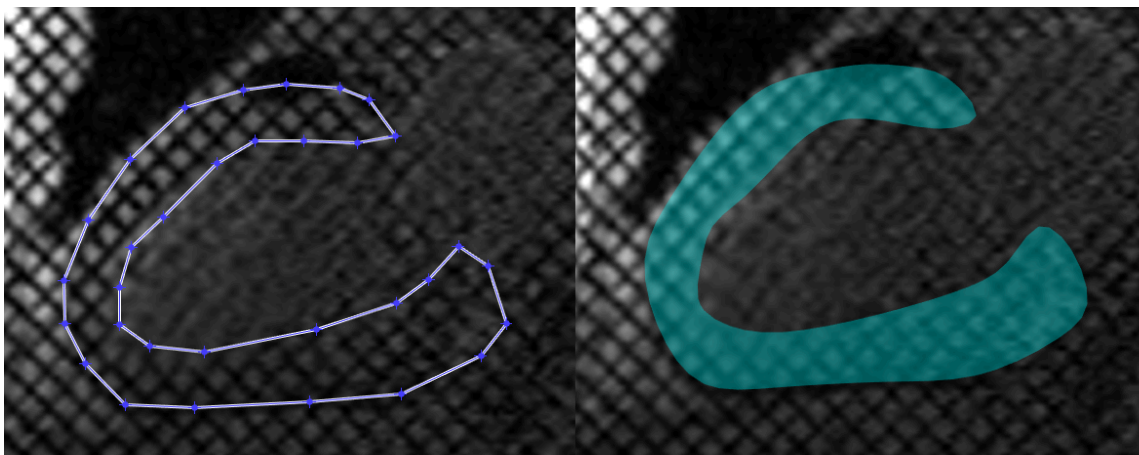
Obr. 2.12: Vyznačení obrysů levé komory s naznačenou maskou vnitřního obrysu.

2.5.2 Extrakce obrysů longitudinálních řezů

U longitudinálních řezů se nejedná o dutinu jako u řezů transversálních, proto nejsou zapotřebí 2 masky, ale postačuje jedna.

```
% Segmentace longitudinálních snímků
clear pos_poly_x pos_poly;
fig = figure('units','normalized','outerposition',[0 0 1 1],'Name','Segmentace
LK','NumberTitle','off');
for i=1:length(TG_2CH_ROI)
    if ~ishandle(fig)
        break
    else
        fig, imshow((TG_2CH_ROI{i}), []);
        title('Vyznačte obrys levé komory');
        if i==1
            imf = impoly;
        else
            imf = impoly(gca, pos_poly_x{i-1});
        end
        pos_poly_x{i} = wait(imf);
        pos_poly{i}=transpose(pos_poly_x{i});
        pos_poly{i}(:, length(pos_poly{i}+1))=pos_poly{i}(:, 1);
        delete(imf);
        hold on
        maska=fnpltt(cscvn(pos_poly{i}));
        in_fill=fill(maska(1,:),maska(2,:),[0 1 1]);
        set(in_fill,'FaceAlpha',.33, 'EdgeAlpha', 0);
        drawnow
        pause (0.5)
    end
    maska_2CH{i}=poly2mask(maska(1,:),maska(2,:),sizeROI_2CH(1),sizeROI_2CH(2));
end
close(fig);
```

Postup je analogický s transversálními řezy, nicméně zjednodušen, protože se jedná o celistvou masku bez dutin. Výběr bodů s následným zakřivením je prezentován na Obr. 2.13.



Obr. 2.13: Výběr polygonu naznačujícího obrys levé komory a transparentní maska.

2.6 Segmentace podle modelu AHA

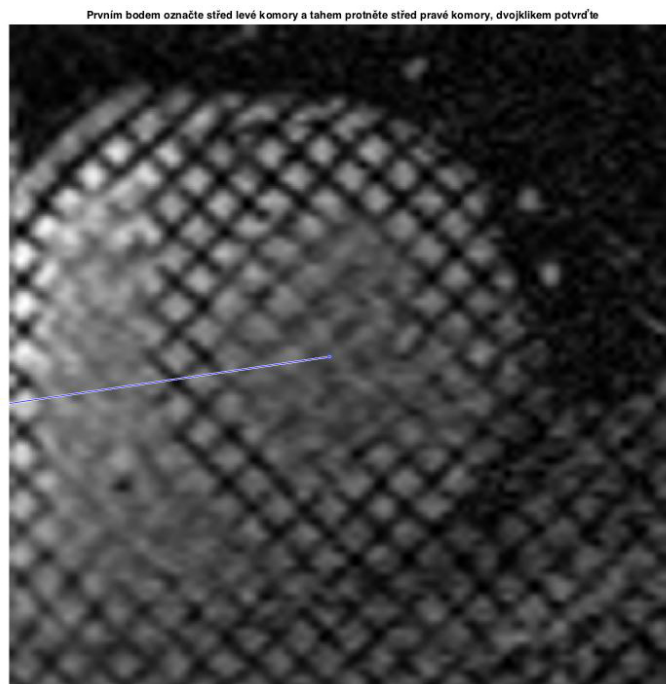
Získané masky je potřeba rozdělit podle modelu AHA. Důvod dělení masek není jen prezentace výsledků, ale také to, že rozdělení výpočtů na menší segmenty výrazně urychlí dobu výpočtu. Princip této segmentace je blíže popsán v kapitole 1.1.2 Segmentace levé komory podle modelu AHA.

2.6.1 Segmentace transversálních masek

Jako první je potřeba definovat natočení snímku MRI. I když platí, že v každé sérii snímků jednoho pacienta jsou všechny 3 transversální snímky natočeny stejným směrem, mezi různými pacienty se tento úhel liší a je tedy potřeba ho manuálně definovat. Úhel natočení odpovídá úhlu mezi horizontální osou obrázku a lateroseptální osou myokardu získanou z bazálního řezu myokardu.

```
fig=imshow(TG_SA1_ROI{1}, []);  
set(gcf, 'units','normalized','outerposition',[0 0 1 1],'Name','Výběr lateroseptální osy LK','NumberTitle','off');  
title('Prvním bodem označte střed levé komory a tahem protněte střed pravé komory, dvojklikem potvrďte');  
imf = imline;  
osa_TG_SA_ROI=wait(imf);  
rotace=atan2(osa_TG_SA_ROI(2,2)-osa_TG_SA_ROI(1,2),osa_TG_SA_ROI(2,1)-osa_TG_SA_ROI(1,1))*180/pi; % lateroseptální natočení osy ve stupních  
close all;
```

Proměnná `rotace` byla získána příkazem `atan2` a následným převodem z radiánů na stupně. Během tohoto procesu je nevyhnutné dodržet posloupnost zadávání této osy. Při záměně by mohlo dojít k převrácení masek a tudíž k špatnému přiřazení segmentů. Definice lateroseptální osy je ilustrována na Obr. 2.14.



Obr. 2.14: Definice lateroseptální osy.

Následná segmentace je u sekvencí Base a Mid totožná. U Apikální sekvence se dělí pouze na 4 části. Pozice segmentů je vypočítána z natočení lateroseptální osy a dalších předem definovaných parametrů, zejména středu snímku a počtu požadovaných segmentů.

```
for i=1:4 % segmentace apical
    [FI,F]=cart2pol((0:0.1:999.9),0);
    FI=FI+(((90*i)+135+rotace)*pi)./180);
    [xf,yf]=pol2cart(FI,F);
    x_4seg{i}=xf(end)+sizeROI_SA(1)/2;
    y_4seg{i}=yf(end)+sizeROI_SA(2)/2;
end
```

K výpočtu se používá převod kartézských do polárních souřadnic a také zpětně převod polárních do kartézských s využitím příkazů `cart2pol` a `pol2cart`. Toto zaručuje přesné a rovnoměrné rozdělení masek všech řezů.

```
x_ap_anterior=[sizeROI_SA(1)/2 x_4seg{3} x_4seg{4}];
y_ap_anterior=[sizeROI_SA(2)/2 y_4seg{3} y_4seg{4}];
maska_ap_anterior=poly2mask(x_ap_anterior,y_ap_anterior,sizeROI_SA(1),sizeROI_SA(2));
x_ap_septal=[sizeROI_SA(1)/2 x_4seg{2} x_4seg{3}];
y_ap_septal=[sizeROI_SA(2)/2 y_4seg{2} y_4seg{3}];
maska_septal=poly2mask(x_ap_septal,y_ap_septal,sizeROI_SA(1),sizeROI_SA(2));
x_ap_inferior=[sizeROI_SA(1)/2 x_4seg{1} x_4seg{2}];
y_ap_inferior=[sizeROI_SA(2)/2 y_4seg{1} y_4seg{2}];
maska_ap_inferior=poly2mask(x_ap_inferior,y_ap_inferior,sizeROI_SA(1),sizeROI_SA(2));
x_ap_lateral=[sizeROI_SA(1)/2 x_4seg{1} x_4seg{4}];
y_ap_lateral=[sizeROI_SA(2)/2 y_4seg{1} y_4seg{4}];
maska_lateral=poly2mask(x_ap_lateral,y_ap_lateral,sizeROI_SA(1),sizeROI_SA(2))
```

K vytvoření masek je využita funkce `poly2mask` která ze zadaných souřadnic bodů, konkrétně středu obrázku a koncových bodů natočených úseček, a rozměrů původního snímku vytvoří logickou matici, která slouží jako maska k další segmentaci. Masky apikálních segmentů jsou zobrazeny na Obr. 2.15.



Obr. 2.15: Masky všech apikálních segmentů.

Samotné rozčlenění extrahovaných obrysů levé komory je nyní velice jednoduché. Stačí vynásobit masky sekvence se správnou maskou segmentu.

```
for i=1:length(TG_SA3_ROI) % členění masek apical
    maska_apical_anterior{i}=maska_SA3{i}.*maska_ap_anterior;
    maska_apical_septal{i}=maska_SA3{i}.*maska_septal;
    maska_apical_inferior{i}=maska_SA3{i}.*maska_ap_inferior;
    maska_apical_lateral{i}=maska_SA3{i}.*maska_lateral;
end
```

2.6.2 Segmentace longitudinálních masek

Longitudinální řezy jsou segmentovány na 7 částí. Obdobně jako u transversálních snímků hraje roli natočení myokardu. To je dopočteno z osy zadané při výběru ROI.

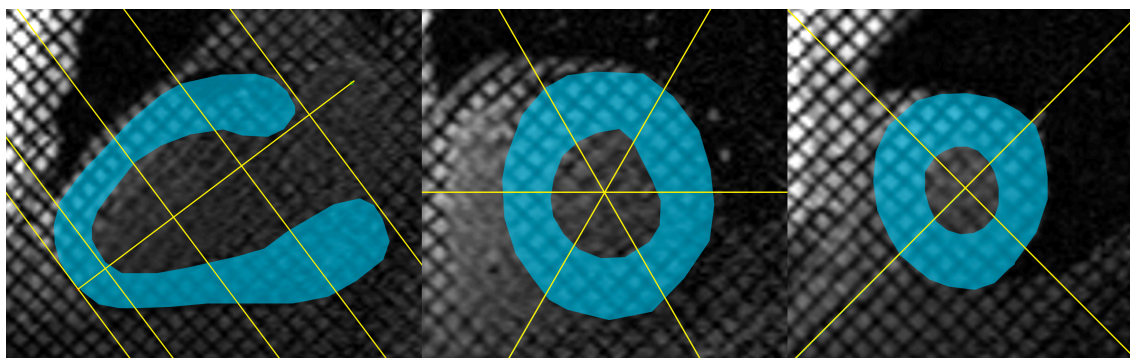
```
uhel_2CH=atan2(osa_TG_2CH_ROI(2,2)-osa_TG_2CH_ROI(1,2),osa_TG_2CH_ROI(2,1)-
osa_TG_2CH_ROI(1,1))*180/pi;
for i=1:2
    [FI,F]=cart2pol((0:0.1:999.9),0);
    FI=FI+(((180*(i-1))+90+uhel_2CH)*pi)./180);
    [xf,yf]=pol2cart(FI,F);
    x_2CHseg{i}=xf(end)+osa_TG_2CH_ROI(1,1);
    y_2CHseg{i}=yf(end)+osa_TG_2CH_ROI(1,2);
end
for i=1:2
    [FI,F]=cart2pol((0:0.1:999.9),0);
    FI=FI+(((180*(i-1))+90+uhel_2CH)*pi)./180);
    [xf,yf]=pol2cart(FI,F);
    x_2CHseg{i+2}=xf(end)+TG_2CH_APin_X;
    y_2CHseg{i+2}=yf(end)+TG_2CH_APin_Y;
end
for i=1:2
    [FI,F]=cart2pol((0:0.1:999.9),0);
    FI=FI+(((180*(i-1))+90+uhel_2CH)*pi)./180);
    [xf,yf]=pol2cart(FI,F);
    if osa_TG_2CH_ROI(1,1)<osa_TG_2CH_ROI(1,2)
        x1_2CHseg=TG_2CH_APin_X+(TG_2CH_BAout_X-TG_2CH_APin_X)/3);
        x_2CHseg{i+4}=xf(end)+x1_2CHseg;
    else
        x1_2CHseg=TG_2CH_APin_X-((TG_2CH_APin_X-TG_2CH_BAout_X)/3);
        x_2CHseg{i+4}=xf(end)+x1_2CHseg;
    end
    if osa_TG_2CH_ROI(2,1)<osa_TG_2CH_ROI(2,2)
        y1_2CHseg=TG_2CH_APin_Y+(TG_2CH_BAout_Y-TG_2CH_APin_Y)/3);
        y_2CHseg{i+4}=yf(end)+y1_2CHseg;
    else
        y1_2CHseg=TG_2CH_APin_Y-((TG_2CH_APin_Y-TG_2CH_BAout_Y)/3);
        y_2CHseg{i+4}=yf(end)+y1_2CHseg;
    end
end
for i=1:2
    [FI,F]=cart2pol((0:0.1:999.9),0);
    FI=FI+(((180*(i-1))+90+uhel_2CH)*pi)./180);
    [xf,yf]=pol2cart(FI,F);
    if osa_TG_2CH_ROI(1,1)<osa_TG_2CH_ROI(1,2)
        x2_2CHseg=TG_2CH_APin_X+(2*((TG_2CH_BAout_X-TG_2CH_APin_X)/3));
        x_2CHseg{i+6}=xf(end)+x2_2CHseg;
    else
        x2_2CHseg=TG_2CH_APin_X-(2*((TG_2CH_APin_X-TG_2CH_BAout_X)/3));
        x_2CHseg{i+6}=xf(end)+x2_2CHseg;
    end
    if osa_TG_2CH_ROI(2,1)<osa_TG_2CH_ROI(2,2)
        y2_2CHseg=TG_2CH_APin_Y+(2*((TG_2CH_BAout_Y-TG_2CH_APin_Y)/3));
        y_2CHseg{i+6}=yf(end)+y2_2CHseg;
    else
        y2_2CHseg=TG_2CH_APin_Y-(2*((TG_2CH_APin_Y-TG_2CH_BAout_Y)/3));
        y_2CHseg{i+6}=yf(end)+y2_2CHseg;
    end
end
for i=1:2
    [FI,F]=cart2pol((0:0.1:999.9),0);
    FI=FI+(((180*(i-1))+90+uhel_2CH)*pi)./180);
    [xf,yf]=pol2cart(FI,F);
    x_2CHseg{i+8}=xf(end)+TG_2CH_BAout_X;
    y_2CHseg{i+8}=yf(end)+TG_2CH_BAout_Y;
end
```

Prostor nad tzv. apical cap, který je popsán na Obr. 1.1, je úměrně rozdělen na 6 částí, 3 po každé straně osy. Tento prostor je určen pomocí dvou bodů taktéž získaných při výběru ROI. Zde však bylo potřeba ošetřit různé možnosti natočení myokardu. Po výpočtu pozice segmentace jsou stejně jako u transversálních snímků vytvořeny masky odpovídající každému segmentu pomocí funkce `poly2mask`.

```
x_4CH_apex=[x_4CHseg{1} x_4CHseg{2} x_4CHseg{4} x_4CHseg{3}];
y_4CH_apex=[y_4CHseg{1} y_4CHseg{2} y_4CHseg{4} y_4CHseg{3}];
maska_4CHseg_apex=poly2mask(x_4CH_apex,y_4CH_apex,sizeROI_4CH(1),sizeROI_4CH(2));
x_4CH_apical_septal=[x_4CHseg{3} TG_4CH_APIin_X x1_4CHseg x_4CHseg{5}];
y_4CH_apical_septal=[y_4CHseg{3} TG_4CH_APIin_Y y1_4CHseg y_4CHseg{5}];
maska_4CHseg_apical_septal=poly2mask(x_4CH_apical_septal,y_4CH_apical_septal,sizeROI_4CH(1),sizeROI_4CH(2));
x_4CH_mid_inferoseptal=[x_4CHseg{5} x1_4CHseg x2_4CHseg x_4CHseg{7}];
y_4CH_mid_inferoseptal=[y_4CHseg{5} y1_4CHseg y2_4CHseg y_4CHseg{7}];
maska_4CHseg_mid_inferoseptal=poly2mask(x_4CH_mid_inferoseptal,y_4CH_mid_inferoseptal,sizeROI_4CH(1),sizeROI_4CH(2));
x_4CH_basal_inferoseptal=[x_4CHseg{7} x2_4CHseg TG_4CH_BAout_X x_4CHseg{9}];
y_4CH_basal_inferoseptal=[y_4CHseg{7} y2_4CHseg TG_4CH_BAout_Y y_4CHseg{9}];
maska_4CHseg_basal_inferolateral=poly2mask(x_4CH_basal_inferoseptal,y_4CH_basal_inferoseptal,sizeROI_4CH(1),sizeROI_4CH(2));
x_4CH_apical_lateral=[x_4CHseg{4} TG_4CH_APIin_X x1_4CHseg x_4CHseg{6}];
y_4CH_apical_lateral=[y_4CHseg{4} TG_4CH_APIin_Y y1_4CHseg y_4CHseg{6}];
maska_4CHseg_apical_lateral=poly2mask(x_4CH_apical_lateral,y_4CH_apical_lateral,sizeROI_4CH(1),sizeROI_4CH(2));
x_4CH_mid anterolateral=[x_4CHseg{6} x1_4CHseg x2_4CHseg x_4CHseg{8}];
y_4CH_mid anterolateral=[y_4CHseg{6} y1_4CHseg y2_4CHseg y_4CHseg{8}];
maska_4CHseg_mid anterolateral=poly2mask(x_4CH_mid anterolateral,y_4CH_mid anterolateral,sizeROI_4CH(1),sizeROI_4CH(2));
x_4CH_basal anterolateral=[x_4CHseg{8} x2_4CHseg TG_4CH_BAout_X x_4CHseg{10}];
y_4CH_basal anterolateral=[y_4CHseg{8} y2_4CHseg TG_4CH_BAout_Y y_4CHseg{10}];
maska_4CHseg_basal anterolateral=poly2mask(x_4CH_basal anterolateral,y_4CH_basal anterolateral,sizeROI_4CH(1),sizeROI_4CH(2));
```

Členění obrysů levé komory je následně identické jako u transversálních řezů.

Jelikož je tato část skriptu velice rozsáhlá, byly zde ilustrovány jenom segmentace apikálního a dvoudutinového řezu. Pro zbylé řezy je tento proces analogický. Na Obr. 2.16 můžeme pozorovat načrtnutí segmentace longitudinálního řezu a také porovnání šestisegmentového a apikálního členění transversálních řezů.



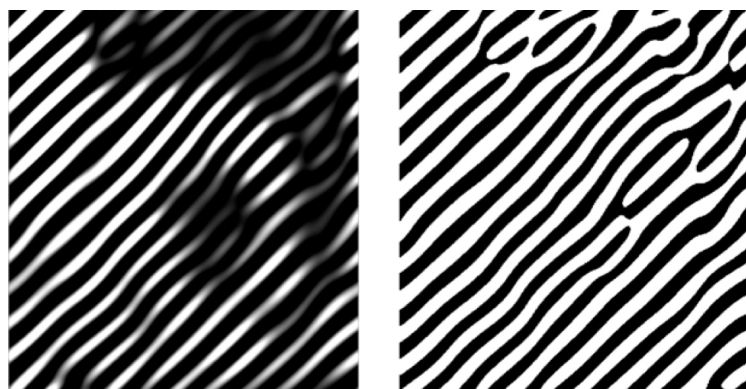
Obr. 2.16: Segmentace podle modelu AHA.

2.7 Rekonstrukce mřížky

Rekonstrukce mřížky je časově a výpočetně náročnější část skriptu, z toho důvodu byl stejně jako u Fourierovy transformace použit waitbar.

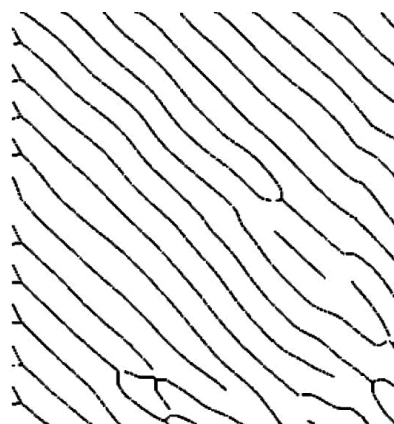
Jako první bylo potřeba detekovat hrany mřížky vzniklé zpětnou Fourierovou transformací. K tomu byla využita konvoluce s využitím Prewittových operátorů [17]. Konkrétní použité operátory jsou popsány v kapitole 2.4.3 Další filtry pro následující operace.

Takto vzniklé výsledky konvoluce bylo potřeba prahovat za využití filtru typu horní propust. Ve výsledku se v upravené mřížce nachází jenom body s hodnotami jasu 0 a 255. Na Obr. 2.17 pozorujeme výsledky konvoluce a prahování.



Obr. 2.17: Detekce hran a prahování.

Po prahování bylo potřeba najít střed oblastí s hodnotou jasu 255. K tomu byly použity morfologické operace `bwmorph` s parametry `'skel'` a `'spur'`. Parametr `'skel'` provede takzvanou skeletonizaci, čili najde střed oblastí a ten reprezentuje pouze jedním bodem. Parametr `'spur'` následně odstraní body které z jinak souvislé čáry odstupují [18]. Výsledek skeletonizace je znázorněn na Obr. 2.18.



Obr. 2.18: Zrekonstruovaná mřížka v jenom směru.

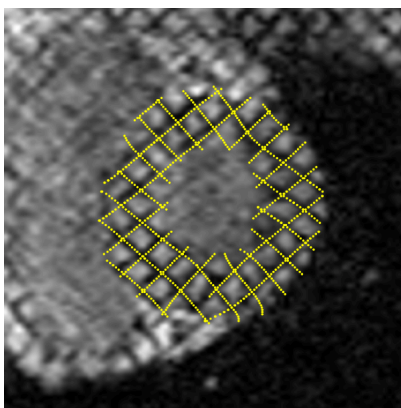
Nyní je potřeba najít pozice bodů zrekonstruované mřížky z důvodu jejich prezentace na původních snímcích. Nemá význam prezentovat mřížku v celém rozsahu obrázku, podstatná je prezentace pouze v oblasti vymezené levou komorou. Z toho důvodu byl výsledný obrázek vynásoben maskou odpovídajícího snímku v sekvenci daného řezu. Zároveň byl použit filtr, který ponechá pouze každý pátý bod zrekonstruované mřížky. Toto opatření značně urychlilo proces výpočtu. Těmto třem operacím pro jeden řez odpovídá následující část skriptu.

```

for i=1:length(TG_SA1_ROI) % rekonstrukce mřížky BASE
    mrizka1_SA1{i}=uint8(conv2(abs(iF1_TG_SA1_ROI{i}),(prewitt1/3)));
    mrizka2_SA1{i}=uint8(conv2(abs(iF2_TG_SA1_ROI{i}),(prewitt2/3)));
    for j=1:sizeROI_SA(1)
        for k=1:sizeROI_SA(2)
            if mrizka1_SA1{i}(j,k)>0
                mrizka1_SA1{i}(j,k)=0;
            else
                mrizka1_SA1{i}(j,k)=255;
            end
            if mrizka2_SA1{i}(j,k)>0
                mrizka2_SA1{i}(j,k)=0;
            else
                mrizka2_SA1{i}(j,k)=255;
            end
        end
    end
    mrizka1_SA1{i}=bwmorph(mrizka1_SA1{i},'skel',inf);
    mrizka1_SA1{i}=bwmorph(mrizka1_SA1{i},'spur',25);
    mrizka1_SA1{i}=blkdiag(zeros(resize),mrizka1_SA1{i}([2:(sizeROI_SA(1)-
    resize+1)],[2:(sizeROI_SA(2)-resize+1)]));
    mrizka2_SA1{i}=bwmorph(mrizka2_SA1{i},'skel',inf);
    mrizka2_SA1{i}=bwmorph(mrizka2_SA1{i},'spur',25);
    mrizka2_SA1{i}=blkdiag(zeros(resize),mrizka2_SA1{i}([2:(sizeROI_SA(1)-
    resize+1)],[2:(sizeROI_SA(2)-resize+1)]));
    mrizka2_SA1{i}=[mrizka2_SA1{i}(:,(resize+1):end),
    zeros(sizeROI_SA(1),resize)];
    mrizka1_SA1{i}=mrizka1_SA1{i}.*round(filter_mrizka_SA).*maska_SA1{i};
    mrizka2_SA1{i}=mrizka2_SA1{i}.*round(filter_mrizka_SA).*maska_SA1{i};
    [yM1_SA1{i},xM1_SA1{i}]=find(mrizka1_SA1{i)==1);
    [yM2_SA1{i},xM2_SA1{i}]=find(mrizka2_SA1{i)==1);
    Perc=Perc+1;
    waitbar(Perc/PercMax,WB,sprintf('%d%%',ceil(Perc/PercMax*100)));
end

```

Pomocí funkce find jsou zjištěny souřadnice bodů a ty pak mohou být prezentovány jako animace na původních snímcích. Vizte Obr. 2.19.



Obr. 2.19: Zrekonstruovaná mřížka prezentovaná na původním snímku.

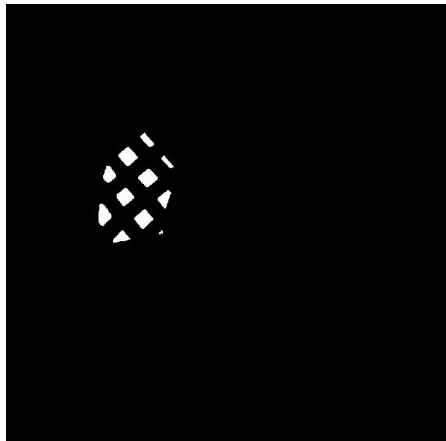
2.8 Průniky mřížky

Během této části skriptu vzniká velké množství dočasných proměnných, které je nejdříve potřeba odstranit z důvodu různého rozsahu v každé sérii pacienta. Pro ilustraci doby výpočtu je také použit waitbar.

2.8.1 Souřadnice průniků mřížky

Průniky se zjišťují pro každý segment zvlášť. Je to z toho důvodu, že pro počítání odpovídajících si pozic v rámci sekvence nebude muset program prohledávat celou plochu snímku, ale jenom průniky v daném segmentu, kterých jsou maximálně desítky.

Pro průnik mřížky je použit podobný postup jako pro její rekonstrukci. Používáme konvoluce s použitím Prewittových operátorů, ale v tomto případě výsledky pro jednotlivé směry sečteme a prahujeme již kombinovaný obrázek. Hledáme průniky oblastí, které mají hodnotu 255, čili hranice dolní propusti filtru byla nastavena jako 260 pro jejich součet. Průniky jsou ilustrovány na Obr. 2.20.



Obr. 2.20: Průniky mřížky po aplikaci masky.

Dále je potřeba najít prostřední bod tohoto průniku, neboli jeho těžiště. Proto byl použit příkaz `regionprops` s parametrem `'centroid'`. Tento příkaz vytvoří matici s pozicemi všech těžišť.

```
for i=1:length(TG_SA1_ROI) % průnik mřížky segmentu basal anterior
    mrizka1_basal_anterior{i}=uint8(conv2(abs(iF1_TG_SA1_ROI{i}), (prew
    itt1/3)));
    mrizka1_basal_anterior{i}=blkdiag(zeros(resize),mrizka1_basal_ante
    rior{i}([2:(sizeROI_SA(1)-resize+1)],[2:(sizeROI_SA(2)-resize+1)]
    ));
    mrizka2_basal_anterior{i}=uint8(conv2(abs(iF2_TG_SA1_ROI{i}), (prew
    itt2/3)));
    mrizka2_basal_anterior{i}=blkdiag(zeros(resize),mrizka2_basal_ante
    rior{i}([2:(sizeROI_SA(1)-resize+1)],[2:(sizeROI_SA(2)-resize+1)]
    ));
```

```

if i==1
    prunik_basal_anterior{i}=(double(mrizka1_basal_anterior{i})+do
    uble(mrizka2_basal_anterior{i})).*maska_basal_anterior{1};
else
    prunik_basal_anterior{i}=(double(mrizka1_basal_anterior{i})+do
    uble(mrizka2_basal_anterior{i})).*(imdilate(maska_basal_anteri
    or{i},expand));
end
prunik_basal_anterior{i}=prunik_basal_anterior{i}>260;
cent_basal_anterior{i} = regionprops(prunik_basal_anterior{i},
'centroid');
centroids_basal_anterior{i} = cat(1, cent_basal_anterior{i}.Centro
id);
end; Perc=Perc+1;
waitbar(Perc/PercMax,WB,sprintf('%d%%',ceil(Perc/PercMax*100)));

```

2.8.2 Seskupení odpovídajících průníků

Po získání pozic všech průníků v daném segmentu ve všech snímcích dané sekvence je potřeba zjistit, které z nich odpovídají pohybu mřížky a jsou tedy na sebe navázané. Je potřeba hledat v okolí bodu snímku v čase i , jestli se v daném poloměru nachází průnik i v snímku v čase $i + 1$. Experimentálně bylo zjištěno, že nejvhodnějším okolím bodu pro hledání je 20 pixelů. Pokud je takový bod nalezen, je přiřazen na stejnou pozici v matici, jakou má jeho odpovídající bod v čase i . Pokud ne, je na pozici přiřazena 0. Z toho důvodu se počet bodů s postupem sekvence snižuje.

```

serazeni_basal_anterior=cell(1,length(F_TG_SAl_ROI)+1);
for i=1:(length(F_TG_SAl_ROI)-1)
    serazeni_basal_anterior{i+1}=zeros(length(centroids_basal_anterior{1}),2);
end
serazeni_basal_anterior{1}=centroids_basal_anterior{1};
for i=1:(length(F_TG_SAl_ROI)-1)
    serazeni_basal_anterior{i+1}=zeros(length(centroids_basal_anterior{1}),2);
    for k=1:length(serazeni_basal_anterior{i})
        for l=1:length(centroids_basal_anterior{i+1})
            dist = rangesearch(serazeni_basal_anterior{i}(k,1:2),centroids_bas
            al_anterior{i+1}(l,1:2),20);
            if serazeni_basal_anterior{i}(k,1)~=0 && serazeni_basal_anterior{i
            +1}(k,1)==0
                if dist{1,1}==1
                    serazeni_basal_anterior{i+1}(k,1:2)=centroids_basal_anteri
                    or{i+1}(l,1:2);
                else
                    serazeni_basal_anterior{i+1}(k,1:2)=0;
                end
            end
        end
    end
end
end; Perc=Perc+1;
waitbar(Perc/PercMax,WB,sprintf('%d%%',ceil(Perc/PercMax*100)));

```

Hledání v okolí bodu se provádí funkcí `rangesearch` s parametrem vzdálenosti prostoru hledaného bodu 20.

2.8.3 Odstranění prázdných pozic v matici

Zpracování výsledků má význam pouze pokud se jedná o úplnou sekvenci. Tudiž pokud se v průběhu sekvence ztratí průnik, je celá tato série bodů bezpředmětná. Kvůli tomu byl vytvořen skript, který takové série eliminuje.

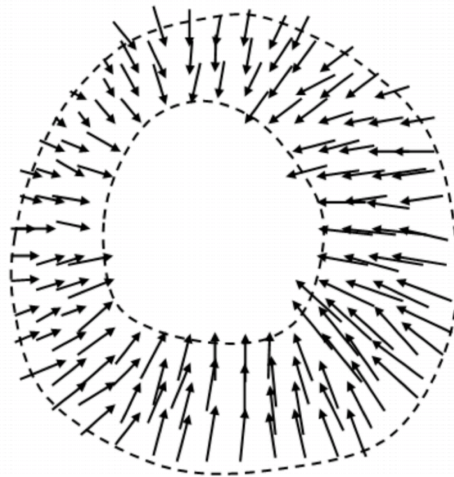
```

delete0=find(serazeni_basal_anterior{length(TG_SA1_ROI)}(:,1)==0);
for i=1:length(TG_SA1_ROI)
    serazeni_basal_anterior{i}(delete0,:)=[];
end

```

2.9 Extrakce pohybového vektoru

Pohybový vektor definuje rozsah pohybu jednoho průniku mřížky během celé sekvence. Počáteční a koncová část vektoru odpovídá nejvzdálenějšímu a nejbližšímu bodu od středu levé komory. Tyto body odpovídají pozici v prostředním úseku systoly a diastoly. Během systoly je levá komora nejvíc kontrahovaná, tudíž průnik mřížky bude nejbliž ke středu. Naopak během diastoly je levá komora relaxovaná, čili bod průniku je nejvzdálenější. V ideálním případě by množina vektorů pokrývala celou oblast levé komory, jak lze pozorovat na Obr. 2.21. Při rekonstrukci reálných dat však rekonstruovaná mřížka ztrácí s ubývajícím kontrastem mřížky své průniky. Proto lze nalézt jenom několik bodů v daném segmentu. U longitudinálních řezů v mnoha případech průniky úplně zanikají.



Obr. 2.21: Ideální podoba pohybového vektoru [19].

Pro zobrazení tohoto typu vektorů je používán příkaz `quiver`, kterého parametry jsou souřadnice počátečního bodu a délka vektoru.

2.10 Deformace levé komory

Deformaci levé komory lze vyjádřit jako relativní vzdálenost pozice průniku mřížky od středu levé komory. Protože ne každý průnik mřížky začíná ve stejné vzdálenosti od středu, bylo potřeba od každého vektoru odečíst jeho minimální hodnotu. Pro relativní zobrazení tedy bereme rozsah v jakém se bod pohybuje s tím faktem, že nejmenší hodnota v rámci sekvence se bude rovnat 0. Na ose x tohoto grafu použijeme časovou informaci

o pořízení daného snímku v rámci sekvence. Tento údaj charakterizuje, ve kterém okamžiku, neboli snímku (frame) byl snímek pořízen. Tuto hodnotu lze vypočítat z metadat, konkrétně z proměnné `RepetitionTime`, která charakterizuje frekvenci pořizování snímků [13]. Postačí tudíž vynásobit pořadí snímku v sekvenci s tímto parametrem.

2.10.1 Graf příčné deformace

Změnu pozice v závislosti na pozici předešlé vyjadřuje tzv. strain. Strain se počítá jako poměr rozdílu pozic v čase T_1 a T_0 a pozici v čase T_0 . Tento vztah vyjadřuje rovnice 2.1 [20]:

$$Strain (\varepsilon) = \frac{L - L_0}{L_0} \quad (2.1)$$

Kde L je vzdálenost bodu od středu levé komory v čase T_1 a L_0 je vzdálenost toho samého bodu v čase T_0 .

Hodnoty pro osu x byl opět použit `RepetitionTime` jako v předchozím případě. Pro apikální řez tomu odpovídá následující část skriptu:

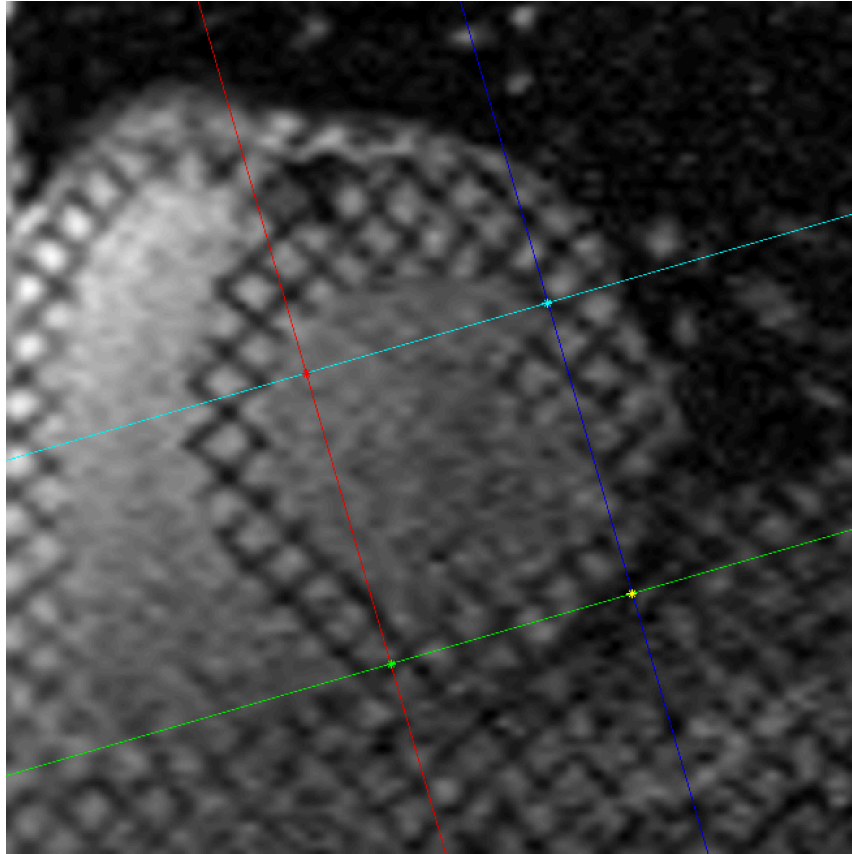
```
for i=1:length(TG_SA3_ROI)
    if i==1
        strain_apical_anterior(i)=0;
        strain_apical_septal(i)=0;
        strain_apical_inferior(i)=0;
        strain_apical_lateral(i)=0;
    else
        strain_apical_anterior(i)=(vzdalenost_apical_anterior{1}(i)-vzdalenost_apical_anterior{1}(i-1))/vzdalenost_apical_anterior{1}(i-1);
        strain_apical_septal(i)=(vzdalenost_apical_septal{1}(i)-vzdalenost_apical_septal{1}(i-1))/vzdalenost_apical_septal{1}(i-1);
        strain_apical_inferior(i)=(vzdalenost_apical_inferior{1}(i)-vzdalenost_apical_inferior{1}(i-1))/vzdalenost_apical_inferior{1}(i-1);
        strain_apical_lateral(i)=(vzdalenost_apical_lateral{1}(i)-vzdalenost_apical_lateral{1}(i-1))/vzdalenost_apical_lateral{1}(i-1);
    end
end
```

Pro zprůhlednění prezentovaných dat byla pro každý segment v řezu zavedena jednotná barva. Následně se grafy prezentují pomocí funkce `plot`.

2.10.2 Graf torzní deformace

Torzni deformace vyjadřuje posun bodů ve směru rotace svalu, hodnotí se pomocí cirkumferenčního strainu, kterého výpočet se provádí pomocí čtyřúhelníku, kterým se proloží 4 body mřížky a zaznamenává se jeho deformace v čase [21].

Z důvodu, že řezy SA1 a SA2 mají segmentů 6, proloží se čtyřúhelníkem body pouze ve vybraných 4 segmentech a proces se ještě dvakrát zopakuje s natočeným čtyřúhelníkem. V praxi to vypadá tak, že vybraný bod ze 4 segmentů se proloží přímkou tak, aby s přímkou procházející dalším bodem svíral úhel 90° . Tuto konfiguraci je možné pozorovat na Obr. 2.22.



Obr. 2.22: Uspořádání kolmic pro pozorování torzní deformace.

Pro sledování deformace tohoto čtyřúhelníku potřebujeme znát pozici průsečíků v čase. To se realizuje pomocí funkcí `polyfit` a `polyval`, pomocí kterých lze získat souřadnice průsečíku.

```
for i=1:length(TG_SA1_ROI)
    x1 = [Bcir1x{i}(1) Bcir1x{i}(end)];
    y1 = [Bcir1y{i}(1) Bcir1y{i}(end)];
    x2 = [Bcir2x{i}(1) Bcir2x{i}(end)];
    y2 = [Bcir2y{i}(1) Bcir2y{i}(end)];
    p1 = polyfit(x1,y1,1);
    p2 = polyfit(x2,y2,1);
    Bx_intersect12{i} = fzero(@(x) polyval(p1-p2,x),3);
    BY_intersect12{i} = polyval(p1,Bx_intersect12{i});
end
```

`for`

Detekcí rozdílů vzdáleností těchto průniků v čase dokážeme určit torzní vychýlení v ose kolmé na proloženou přímku. Jelikož tento proces opakujeme celkově třikrát, v každé sérii zvolíme 2 segmenty pro které budeme torzi počítat.

```
for i=1:length(TG_SA1_ROI)
    if i==1
        Cstrain_basal_anteroseptal(i)=0;
        Cstrain_basal_inferolateral(i)=0;
    else
        Cstrain_basal_anteroseptal(i)=pdist([Ax_intersect12{i},Ax_intersect12{i-1};Ax_intersect12{i-1},AY_intersect12{i-1}], 'euclidean');
        Cstrain_basal_inferolateral(i)=pdist([Ax_intersect34{i},Ax_intersect34{i-1};Ax_intersect34{i-1},AY_intersect34{i-1}], 'euclidean');
    end
end
```

`end`
`end`

Vždy jsou vybrány 2 protilehlé segmenty.

Pro apikální průřez, ve kterém se nacházejí pouze 4 segmenty, se tento proces provede jenom jednou a sleduje se pro všechny průniky najednou.

Graf torzní deformace je prezentován stejně jako graf relativních vzdáleností od středu levé komory. Pro relativní zobrazení vychýlení v dané ose je celý vektor ponížěn o nejmenší hodnotu a pozorují se rozdíly v relativním vychýlení segmentů navzájem. Na časové ose se nacházejí stejné hodnoty jako pro předchozí 2 grafy.

Z důvodu, že série těchto hodnot začíná v nule, pro detailnější zobrazování jsou prezentovány grafy až od pozice v čase T_1 .

```
hold on
plot(TG_SA1_Times(2:end),(Cstrain_basal_anterior(2:end)-min(Cstrain_basal_anterior(2:end))),'color', color_basal_anterior, 'linewidth', 2, 'DisplayName', 'Basal anterior')
plot(TG_SA1_Times(2:end),(Cstrain_basal_anteroseptal(2:end)-min(Cstrain_basal_anteroseptal(2:end))),'color', color_basal_anteroseptal, 'linewidth', 2, 'DisplayName', 'Basal anteroseptal')
plot(TG_SA1_Times(2:end),(Cstrain_basal_inferoseptal(2:end)-min(Cstrain_basal_inferoseptal(2:end))),'color', color_basal_inferoseptal, 'linewidth', 2, 'DisplayName', 'Basal inferoseptal')
plot(TG_SA1_Times(2:end),(Cstrain_basal_inferior(2:end)-min(Cstrain_basal_inferior(2:end))),'color', color_basal_inferior, 'linewidth', 2, 'DisplayName', 'Basal inferior')
plot(TG_SA1_Times(2:end),(Cstrain_basal_inferolateral(2:end)-min(Cstrain_basal_inferolateral(2:end))),'color', color_basal_inferolateral, 'linewidth', 2, 'DisplayName', 'Basal inferolateral')
plot(TG_SA1_Times(2:end),(Cstrain_basal_anterolateral(2:end)-min(Cstrain_basal_anterolateral(2:end))),'color', color_basal_anterolateral, 'linewidth', 2, 'DisplayName', 'Basal anterolateral')

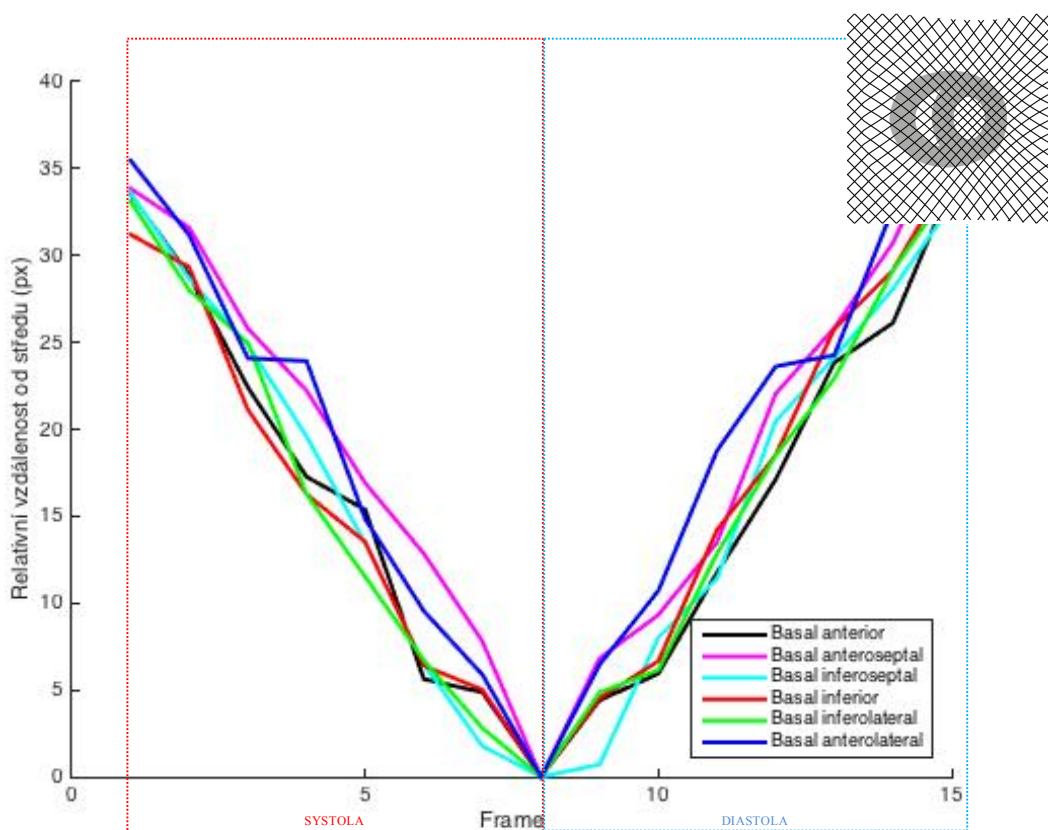
xlabel('Frame sekvence')
ylabel('Torzní deformace')
legend('show')
```

3 Výsledky

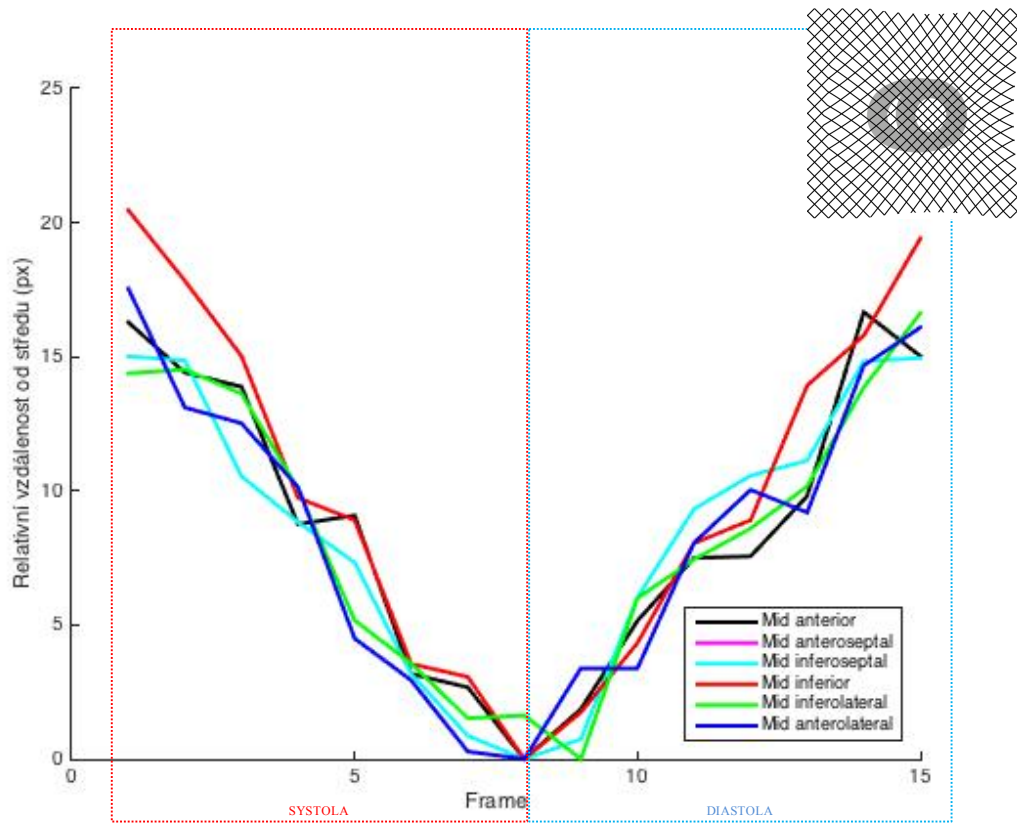
Výsledná data jsou prezentována několika způsoby. Zrekonstruovaná mřížka je nejlépe demonstrována v animaci, proto se nachází na CD nosiči jako příloha.

Deformace levé komory je znázorněna jak pohybovým vektorem, tak grafy deformací pro jednotlivé řezy. Prvním grafem je relativní vzdálenost od středu levé komory závislá na čase. Tento graf mapuje pozici průniku mřížky v čase. Dalším grafem je tzv. radiální strain, který vyjadřuje poměrnou změnu polohy v čase. Graf cirkumferenčního strainu porovnává torzní reformaci mezi segmenty navzájem.

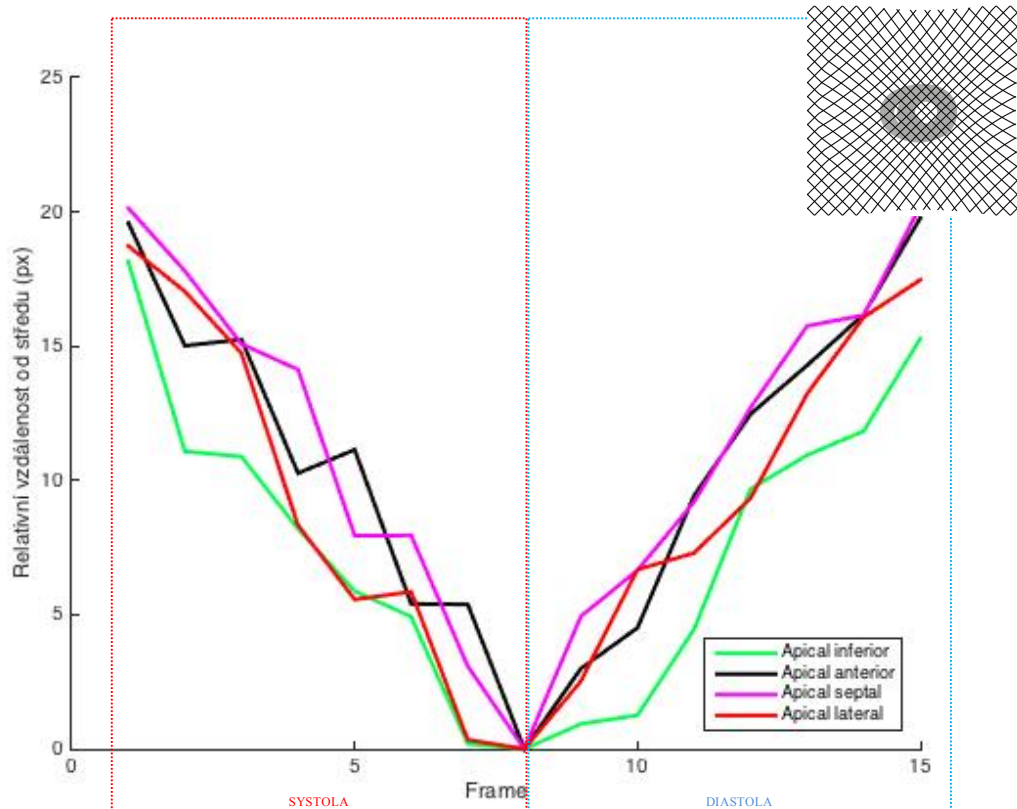
3.1 Výsledky analýzy syntetických dat



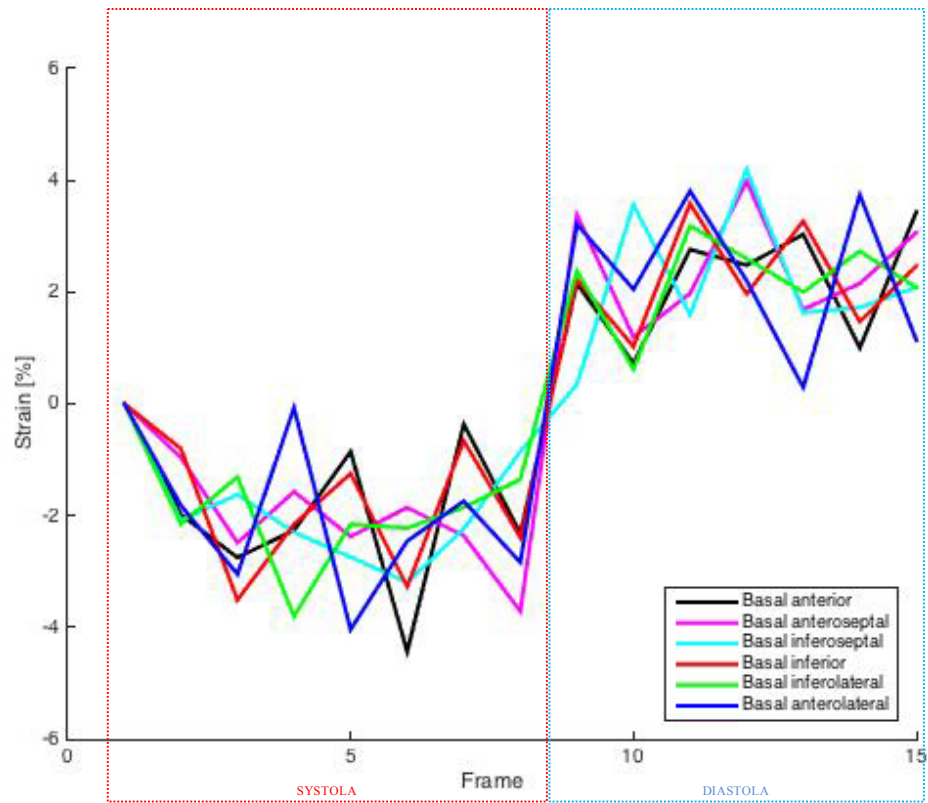
Obr. 3.1: Graf relativní vzdálenosti od středu pro bazální řez.



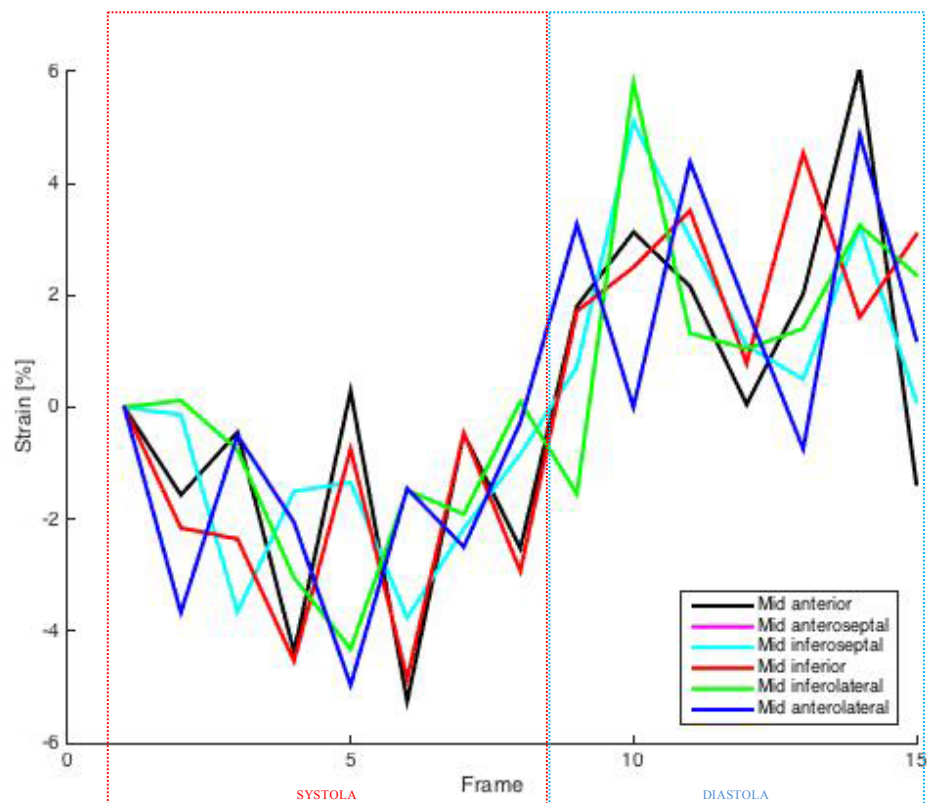
Obr. 3.2: Graf relativní vzdálenosti od středu pro střední řez.



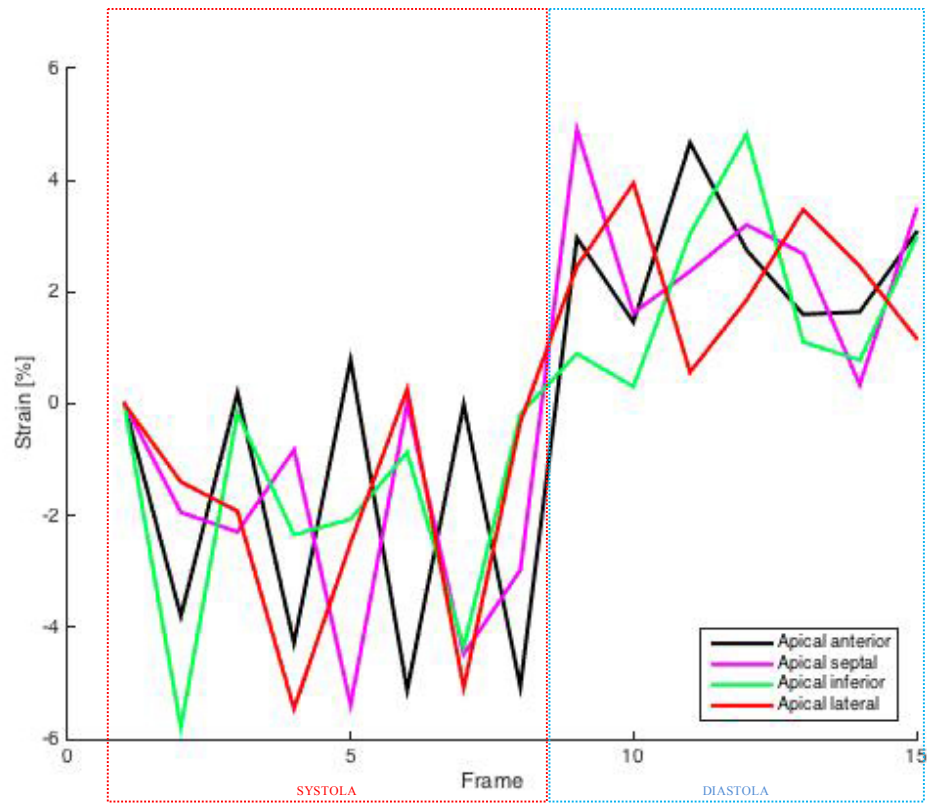
Obr. 3.3: Graf relativní vzdálenosti od středu pro apikální řez.



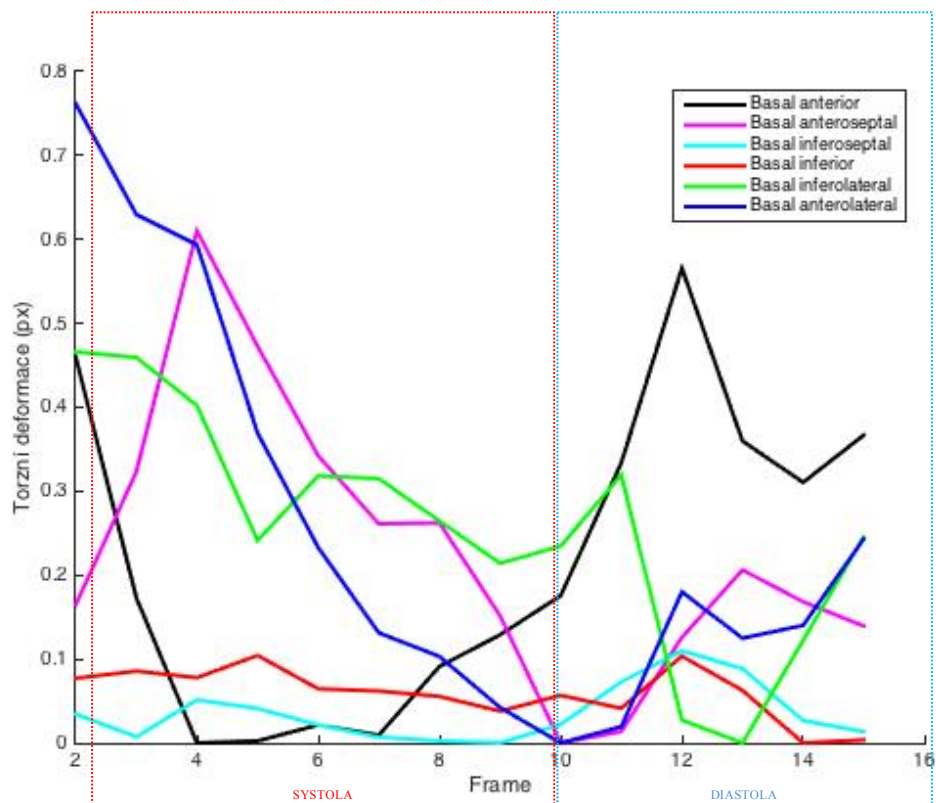
Obr. 3.4: Graf příčné deformace, radial strain bazálního řezu.



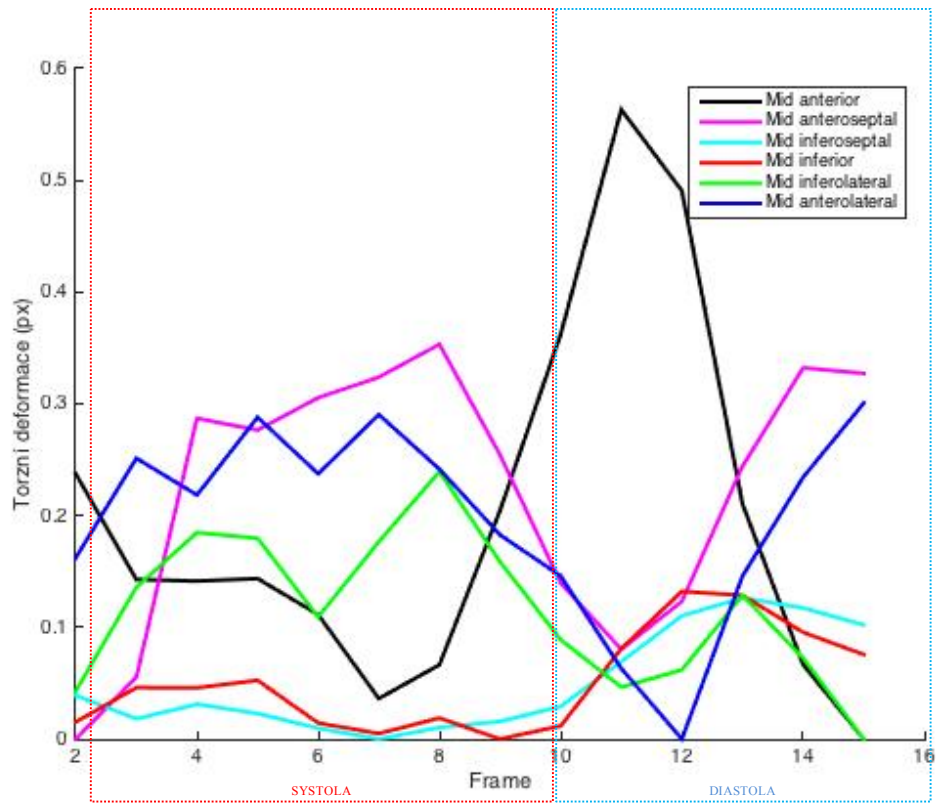
Obr. 3.5: Graf příčné deformace, radial strain středního řezu.



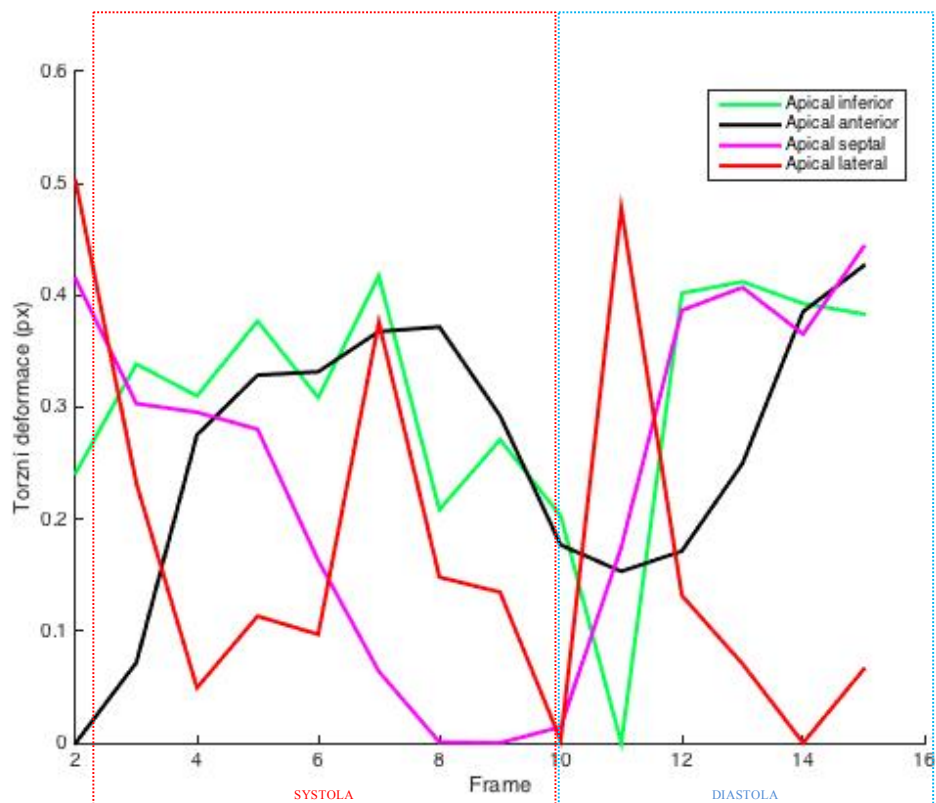
Obr. 3.6: Graf příčné deformace, radial strain apikálního řezu.



Obr. 3.7: Graf torzní deformace, circumferential strain bazálního řezu.



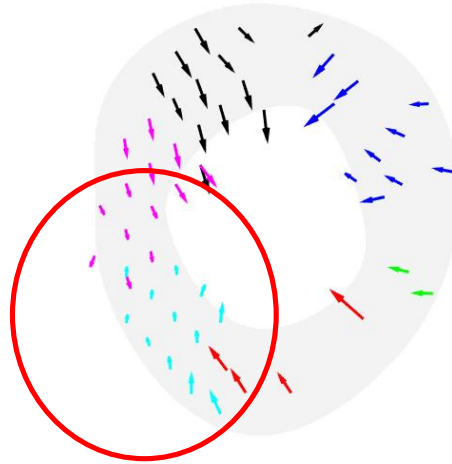
Obr. 3.8: Graf torzní deformace, circumferential strain středního řezu.



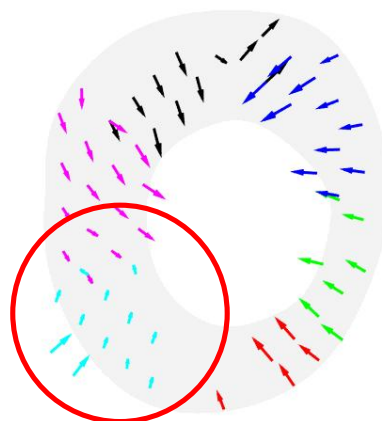
Obr. 3.9: Graf torzní deformace, circumferential strain apikálního řezu.

3.2 Pohybový vektor

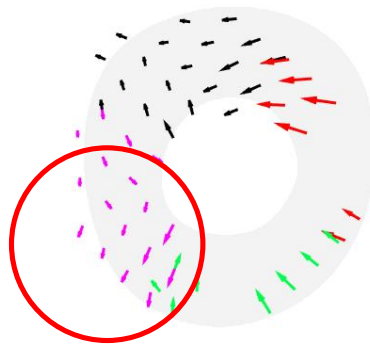
Pomocí pohybového vektoru lze vidět ve kterých částech srdce se nachází postižená oblast. U následujících vektorech byla označena oblast, která se vyznačuje menší aktivitou v porovnání se zbytkem svalu.



Obr. 3.10: Pohybový vektor nemocného v bazálním řezu.

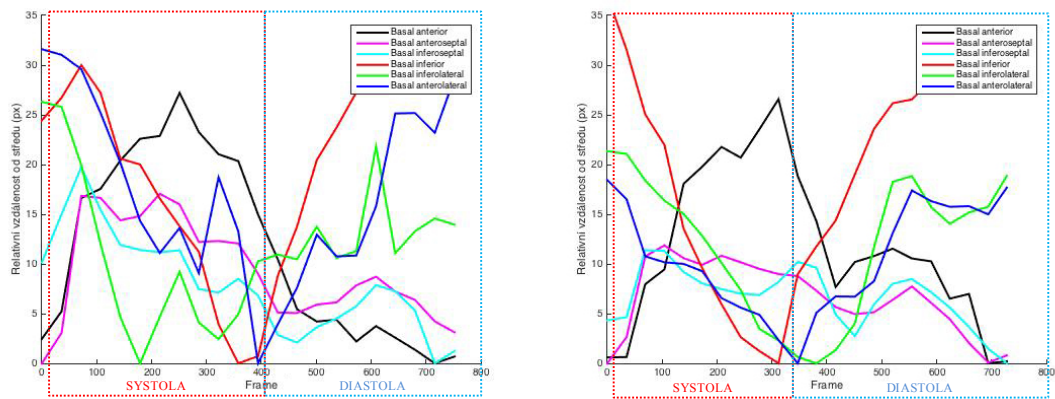


Obr. 3.11: Pohybový vektor nemocného v středním řezu.



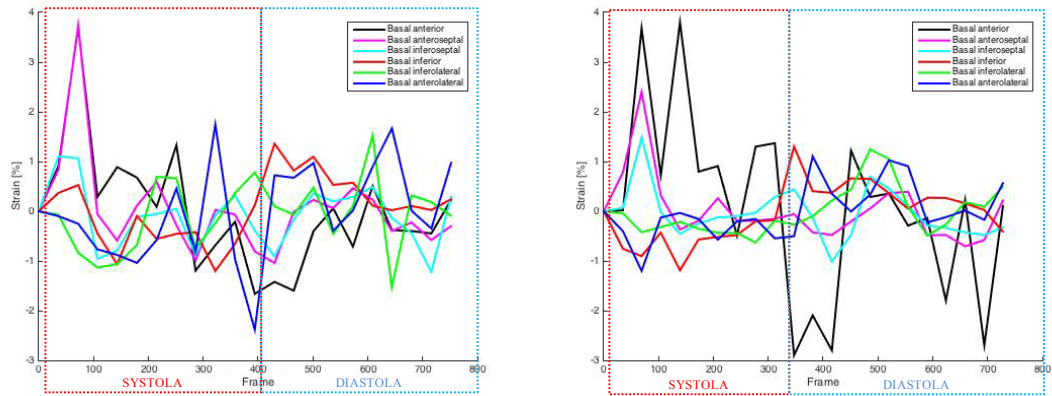
Obr. 3.12: Pohybový vektor nemocného v apikálním řezu.

3.3 Porovnání grafů zdravého a nemocného



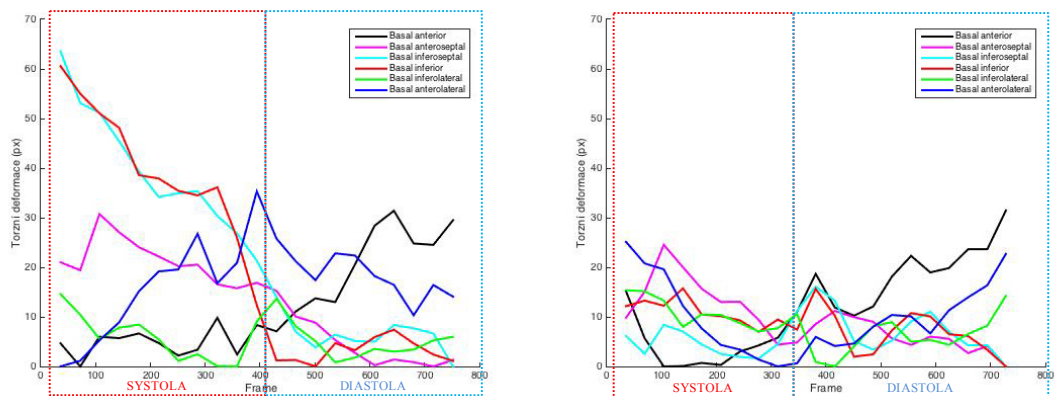
Obr. 3.13: Grafy relativní vzdálenosti od středu pro bazální řez, vlevo zdravý, vpravo nemocný.

I když se jedná o pilotní měření a zobrazení je pouze ilustrativní, můžeme pozorovat, že většina segmentů má velmi podobný průběh. U segmentů Basal inferoseptal a Basal anteroseptal můžeme pozorovat mírnou změnu tvaru vůči zdravému pacientovi.



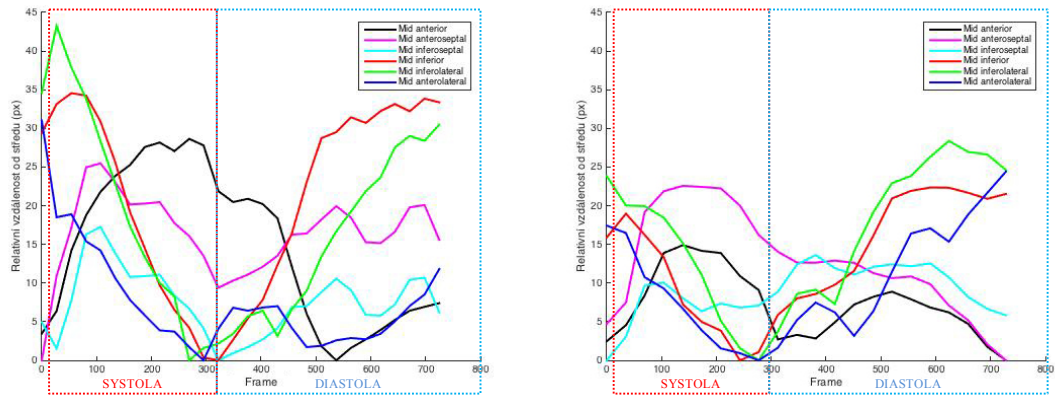
Obr. 3.14: Grafy příčné deformace pro bazální řez, vlevo zdravý, vpravo nemocný.

Z grafu pro nemocného byl odstraněn segment Basal anterior, protože několikanásobně převyšoval rozsah ostatních segmentů a kvůli tomu byl zbytek velmi nepřehledný.

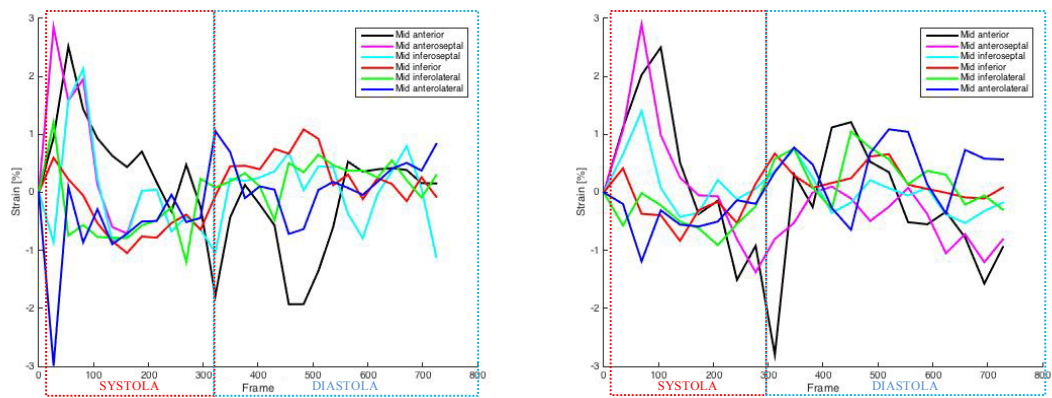


Obr. 3.15: Grafy torzní deformace pro bazální řez, vlevo zdravý, vpravo nemocný.

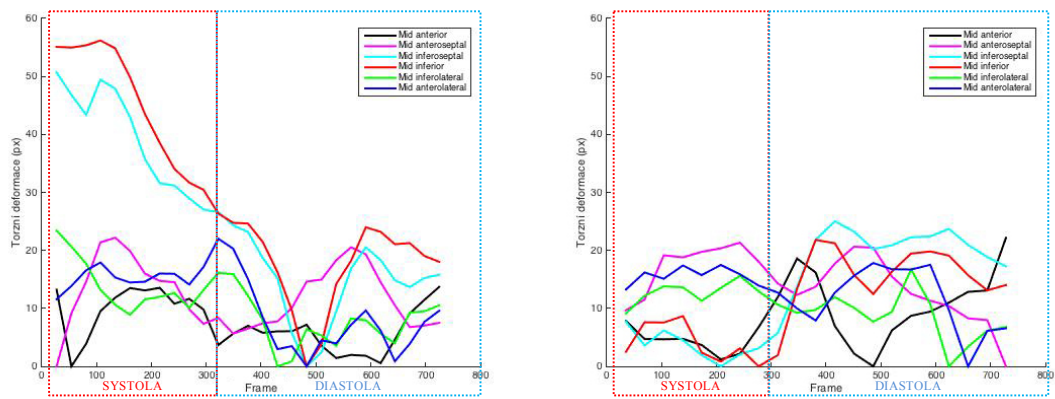
Z grafů je zřejmé, že torzní deformace u zdravého jedince je více než 2 krát větší než u nemocného. Není známo přesně o jakou nemoc se jedná, Mezi různým postižením myokardu se budou grafy od sebe lišit. Zde můžeme předpokládat, že se jedná o zjizvení v oblasti septa, kde je nejvýraznější snížená pohyblivost.



Obr. 3.16: Grafy relativní vzdálenosti od středu pro střední řez, vlevo zdravý, vpravo nemocný.

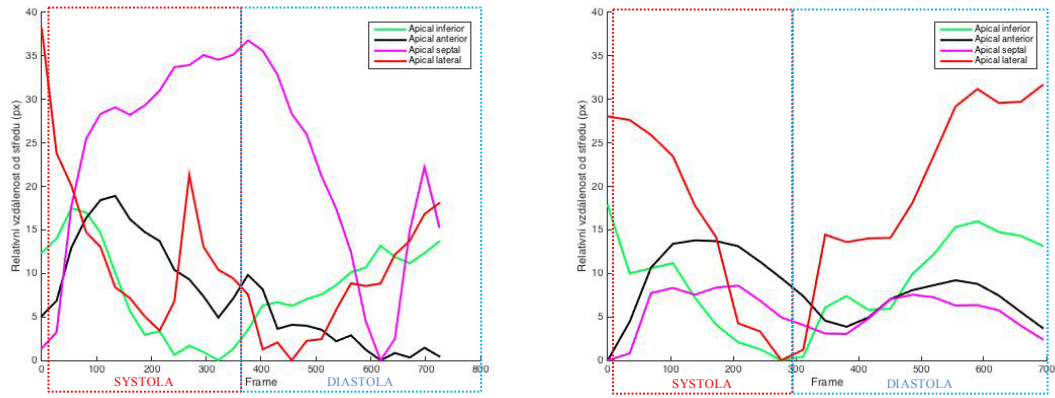


Obr. 3.17: Grafy příčné deformace pro střední řez, vlevo zdravý, vpravo nemocný.

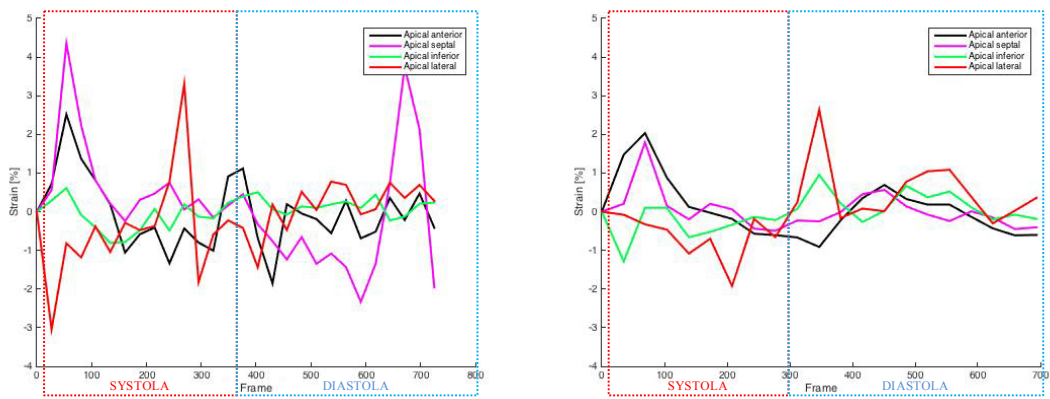


Obr. 3.18: Grafy torzní deformace pro střední řez, vlevo zdravý, vpravo nemocný.

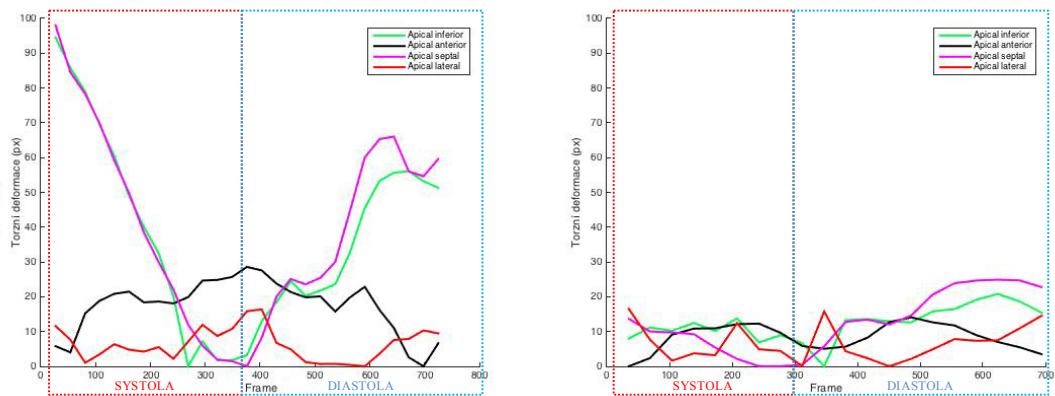
Ve středním řezu můžeme pozorovat rozdíly zejména v segmentu Mid inferoseptal. Zároveň se projevuje snížená torzní deformace.



Obr. 3.19: Grafy relativní vzdálenosti od středu pro apikální řez, vlevo zdravý, vpravo nemocný.



Obr. 3.20: Grafy příčné deformace pro apikální řez, vlevo zdravý, vpravo nemocný.



Obr. 3.21: Grafy torzní deformace pro apikální řez, vlevo zdravý, vpravo nemocný.

Jelikož je se apex projevuje nejvýrazněji torzním pohybem ze všech řezů, je evidentní, že v tomto případě se jedná o jasnou poruchu v segmentu Apical septal.

4 Diskuse

Výsledky zpracování syntetických dat dopadly dle předpokladu. Nejmenší relativní vzdálenost (systola) se nachází přesně uprostřed grafu a názorně tomu odpovídá i graf radiálního strainu. Fakt, že hodnoty fluktuují kolem pomyslné křivky je způsobeno metodou rekonstrukce mřížky. Hodnoty radiálního strainu měly být podle předpokladu nulové, byl však zaznamenán malý torzní pohyb pravděpodobně způsobený vychýlením se rekonstruovaného bodu kolem skutečného průniku na obrázku.

Všechny tři pohybové vektory vypovídají, že se u pacienta jedná o poruchu v septální oblasti myokardu. Lze pozorovat výrazné zkrácení vektorů v porovnání se zachyceným zbytkem.

Pro zjednodušení byly pro každý graf použity stejné barvy pro konkrétní segment. Lze tedy nacházet návaznost i mezi grafy a pohybovým vektorem.

Grafy samy o sobě se velmi neliší, je to způsobeno zejména tím, že se nejedná o souvislé křivky, ale spíš o často se měnící lomenou čáru. Z toho důvodu má zejména strain tendenci fluktuovat kolem jedné hodnoty místo kopírování grafu relativních vzdáleností.

V tomto případě šlo pozorovat jemné rozdíly v segmentech blízcích se septální straně myokardu. Nejprůkaznější byly hodnoty torzní deformace a prezentace pohybovým vektorem.

Metoda rekonstrukce mřížky se však projevuje jako velmi ztrátová. Během procesu se z různých důvodů ztratí průniky mřížky a dochází tak k absenci bodů pro prezentaci výsledků. Tento problém je nejvýraznější u longitudinálních řezů. Ani v jedné sérii testovaných dat, či už syntetických nebo reálných, se nezachoval průnik mřížky ve všech segmentech. Je to způsobeno tím, že prakticky už v polovině sekvence není mřížka vůbec viditelná, proto je Fourierova transformace pro tento typ rekonstrukce nepoužitelná.

U transversálních řezů je problém menší. Ve všech testovaných sériích byly analyzovány všechny segmenty. V některých případech ale mřížka také zmizela a zachoval se jenom jeden bod v segmentu, který se ke konci nepohyboval přesně podle mřížky.

Tato metoda je náročná zejména na obsluhu a to z důvodu potřeby přesného vymezení okrajů levé komory. U některých sekvencí zejména v pokročilejším čase je náročné rozlišit struktury. Proto je hlavním požadavkem zkušenost obsluhy v oblasti zobrazovacích metod.

Během testování se při extrakci okrajů levé komory osvědčilo použití grafického tabletu, který proces nejen zpřesnil, ale také značně urychlil. Nicméně zpracování jedné série snímků trvá v průměru hodinu a je závislé na výkonu počítače.

5 Závěr

Cílem této práce bylo vytvořit program, který by byl nápomocný v případech rozhodování, zdali má pacient podstoupit srdeční resynchronizační léčbu. Jelikož se poruchy převodního systému srdečního projevují mechanickou dyssynchronií segmentů levé komory mezi sebou navzájem, nabízí se jako vhodná metoda monitorace mechanické deformace levé komory.

Za tímto účelem bylo provedeno načtení dat s přídavným gradientem ve formátu DICOM do MatLabu, kde byla následně za použití Fourierovy transformace zrekonstruovaná mřížka kopírující stahy tkáně uvnitř svaloviny levé komory. Vymezení obrysů levé komory bylo provedeno manuální segmentací každého snímku v sekvenci.

Po zjištění pozic průniků zrekonstruované mřížky bylo potřeba prezentovat zjištěné data. Pro vyjádření rozsahu pohybu levé komory byl extrahován pohybový vektor který znázorňuje maximální stah a relaxaci v rámci levé komory.

Pro vyjádření polohy průniku mřížky v čase byl vytvořen graf zachycující relativní vzdálenost bodu od středu levé komory. Na tyto výsledky navazuje graf příčné deformace, kterého hodnoty byly vypočítány jako závislost změny polohy bodu v závislosti na jeho předchozí poloze v sekvenci. Torzní deformace určuje pohyb v rámci jednoho řezu a je vyjádřena jako relativní vychýlení bodu ve směru torze.

Tato metoda se osvědčila pouze pro zpracování příčných průřezů myokardu. Pro podélné snímky je tato metoda nevyhovující.

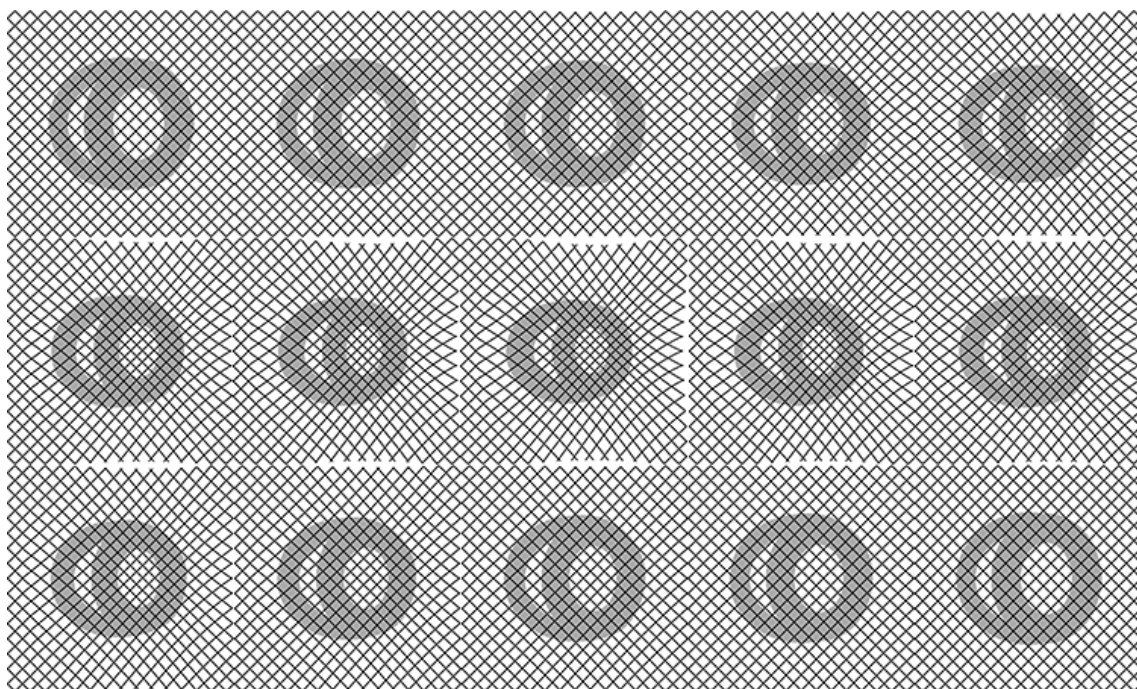
Hlavními důvody omezené efektivity této metody je příliš malý rozměr snímku s přídavným gradientem a příliš krátká doba trvání gradientní mřížky v rámci sekvence. Zejména z těchto dvou důvodů se nedá říct, že by tato metoda byla vždy a za každých okolností použitelná. Velmi závisí na individuálních faktorech jako je zručnost obsluhy a kvalita pořízených snímků.

Seznam použité literatury

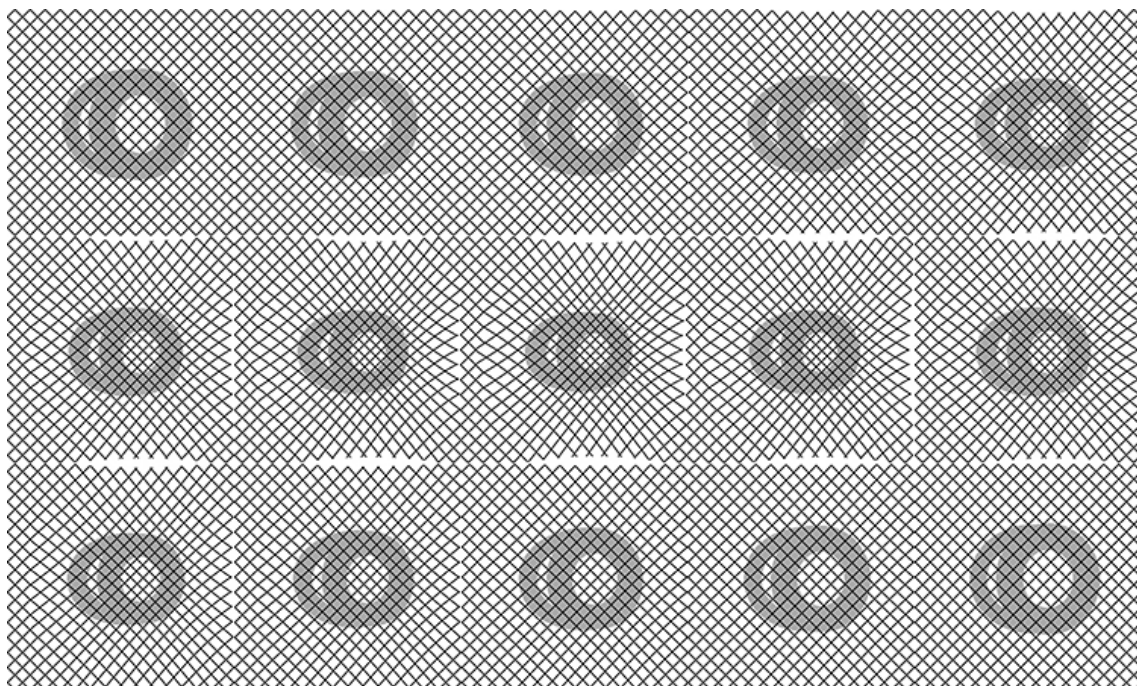
- [1] RIEDLBAUCHOVÁ, Lucie. *Factors modifying effect of cardiac resynchronization therapy in chronic heart failure treatment*. Plzeň, 2004. Habilitační práce. Univerzita Karlova v Praze. 2. Lékařská fakulta. Kardiologická klinika 2. LF UK a FN Motol
- [2] Srdeční resynchronizační léčba. In: *IKEM* [online]. [cit. 2017-05-03]. Dostupné z: <https://www.ikem.cz/cs/srdecni-resynchronizacni-lecba/a-405/>
- [3] M. D. Cerqueira, N. J. Weissman, V. Dilsizian, A. K. Jacobs, S. Kaul, W. K. Laskey, D. J. Pennell, J. A. Rumberger, T. Ryan, and M. S. Verani. *Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart*, *Circulation*, vol. 105, no. 4, pp. 539–542, 2002.
- [4] Left ventricular segmentation [online]. In: . [cit. 2017-05-03]. Dostupné z: <http://cursoenarm.net/UPTODATE/contents/mobipreview.htm?19/13/19675>
- [5] PLEVA, Martin a OUŘEDNÍČEK, Petr. *MRI srdce: praktické využití z pohledu kardiologa*. 1. vyd. Praha: Grada, 2012. ISBN 978-80-247-3931-1.
- [6] SMAL, Ihor, Noemi CARRANZA-HERREZUELO, Stefan KLEIN, Wiro NIESSSEN a Erik MEIJERING. *Quantitative comparison of tracking methods for motion analysis in tagged MRI*. In: 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro [online]. IEEE, 2011, s. 345-348 [cit. 2017-05-05]. DOI: 10.1109/ISBI.2011.5872420. ISBN 978-1-4244-4127-3. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5872420>.
- [7] Osman, N. F., Kerwin, W. S., McVeigh, E. R. and Prince, J. L. (1999), *Cardiac motion tracking using CINE harmonic phase (HARP) magnetic resonance imaging*. *Magn. Reson. Med.*, 42: 1048–1060. DOI: 10.1002/(SICI)1522-2594(199912)42:6<1048::AID-MRM9>3.0.CO;2-M
- [8] Segment. In: *Medviso* [online]. [cit. 2017-05-05]. Dostupné z: <http://medviso.com/products/segment/>
- [9] DIAGNOSIS SOFTWARE. In: *Computer Vision Center* [online]. [cit. 2017-05-05]. Dostupné z: <http://iam.cvc.uab.es/portfolio/diagnosis-software/>
- [10] What is cvi42. In: Circle Cardiovascular Imaging [online]. [cit. 2017-05-05]. Dostupné z: <https://www.circlecvi.com/products.php#filter=cvi42>
- [11] DICOM PS3.1 2017b - Introduction and Overview. In: *Digital Imaging and Communications in Medicine* [online]. [cit. 2017-05-07]. Dostupné z: <http://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf>

- [12] Stretchlim. In: *MathWorks* [online]. [cit. 2017-05-07]. Dostupné z: <https://www.mathworks.com/help/images/ref/stretchlim.html>
- [13] DICOM PS3.6 2017b - Data Dictionary. In: *Digital Imaging and Communications in Medicine* [online]. [cit. 2017-05-07]. Dostupné z: <https://www.mathworks.com/help/images/ref/stretchlim.html>
- [14] A. Histace, C. Cavaro-Menard and B. Vigouroux. *Tagged cardiac MRI: detection of myocardial boundaries by texture analysis, Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, 2003, pp. II-1061-4 vol.3. DOI: 10.1109/ICIP.2003.1246868.
- [15] HISTACE, Aymeric, Christine CAVARO-MÉNARD, Vincent COURBOULAY a Michel MÉNARD. *Analysis of Tagged Cardiac MRI Sequences* [online]. s. 404 [cit. 2016-01-05]. DOI: 10.1007/11494621_40. Dostupné z: http://link.springer.com/10.1007/11494621_40
- [16] Curve Fitting Toolbox. In: *MathWorks* [online]. [cit. 2017-05-09]. Dostupné z: <https://www.mathworks.com/help/curvefit/index.html>
- [17] SENTHILKUMARAN, N.; RAJESH, R. *Edge detection techniques for image segmentation—a survey of soft computing approaches. International journal of recent trends in engineering*, 2009, 1.2: 250-254.
- [18] Marques, O. *Practical image and video processing using MATLAB*. Hoboken: Wiley-IEEE Press, 2011. ISBN 9786613239914.
- [19] IBRAHIM, el-SH. *Myocardial tagging by cardiovascular magnetic resonance: Evolution of techniques—pulse sequences, analysis algorithms, and applications*, *Journal of Cardiovascular Magnetic Resonance*, vol. 13, pp. 1–40, 2011
- [20] ABRAHAM, Theodore P.; DIMAANO, Veronica L.; LIANG, Hsin-Yueh. *Role of tissue Doppler and strain echocardiography in current clinical practice*. *Circulation*, 2007, 116.22: 2597-2609.
- [21] STØYLEN, Asbjørn. *Motion and deformation*. In: *Norwegian University of Science and Technology* [online]. 2015 [cit. 2017-05-15]. Dostupné z: http://folk.ntnu.no/stoylen/strainrate/Basic_concepts.html#Strain_in_th

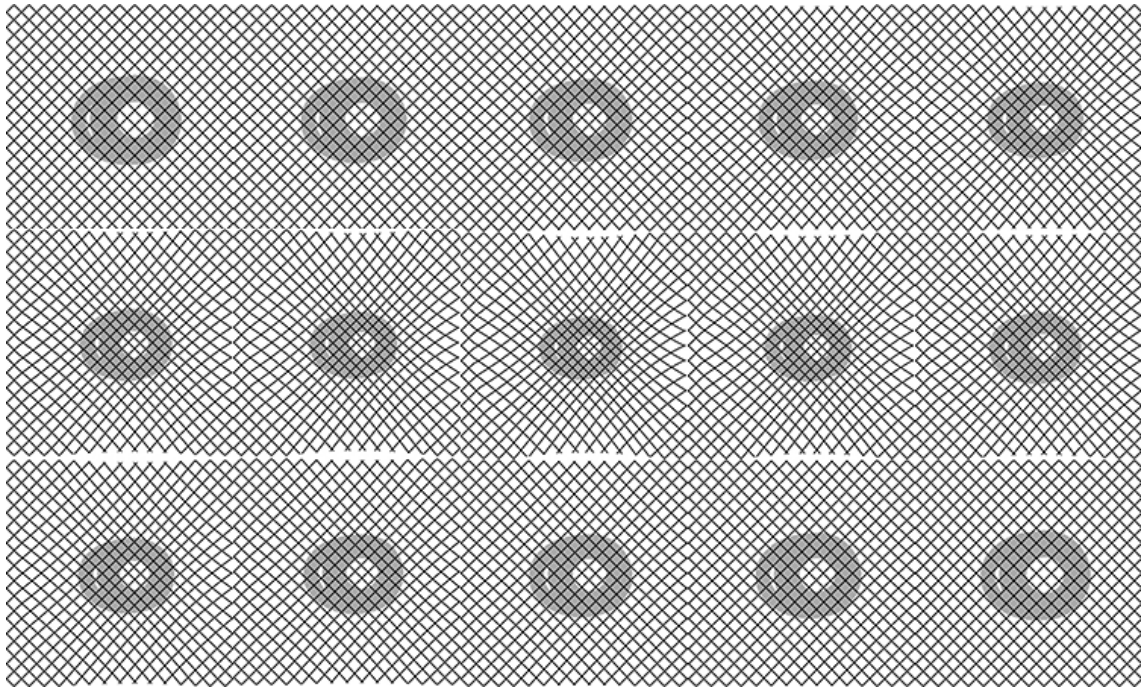
Příloha A: Syntetická data k analýze



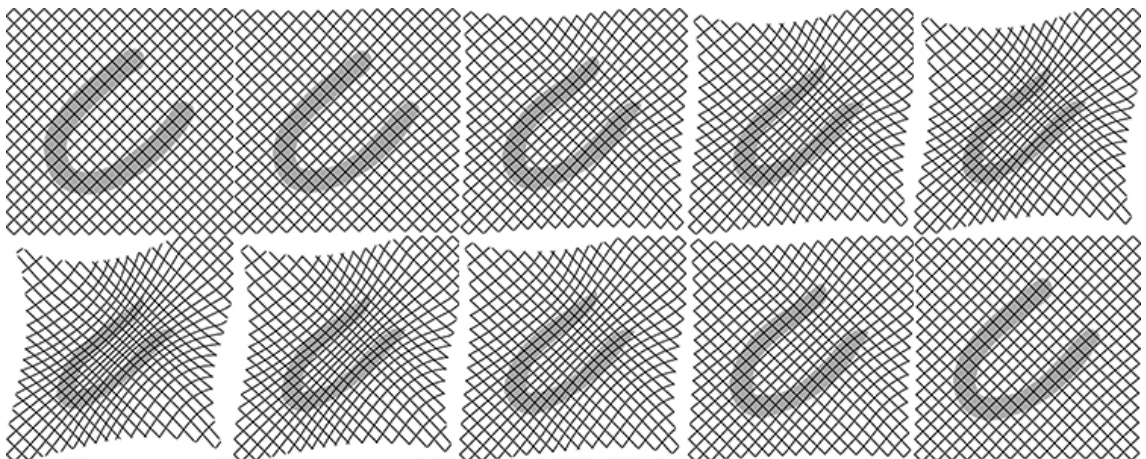
Obr. A.1: Sekvence syntetických snímků v bazálním řezu.



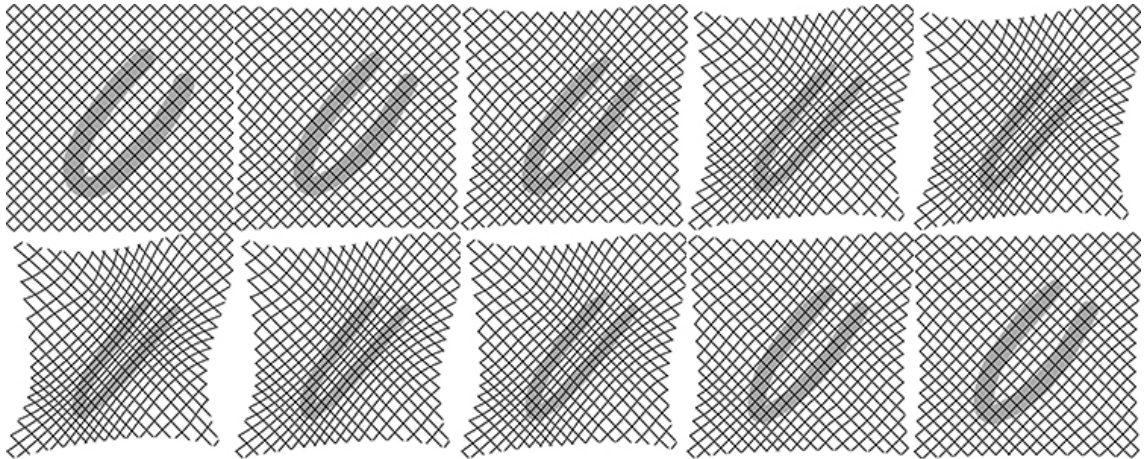
Obr. A.2: Sekvence syntetických snímků v středním řezu.



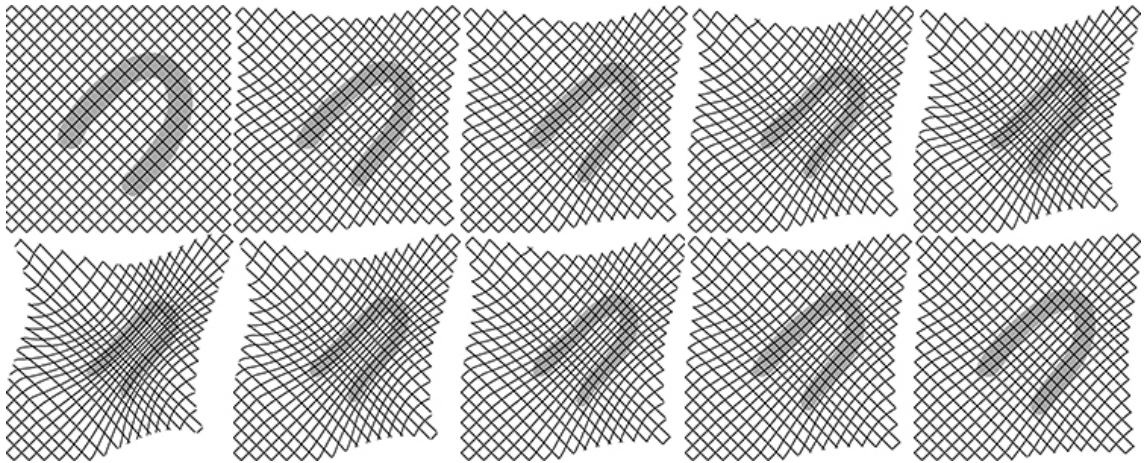
Obr. A.3: Sekvence syntetických snímků v apikálním řezu.



Obr. A.4: Sekvence syntetických snímků v dvoukomorovém řezu.



Obr. A.5: Sekvence syntetických snímků v tříkomorovém řezu.



Obr. A.6: Sekvence syntetických snímků v čtyřkomorovém řezu.

Příloha B: Obsah přiloženého CD

Program v MatLabu,

Data k analýze: Syntetická data,

 Reálná sekvence snímků nemocného,

 Reálná sekvence snímků zdravého jedince,

Videosekvence znázorňující zrekonstruovanou mřížku s reálnými daty.