



THE UNIVERSITY OF QUEENSLAND
A U S T R A L I A

Parallel transmission radiofrequency pulse design for Siemens 7 Tesla Whole-body Magnetic Resonance Imaging (MRI)

by

Yanngyang Xu

School of Information Technology and Electrical Engineering, University of Queensland.

Submitted for the degree of Bachelor(Honours) of Information Technology

November 2017

Yangyang Xu
Yangyang.xu1@uqconnect.edu.au

November 6, 2017

Prof Michael Brünig
Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Brünig
In accordance with the requirements of the Degree of {Bachelor of Engineering (Honours)} in the School of Information Technology and Electrical Engineering, I submit the following thesis entitled

“Parallel transmission radiofrequency pulse design for Siemens 7 Tesla Whole-body Magnetic Resonance Imaging (MRI)”

The thesis was performed under the supervision of Dr Jin Jin. I declare that the work submitted in the thesis is my own, except as acknowledged in the text and footnotes, and that it has not previously been submitted for a degree at the University of Queensland or any other institution.

Yours sincerely
Yangyang Xu

Acknowledgements

I would like to express my sincere acknowledgement and gratitude to my supervisor Dr. Jin Jin, for his patients, invaluable guidance, encouragements, and professional comments. I am also grateful to him for offering me such interesting project and providing me MRI lab experiences.

Abstract

Ultra-high field (UHF) MRI (with static field strength B_0 at or above 7T) has seen rapid development in recent years, since it can increase signal-to-noise ratio for higher resolution anatomical imaging as well as benefits susceptibility weighted imaging (SWI) and fMRI. A radio-frequency coil is used to transmit and receive signals. However, more inhomogeneous transmit/receive fields and potentially dangerous level of specific absorption rate (SAR) are obvious when the radio frequency increases. Inhomogeneous excitation results in images with locally too much or little signal intensities, thereby influencing image quality. High SAR has the risk of body tissue damage. Parallel transmission techniques can alleviate these two problems by controlling field interactions of multiple transmit channels. To help researchers streamline the pulse design process, an efficient, user-friendly, expandable, and editable MATLAB application, ShimTool, is developed in this thesis project, based on the state-of-the-art Siemens research Magnetom 7T scanner.

This thesis first introduces the background and motivations of parallel pulse design and the need to develop an expandable software application. It will also explain the physical theories, experimental workflow, and the MATLAB implementation of the application. Finally, the thesis will in the last two sections (2 and 3) illustrate the results made from the MATLAB application, and conclude with the advantages and potential improvements of the MATLAB application, as well as further work.

Contents

Acknowledgements	3
Abstract.....	4
Contents	5
1. Introduction	7
1.1 Advantages of MRI and UHF MRI	8
1.2 Issues of UHF MRI from Literatures	9
1.3 Prevailing Solutions for Two Main Issues of UHF MRI	9
1.3.1 RF shimming	10
1.3.2 kT-spoke shimming	10
1.3.3 MLS optimization.....	12
1.4 MRI Risk Management	12
1.5 Aim of the Thesis	12
1.6 Thesis Overview.....	13
2. Methods	15
2.1 Electromagnetic simulations	15
2.2 Pulsing strategies.....	15
2.2.1 B ₁ Shimming	16
2.2.2 kT Shimming	17
2.2.3 Evaluations of shim schemes.....	18
2.3 Turbo Spin Echo pulse sequence.....	19
2.4 Experiment	21
2.5 MATLAB Application	24

2.5.1	Application Workflow	26
2.5.2	Object-Oriented Programming	29
2.5.3	Data Import and Export	30
2.5.4	MR Image Processing	32
2.5.5	Sliders	35
2.5.6	ROI Toolbox	36
2.5.7	Shimming Simulation	44
3.	Results	52
3.1	Simulation Results	52
3.1.1	Knee	52
3.1.2	Hip	58
3.1.3	Spine	61
4.	Discussion and Conclusion	65
4.1	Experiment Results Discussion	65
4.2	Users' Perspective	65
4.3	Developers' Perspective	66
4.4	Future Extension	66
5.	References	67
6.	Appendix	69

1. Introduction

In the past decades, the physical process of nuclear magnetic resonance (NMR) had been used in probing composition of tissues inside a biological sample. Meanwhile, the same principle has been used in anatomical and functional imaging, called magnetic resonance imaging (MRI), which has become one of the most important medical imaging modalities nowadays.



Figure 1

In a MRI experiment, the whole scanning process is controlled by a host computer outside of the MRI room. A MRI experiment relies on the manipulation and interpretation of three magnetic fields produced by the scanner, namely the main static field (B_0), radio frequency (RF) field (B_1), and gradient fields. These three fields can excite, interact with and spatially encode protons. Fat, muscle and water are main compositions in body tissues, and they have hydrogen atoms as a main constituent [1, 2]. Therefore, hydrogen atoms are most used in MRI due to its natural abundance. In a classical model of nuclear magnetic resonance, hydrogen atoms submerged in the main field possess the properties of nuclear magnetic moment or spin. And there will be an excess of spin that is parallel to the main field. Larmor frequency is the resonance frequency of magnetic moments as a function of the external magnetic field [1] $\omega_0 = \gamma B_0$, where γ is the gyromagnetic ratio of the atom in question. Therefore, higher B_0 strength are associated with higher Larmor frequency (ω_0), which has important impact on the magnetic and electric fields. To excite the spins, RF pulse generated by the RF coil need to be on the Larmor frequency. Gradient field creates a spatial field gradient on top of the main magnetic field which are used to encode the

magnetic resonance (MR) signal. This is achieved by associating resonance frequency with spatial location [1, 2], as seen in Larmor equation where frequency is a function of external fields. Normally, depending on the imaging body sections, there are various shapes and sizes of RF coils. There are various types of RF coils, such as birdcage coils and saddle coils. Surface coil arrays are becoming popular for UHF imaging applications. An 8-channel surface-type array coil is shown in Figure 1 as an example. Its elements can be used for both transmit and receive. In transmission, RF pulses are applied on resonance with the Larmor frequency. With a parallel transmit system, the amplitude and phase of each transmit channel can be adjusted over time to achieve favorable interference patterns [3]. When such RF pulses are produced, the protons are able to absorb the signal and become excited. After the external RF pulse is removed, the excited protons will go through a relaxation process to return to equilibrium state, while emitting RF energy. The emitted RF signals (echoes) are detected by the receiver coil(s), and then analysed and converted to MRI images [4].

1.1 Advantages of MRI and UHF MRI

1.5T and 3T systems remain the workhorse for diagnosis in clinical settings nowadays, while UHF MRI is mostly used in research environment, with clearly demonstrated advantages [5, 6]. It is likely to become an important, next generation diagnostic tool.

There are significant benefits to use UHF in the MRI, such as higher SNR and more apparent susceptibility contrast [7]. Signal-to-noise ratio (SNR) is the rate of signal intensity over the noise intensity as available from the received echoes. Because the spin polarisation is increased with higher field strength, the SNR is thus enhanced, which can be used to improve image resolution and/or imaging speed. The UHF especially benefits the neurological applications, because of improvements of SNR, contrast-to-noise ratio and spatial resolution. For example, the layers of cerebral cortex can be seen in UHF imaging, and is not present in low field MRI [6]. The UHF also reduces T₂ relaxation, which increase the sensitivity of susceptibility-weighted imaging (SWI), which improve the performance of vascular mapping [6]. Figure 2 [8] shows an example where the higher SNR the stronger susceptibility contrast at 7T provides better depiction of blood vessels in the brain.

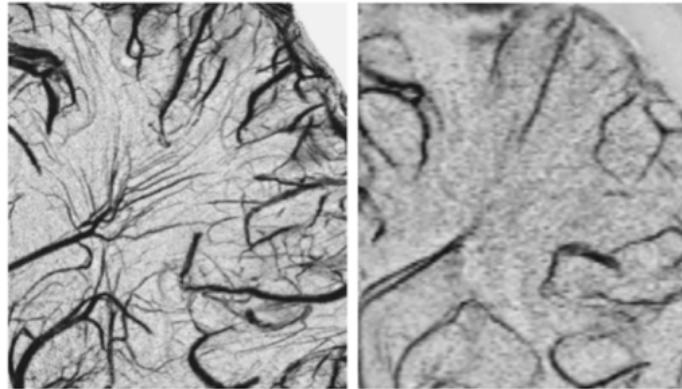


Figure 2

Left is brain SWI in 7T, right is brain SWI in 3T

1.2 Issues of UHF MRI from Literatures

However, some related side effects challenge current exploration of UHF MRI, which are inhomogeneity of the transmit magnetic fields and locally high SAR in the human body. B_1 inhomogeneity will produce regional signal variation and signal losses due to the shortened wavelength [6, 8, 9]. It has been reported that B_1 inhomogeneity at 7T can be at least 1.6 times that at 4T for head imaging [11]. The SAR measures the rate of RF electromagnetic energy absorption by the human body. SAR is generally quadratically proportional to Larmor frequency and hence the magnetic field strength, [5, 6]. Especially in high field, many imaging methods need long pluses and high power, which can increase the SAR.

Currently, the parallel transmission (pTx) application is widely used in UHF MRI, because its ability of controlling the RF fields by adjusting the interference patterns of multiple independent channels (coils) [12]. In such a way, it can help UHF MRI solve the above two main issues (B_1 inhomogeneity and SAR) when the magnetic and electric fields are tailored by using parallel transmission techniques, such as, B_1 shimming and kT-points [8, 12].

1.3 Prevailing Solutions for Two Main Issues of UHF MRI

The RF shimming and kT-spoke shimming are two popular solutions of using parallel transmit (pTx) system to solve the B_1 challenges in UHF [12].

1.3.1 RF shimming

B_1 shimming (or RF shimming) is a basic form of pTx, all channels have same waveform, but the optimised amplitudes (magnitudes) and phases [13, 14]. It has been found that B_1 shimming can reduce SAR and improve inhomogeneity by increasing the excitation efficiency in 7T MRI [15-17]. For example, when a region of interest (ROI) around hip joint cartilage is selected, B_1 shimming can be improve the image quality around ROI and the transmit efficiency. Phase-only shim, which only optimises the phases of transmission coils, is usually very effective in improving the homogeneity of small ROI, such like prostate [18]. Phase only B_1 shimming can also be effective with larger ROI. For example, it can improve the transmit homogeneity of neuroimaging using an 8-channel head coil, as shown in Figure 3, where the signal drop can be improved by adjusting the related phases of the 8 channels [14].

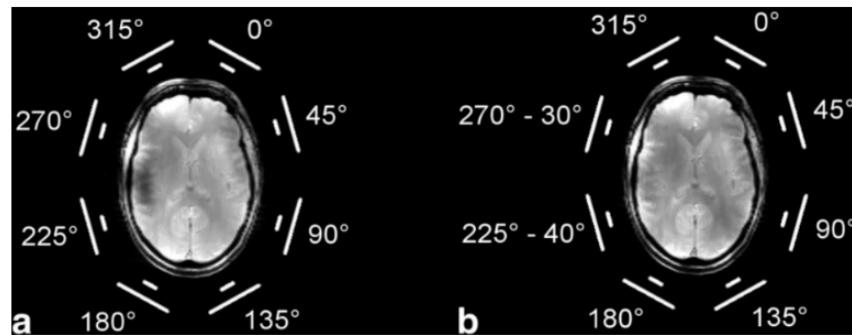


Figure 3 [16]

(a) A head image around by polarized elliptical coil (b) Corrected image of (a)

1.3.2 kT-spoke shimming

kT-spoke is developed from “original transmit k-space trajectory” and inspired by Fast-kz trajectory. kT-spoke (or kT-points) has multiple spokes; this pulsing strategy is an extension of the B_1 shimming. Specifically, a unique setting of phase and amplitude will be created for each spoke similar to a B_1 shimming scheme, and these spokes will be interleaved by short gradient blips. In kT-spoke, both RF and gradient pulses are used to excite uniform flip angle (FA) patterns [23]. Since the multiple RF pulses are used with the magnetic field gradients, kT-spoke has more freedom of controlling excitation than B_1 shimming. Typically, the RF energy is increased and variation of B_1^+ can be better controlled with enhanced B_1^+ inhomogeneity. Larger number of kT-spokes can typically further improve homogeneity while reduce the pulse duration of sub-pulses

[24]. The k-space trajectory and the “time-reversed integration of the gradient waveforms to be played during excitation” are the same thing [20]. The kT-spoke are the locations where the RF signals are transmitted [21]. The k-space trajectory is defined by spoke locations, which is defined by the gradient blips [19]. The location of kT-spoke can be determined by applying inverse Fourier transform of mask that is for selecting a region of interest [22]. However, 5-spoke trajectory cannot resolve sharp phase feature, because the pulse cannot produce accurate FA with such sharp phase feature (phase is 0), inaccurate FA can cause poor inhomogeneity [25]. To mitigate B_1 inhomogeneity, three types of kT-spoke pulses (Fig. 4) are used in the FA-distribution simulation, from the static shim with a single kT-spoke to full Bloch simulation, results illustrate that more kT-spokes can result in less FA variation and provide more homogenous patterns; The Figure 5 reveals that, larger number of spoke and pulse energy can improve B_1 inhomogeneity by decreasing the variation of FA distribution [23].

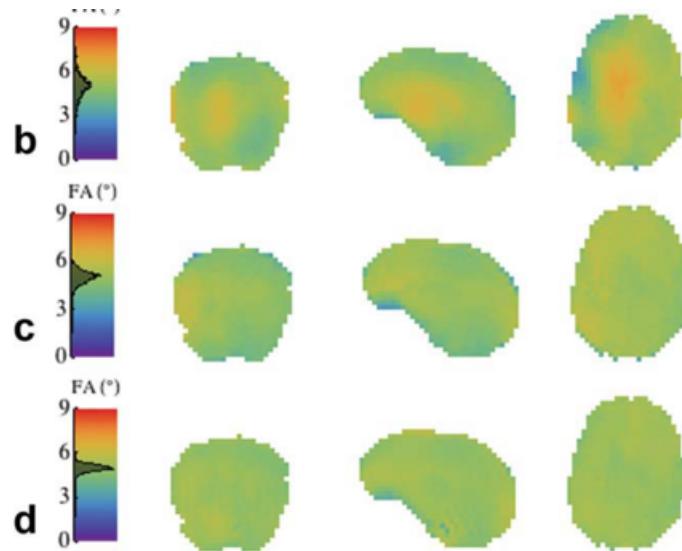


Figure 4 [23]

(b)Static Shim (measured) (c)5 kT-points(measured) (d)5 kT-points (full Blooch simulation)

	RF-Shim	3 k_T -Points IFT	3 k_T -Points GA	5 k_T -Points IFT	5 k_T -Points GA	7 k_T -Points IFT	7 k_T -Points GA
Pulse duration (μs)	500	320	410	430	590	630	810
Bandwidth (kHz)	2.4 ± 0.0	3.5 ± 0.9	3.0 ± 0.8	2.7 ± 0.5	2.0 ± 0.3	1.8 ± 0.1	1.4 ± 0.3
FA variation B_0 ($^\circ \pm \%$)	4.9 ± 17.1	4.9 ± 8.8	4.9 ± 7.7	4.9 ± 6.9	5.0 ± 6.1	4.9 ± 6.1	5.0 ± 5.5
FA variation $B_0 + 100$ Hz ($^\circ \pm \%$)	4.8 ± 17.1	4.9 ± 8.8	4.9 ± 7.7	4.9 ± 6.9	5.0 ± 6.6	4.8 ± 6.5	4.9 ± 6.1
FA variation $B_0 - 100$ Hz ($^\circ \pm \%$)	4.8 ± 17.1	4.9 ± 8.9	4.9 ± 7.7	4.9 ± 7.0	4.9 ± 6.4	5.0 ± 6.4	4.9 ± 6.0
FA variation $B_0 + 200$ Hz ($^\circ \pm \%$)	4.8 ± 17.1	4.9 ± 9.2	4.8 ± 7.8	4.8 ± 7.0	4.9 ± 7.4	4.7 ± 7.7	4.7 ± 7.8
FA variation $B_0 - 200$ Hz ($^\circ \pm \%$)	4.8 ± 17.1	4.9 ± 9.1	4.9 ± 7.9	4.9 ± 7.2	4.7 ± 7.6	4.9 ± 7.2	4.7 ± 8.1
FA variation $B_0 + 300$ Hz ($^\circ \pm \%$)	4.7 ± 17.1	4.8 ± 9.8	4.7 ± 8.0	4.7 ± 7.3	4.8 ± 8.8	4.4 ± 9.8	4.3 ± 11.0
FA variation $B_0 - 300$ Hz ($^\circ \pm \%$)	4.7 ± 17.1	4.8 ± 9.6	4.9 ± 8.0	4.7 ± 7.6	4.5 ± 9.6	4.7 ± 8.6	4.3 ± 12.3

Figure 5

Simulated Results Obtained with Different Excitation Pulses Targeting an Average FA of 5

1.3.3 MLS optimization

In UHF MRI, both RF shim and kT-spoke shim can be combined with the magnitude least-squares (MLS) algorithms. The MLS improves RF pulse weighting by optimising the complex signal based on quantification of magnitude only [20]. Because the image phase is relaxed during optimization of pulse design, the magnitude image should theoretically see further improvement [26].

1.4 MRI Risk Management

There are three main risk controls of MRI that relate to B_0 , B_1 , and gradient field respectively. Patient preparation is critical for MRI studies. Because the electromagnetic induction is a potential risk for high permeability materials, thus these materials are not allowed to be brought into to scanner room during examination. For this reason, MRI may not be suited to people who have metallic materials inside bodies, such as artificial heart valves [27]. High Specific absorption rate (SAR) accumulation, associated with potential tissue heating due to absorption of RF energy, is another potential risk especially at UHF. Currently, MRI system can monitor whole-body SAR in real-time for patient RF safety [5, 6, 8, 10, 27, 28]. Additionally, gradient system produces strong acoustic noise. Therefore, earplugs and headphone will be provided and protect patients from the loud banging noise made from gradient switching. With proper preparation, risks of MRI scans can be minimized [27].

1.5 Aim of the Thesis

The ability to simulate a resultant B_1 profile by combining channel-individual B_1 maps using various shimming strategy is necessary for designing RF pulses in UHF MR imaging. When

combined with appropriate numerical optimisation scheme, it can help researchers improve their pTx pulses. In the UHF MRI, two pulsing strategies, namely the B1 shimming and kT-spoke, are used widely in designing pTx pulses. While the vendors typically supply tool-box to facilitate pTx pulse design [29], these tool box are however limited in function or ease of use. Various research sites have their own implementation of the algorithms. However, researchers typically spend time on adjusting software codes for each experiment, rather than testing and analysing the imaging sequences and protocols. Furthermore, it is not convenient to compare various shimming algorithms beforehand and create the machine-recognisable shim setting files. In this project, a standalone, efficient, user-friendly, expandable and editable integrated solution will be provided in this project, by using a MATLAB object-oriented application with GUIs. application with user-friendly interface is designed and developed. This tool provides a powerful ROI editing function and is able to export calculated pulses to file that can be recognized by the scanner directly. It provides basic functions of B1 shimming and kt-spoke pulse design, while allowing users to further incorporate their own algorithms. This app will help researchers perform, evaluate and compare simulation before being applied in imaging. This app is tested with the Siemens Magnetom 7T research MRI, located at the Center for Advanced Imaging, UQ, St Lucia.

1.6 Thesis Overview

The remainder 2 sections will be talked as follows.

Section 2, the approach and executions for this project. There are four parts in this methods section:

(1) electromagnetic simulations.

(2) The second is about parallel transmit pulsing shimming schemes and related quality evaluation methods.

(3) The experiment workflow

(4) MATLAB application, including the details of application structure, image processing methods, and the use of the theories from the first two parts.

Section 3, the simulation results of knee, hip and spine will be displayed. These results will be gotten by actual MRI machine and this MATLAB application, ShimTool.

Section 4, it will discuss results of experiments, and conclude the advantages and disadvantages of the MATLAB application from two perspectives.

2. Methods

For the following methods, the physic theories will be covered in first two parts, which will be used as the primary idea of the MATLAB application, ShimTool. The experiment workflow will be discussed in 2.3 section.

2.1 Electromagnetic simulations

The shimming algorithms are first developed with simulated transmit magnetic fields, before tested with acquired fields in volunteer imaging. To simulate the electromagnetic simulations, finite-difference-time-domain (FDTD) method are used. The formula can be the partial-differential representations of magnetic (B) and electric (D) flux density in Maxwell's equation are:

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$

$$\frac{\partial \mathbf{D}}{\partial t} = +\nabla \times \mathbf{H} - \sigma \mathbf{E} \quad (1)$$

where the E and H are the electric field and magnetic field intensities respectively, and the σ is the electric conductivity [3].

The time-harmonic solution of the magnetic fields H from simulation are extracted and used in the development of the B1 shimming and kt-spoke algorithm design. The linear combination of transverse magnetic fields are need to represent the transmit magnetic fields:

$$B_1^+ = \frac{B_{1,x} + iB_{1,y}}{2}$$

where $B_{1,x}$ and $B_{1,y}$ are x- and y-components of the magnetic fields.

2.2

Pulsing strategies

There are two types of pulsing strategies studied in this project, namely B_1 shimming and kT shimming.

2.2.1 B_1 Shimming

There are three variants of B_1 shimming, namely amplitude-phase shim, amplitude-phase shim with MLS method, and the phase-only shim are three schemes.

$$B_1^+ = \frac{B_x + iB_y}{2} \quad (3)$$

The B_x and B_y are two components which on the x plane and y plane, $i = \sqrt{-1}$ [30].

In the amplitude-phase shim, different amplitudes and phases will be set for each of 8 channels. The final transmit magnetic field B_1^+ will be channel weighted linear combination of the individual fields :

$$\begin{aligned} \omega_k &= |\omega_k| e^{i\theta_k} \\ B_1^+ &= \sum_{k=1}^n (B_1^+)_k \omega_k e^{i\theta_k} \end{aligned} \quad (4)$$

where the k is the channel index, the n is the total number of channels, ω_k is the complex weighting of the k-th channel ($|\omega_k|$ is the amplitude, $e^{i\theta_k}$ is the phase of the k-th channel), and $i = \sqrt{-1}$ [12]. The optimal weighing can be obtained by optimisation:

$$w = \operatorname{argmin}_w \{ \|B_1^+(w) - b\|_2^2 + \lambda F(w)\}$$

$F(w)$ is a cost function describing constraints, such as transmission power; and λ is the lagrangian multiplier. b is the target magnetic fields [31].

If the amplitude-phase shim uses the MLS method, the final transmit magnetic field B_1^+ will be obtained within multiple iterations of updating the weighting. Typically, the MLS calculation is terminated when the target tolerance is reached or the number of iterations is exhausted. The MLS formula is:

$$w = \operatorname{argmin}_w \{ \| |B_1^+(w)| - b \|_2^2 + \lambda F(w) \} \quad (5)$$

Here, $|B_1^+|$ is the magnitude of the combined transmit magnetic fields.

In the phase-only shim, only the phases of the transmit magnetic fields of all channels are adjusted. Their amplitudes will remain identity and not changed, the formula is:

$$B_1^+ = \sum_{k=1}^n (B_1^+)_k e^{i\theta_k} \quad (6)$$

where the k is the initial channel 1, the n is the number of channels, there is no amplitude, the $e^{i\theta_k}$ is the phase of a channel of 8 channels, and $i = \sqrt{-1}$ [12].

2.2.2 kT Shimming

The kT shimming consists of multiple radiofrequency pulses. It can be treated as a more complex form of B_1 shimming, and the design of individual transmit RF B_1^+ has a similar way to that of the amplitude-phase shimming scheme. The multiple pulses are interleaved by gradient blips, as illustrated in Figure 6. These gradient blips create a k -space trajectory, which can provide additional degrees of freedom to correct B_1 transmission inhomogeneity.

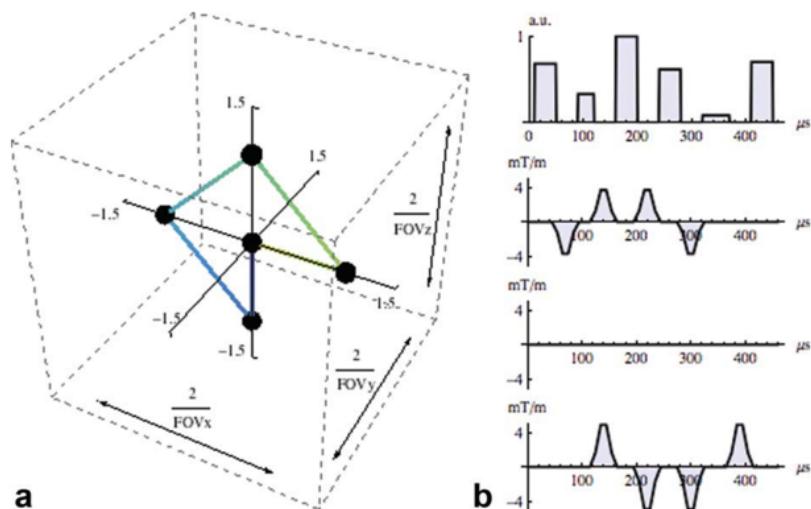


Figure 6
kT Graph

Similarly, the MLS optimisation can be applied to help find the B₁ shimming parameters. To solve the MLS problem, local variable exchange method is also needed, following the formula below:

$$w = \operatorname{argmin}_w \{ |||B_1^+(w, g)| - b||_2^2 + \lambda F(w) \} \quad (7)$$

where w is the complex weighting of channels as described previously; $|B_1^+|$ is the magnitude of the combined transmit magnetic fields with kT pulse, g is the gradient of kT shim; b is the target magnetic fields; $F(w)$ is a cost function describing constraints, such as transmission power; and λ is the Lagrange multiplier [31].

2.2.3 Evaluations of shim schemes

Transmission efficiency and inhomogeneity of a shimming result will be used to evaluate the quality of the shimming scheme. Normally, shimming is used in the region of interesting (ROI), thus this area will be used to calculate the shims which will then be evaluated. The following cases consider a shimming result of an N channels coils, and there exists an ROI mask, M (M spatial locations).

A new combined B_1^+ matrix (combination of N channels) of a shimming scheme with Mask M is formed by Cw , Where C is an $M \times N$ matrix containing the B_1^+ of M points for N channels, and w is the vector of complex weighting coefficients. To expand the Cw :

$$Cw = CC^H w w^H M^{-1} = w^H \Gamma w \quad (8)$$

Where Γ is correlation matrix $M^{-1}CC^H$, H means the conjugate transpose.

Therefore, for parallel transmission, a B₁ shimming efficiency formula that comes from the RF power is:

$$\eta = \frac{\text{average} \|B_1^+\|^2}{\text{average} \|B_{p1}^+\|^2} = \frac{w^H \Gamma w}{w_p^H \Gamma w_p}$$
(9)

Where $\|B_1^+\|^2$ means the square combined B_1^+ of a shimming scheme in ROI, $\|B_{p1}^+\|^2$ means the square combined B_1^+ of a phase-only shimming scheme in the same ROI, and w_p is a phase-only weight.

For the inhomogeneity, the root-mean-square error (RMSE) of a B_1^+ shimming is used to determine:

$$\Delta \|B_1^+\| = \|B_1^+\| - \text{average} \|B_1^+\|$$

$$RMSE = \sqrt{\text{average} \Delta \|B_1^+\|^2}$$

$$Inhomogeneity = \frac{RMSE}{\text{average} \|B_1^+\|}$$
(10)

2.3 Turbo Spin Echo pulse sequence

For the pTx coils in B_1 shimming, a prototype gradient echo field map has a turbo spin echo (TSE) with the proton density (PDw), proton density with fat saturation (PDw-FatSat), T1 and T2 weighting are used to obtain in-vivo images. For the knee, its image protocols are shown in Figure 7.1, and the similar B_1 mapping data in this project, is used in the literature [29]. Top row of Figure 7.2 shows the results of PDw, PDw-FastSat, T1 and T2 weighting image, bottom row shows the corresponding original images. Figure 7.3 shows the PDw image of spine, this spine and knee will be used to apply B_1 shimming.

	TSE		
	PDw/ PDw-FatSat	T ₁ w	T ₂ w
Acquisition time, min:s	3:40	4:09	3:47
Data matrix	512	512	512
Resolution, mm	0.3 × 0.3	0.3 × 0.3	0.3 × 0.3
Slice thickness, mm	2.5	2.5	2.5
FOV, mm	160	160	160
Number of slices	21	21	21
Flip angle	180°	180°	180°
Bandwidth, Hz/pixel	271	271	271
TE/TR, ms	26/5200	13/1100	105/9000
Turbo factor	11	2	2
Grappa	2	2	3
Measurements	1	1	1

Figure 7.1



Figure 7.2

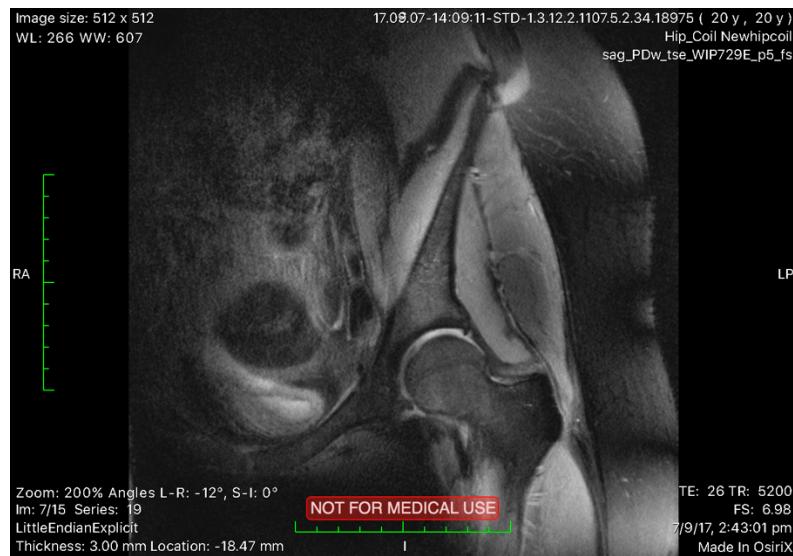


Figure 7.3

2.4 Experiment



Figure 8.1
Scan Knee



Figure 8.2
Scan Hip

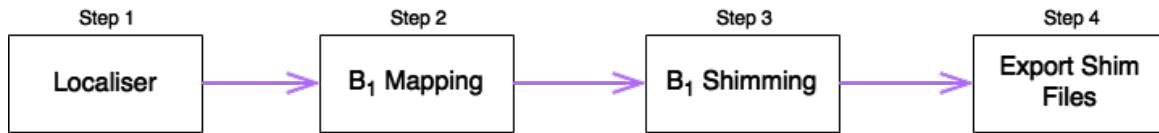


Figure 8.3
Progress

After the volunteer is prepared with correct postures in MRI, such as shown in Figs. 8.1 and 8.2 for knee and hip imaging, Normally, a series of preparations are performed, including Localiser, B0 shimming, and mapping of individual transmit magnetic fields.

Localiser scans typically include sagittal, coronal, and axial images of multiple slices. Figure 8.4 shows a localizer scan of a knee joint as an example. B1 mappings and anatomical scans can then be readjusted to better follow the anatomy of the subject. For example, the yellow box frames demonstrate the adjusted imaging area as planned on the localizer images. After locating positions of scan, the next step is acquiring the related B1 mapping (Fig. 8.5) by a fast turbo-flash based mapping sequence. In this project, all experiments use 8 channel coils. Hence, individual magnitude B1 maps are available by transmitting slice-selective sinc-shaped magnetization

preparation pulse from individual channels consecutively. Individual phase B1 maps are obtained by processing a series of turbo-flash images by using individual transmit channel one by one. After obtaining B1 magnitude and phase maps, the calculation of an appropriate shim can be achieved by using the designed software package, ShimTool. The details of application use and implementation methods will be discussed in the next section.

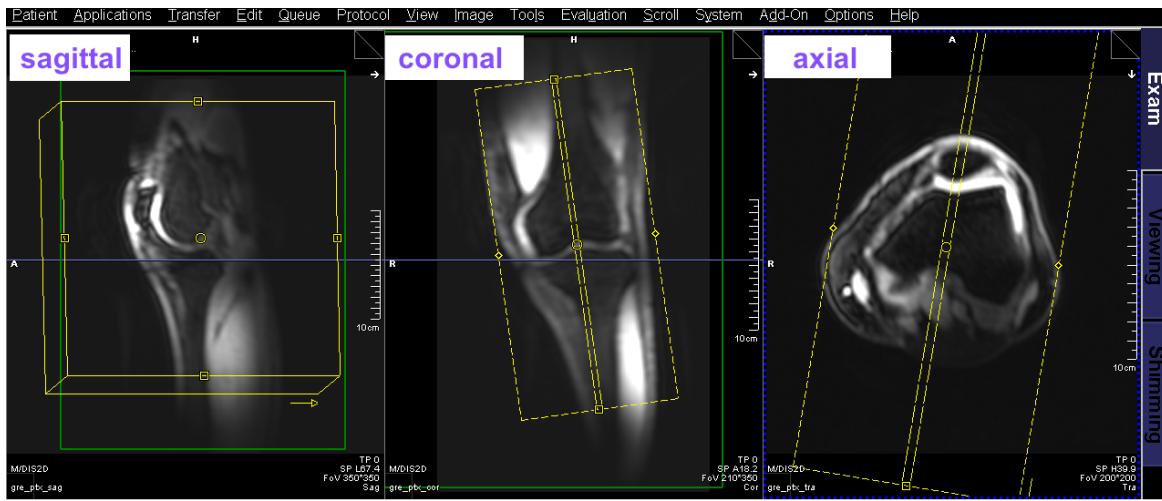
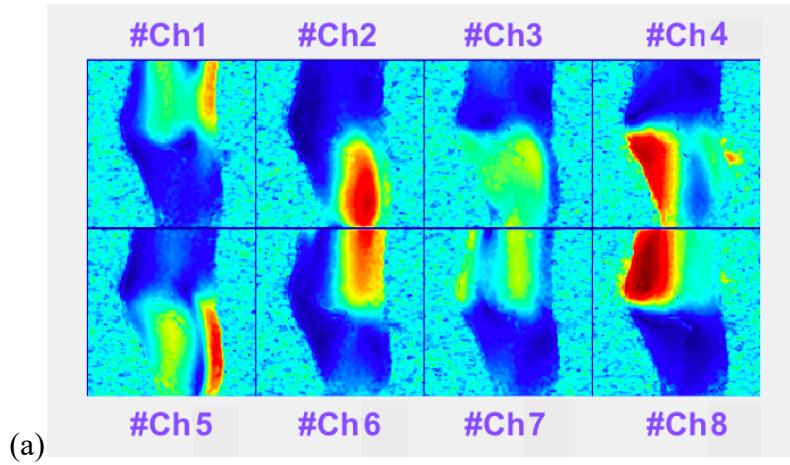


Figure 8.4
Localiser



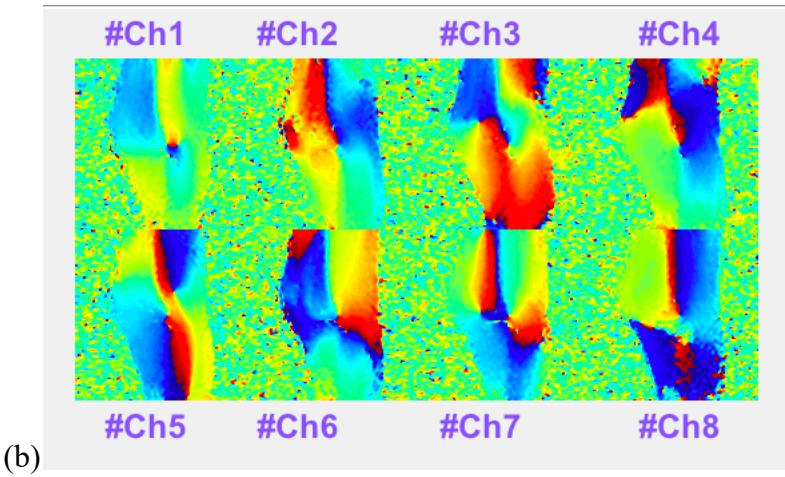


Figure 8.5
 (a) Magnitude B1 map of Slice 6 (b) Phase B1 map of Slice 6

2.5 MATLAB Application

ShimTool is an integrated and object-oriented software. Whole application was programmed in MATLAB R2017a and compiled by its own compiler (The Math Works Inc., Natick, Massachusetts, United States). ShimTool is contained in one folder (Fig 9.1), people can easily copy it into any path. This application has been used to test; the results will be mentioned in Section 3.

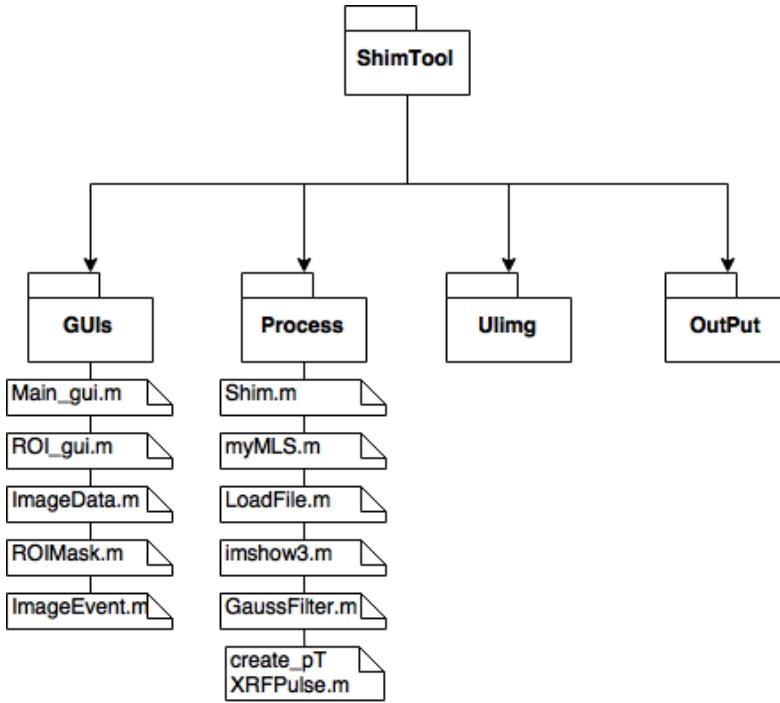


Figure 9.1

GUls includes the MATLAB class files that about interfaces; Process includes the function files used by the interfaces; UIImg included whole icons used; OutPut is a place for getting export shimming schemes.

Interfaces of ShimTool are built by programming, the graphical user interface (GUI) components of Main_gui.m and the ROI_gui.m use the App Designer and GUIDE respectively, which are user interface (UI) components supported by MATLAB, but still have some differences (Fig 9.2). For example, to present a window with a button, the code (Table 1) and GUI style are different.

App Designer	<code>fig = figure('Name','GUIDE');</code> <code>uicontrol('Parent',fig,'String','Button');</code>
GUIDE	<code>fig = uifigure('Name','App Designer GUI');</code> <code>uibutton('Parent',fig, 'Text', 'Button');</code>

Table 1
MATLAB Code

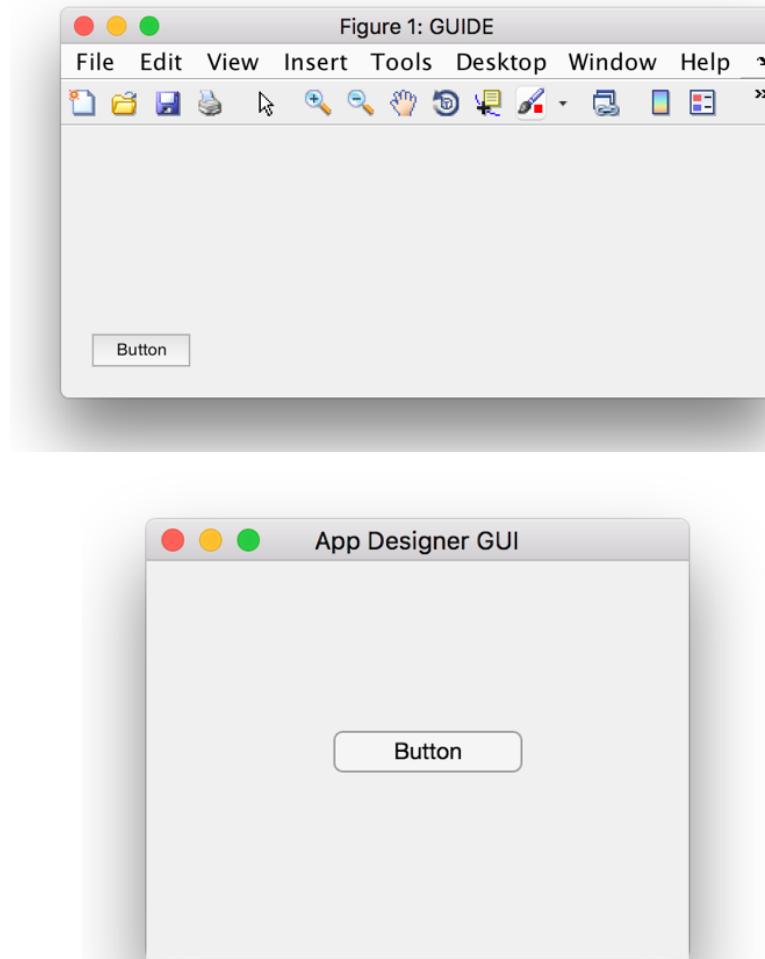


Figure 9.2
GUI examples

App Designer is 2017 year version, its GUI components are more modern than GUIDE. Since it is under development and not fully developed for certain functionalities, the window of the ROI toolbox still needs the GUIDE. Since they are both coded in MATLAB, a window built from App Designer can call a window built from GUIDE, but components cannot be mixed to use.

2.5.1 Application Workflow

After users deployed the ShimTool folder in a path, they can open the MATLAB application. Find the ShimTool folder on the MATLAB “Current Folder” window. Figure 10.2 shows the workflow

of application usage. Once the folder is located and opened, click and access the “GUIs” folder. To start the app, just run the “gui_Main” file (Fig 10.1) To open the medical images, click “patient” icon (Fig 10.3) and select the raw-image folder, then the app will ask for setting the number of channels (Default is 8). Loading image on the panel takes a bit of time. Once the image appealed, users can do shimming experiments. Click the “Draw ROI” icon in the tab “EDITOR” (Fig 10.4); then users can draw the ROIs by the tools, which displayed on the top of “ROI Tool Box” window. Once the ROIs are created and pined, users may wait for a moment to let computer process. After the main panel is updated by the shimming results, users can choose to do more shimming experiments or export the shimming schemes.

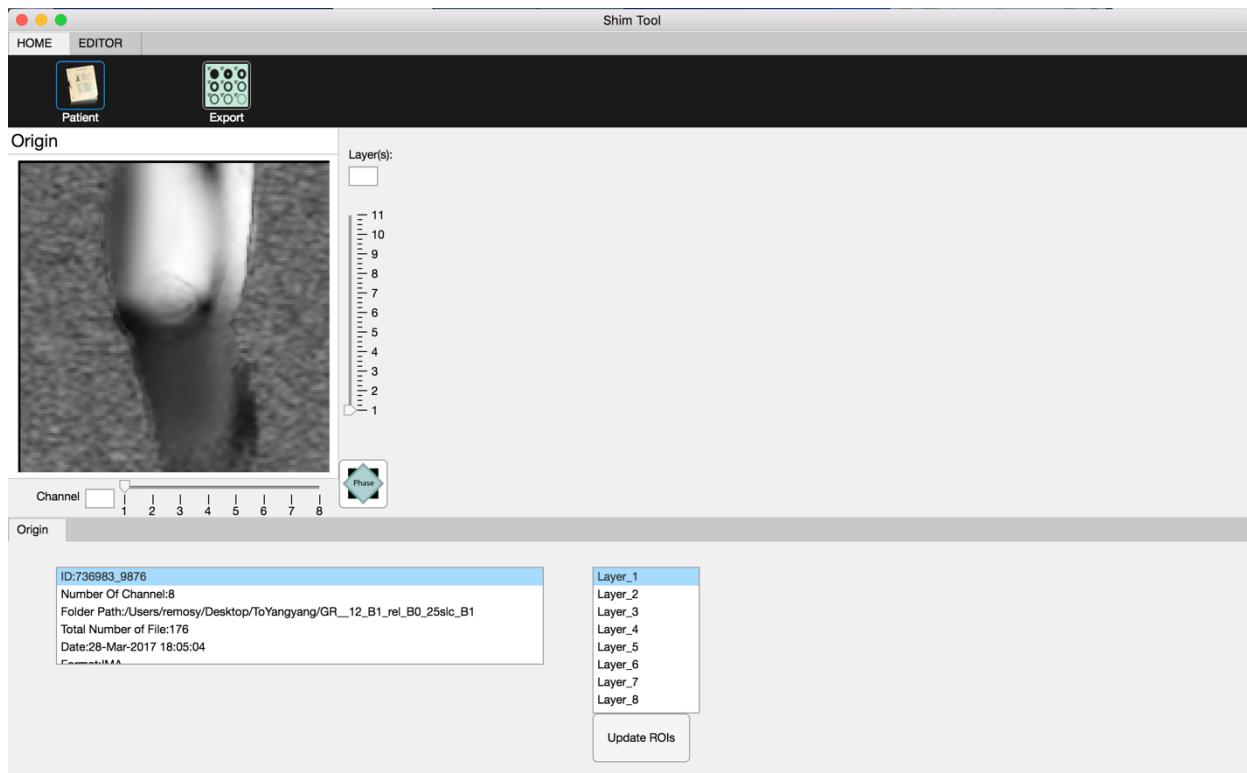


Figure 10.1
Screen shot of ShimTool main GUI

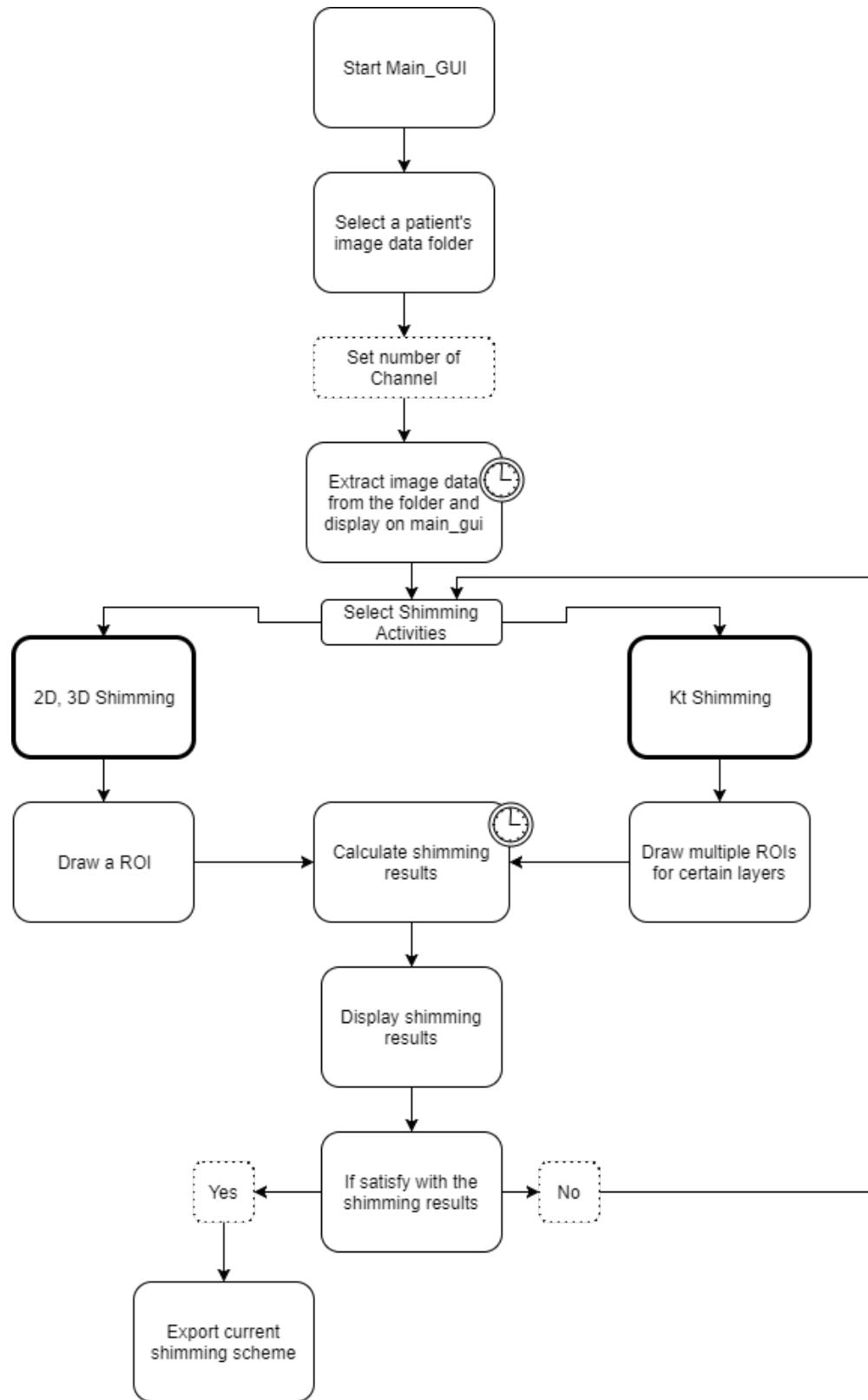


Figure 10.2
Workflow

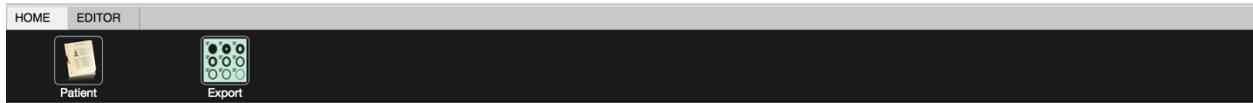


Figure 10.3
Screen shot of ShimTool “HOME” tab

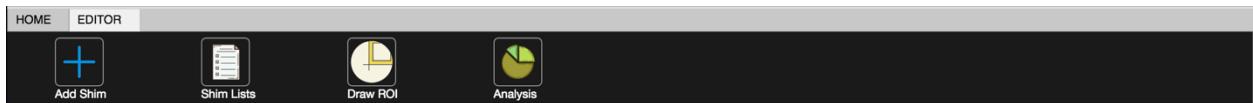


Figure 10.4
Screen shot of ShimTool “EDITOR” tab

2.5.2 Object-Oriented Programming

There are two different classes; one is for GUI, which is the interface shown below, another one is data object. The relationships between the classes are displayed in the Figure 10, the details about two interface classes are illustrated in Appendix. Table 2 describes the contents of the classes files. In the interface classes, the code of GUI features is at the bottom of file. And all of functions in “Process” folder (Fig. 9.1), are implemented in these two GUIs.

2.5.2.1 Interface Classes	
Main_gui	This class represents the interface of the main workplace panel. The ImageData class is a part of this class. It can make a ROI GUI, and the ImageData object used in ROI GUI is from the Main_gui.
ROI_gui	This class represents the interface of a toolbox that can draw the ROI masks. The ImageData and ROIMask objects will be used to update information.
2.5.2.2 Data Classes	
ImageData	This class represents the imported image data. It includes the ROIMask as a sub-class.

ROI Mask	This class represents the ROI masks for an imported medical image, and this medial image is from the <code>ImageData</code> of <code>Main_gui</code> .
---	--

Table 2
Classes

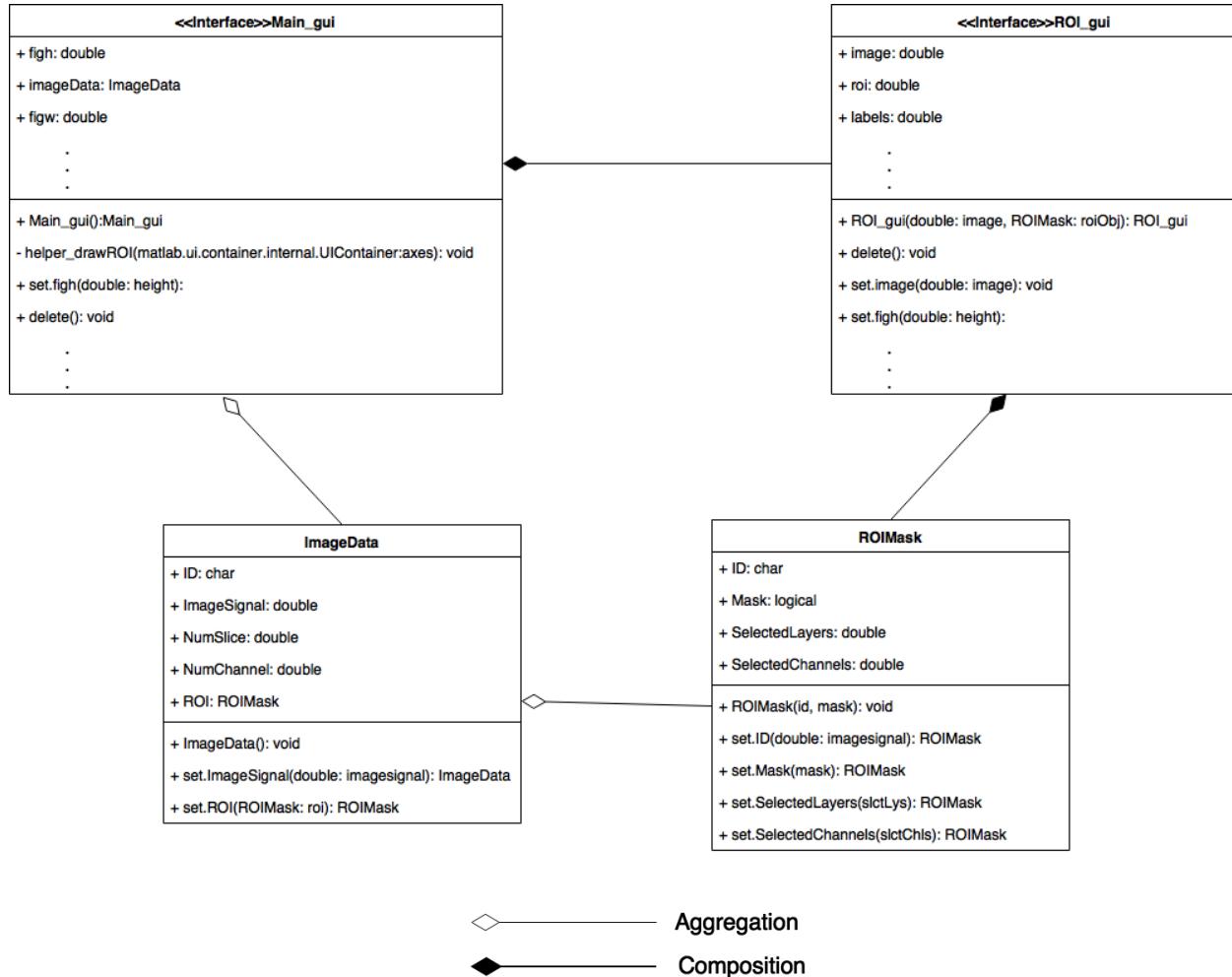


Figure 11
Class Diagram

2.5.3 Data Import and Export

The import (Fig 12.1) is a folder path within raw-image files in IMA, DICM or DCM format. Depending on the number of channels used, the files will include phase and magnitude for each

channel. Because the raw image data folder can be generated by different computers that connect with different MRI machine, thus the file names may need to be edited by users. The previous half image files are magnitudes data, and the last half images files are phase data. Then users need to define the number of a channel, the error message (Fig 12.2) will pop up accordingly. Once an image is imported successfully, the basic information and the slices will be displayed as two scrollable lists on a tab, named “Origin” (Fig 12.3).

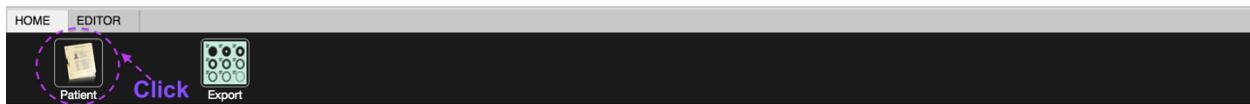


Figure 12.1
Click "Patient"

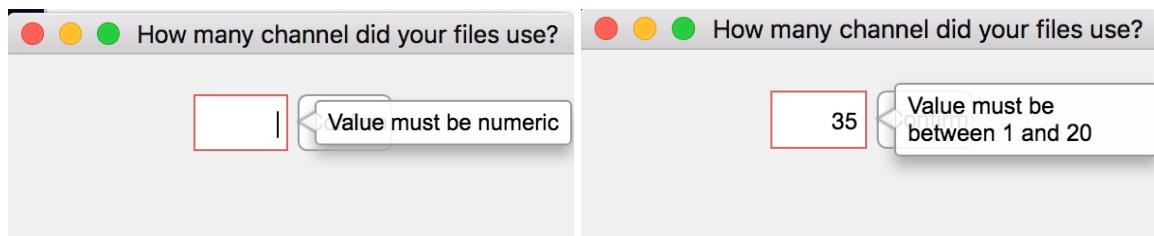


Figure 12.2
Pop up Window

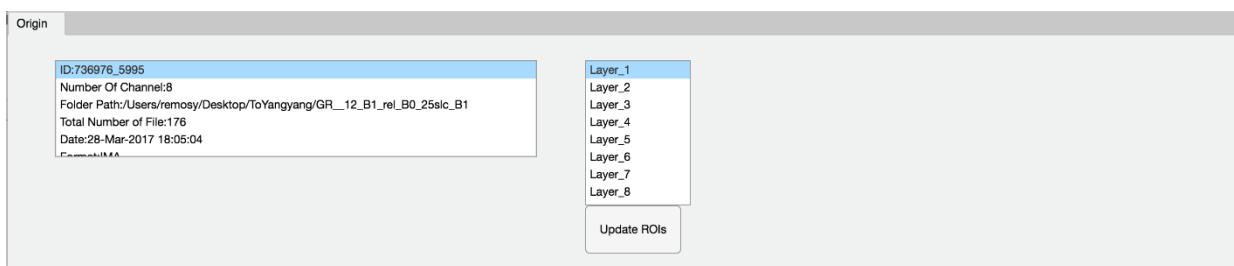


Figure 12.3
“Origin” Tab

The export files are three shimming schemes in “.ini” format, and they are generated in the “OutPut” folder consecutively (Fig 12.5). The files names are generated according to time by the ShimTool application, users can edit for their further purposes. Exported files can be read on

MATLAB. The number of channels, maximum RF, samples, pulse name, comment and the shim schemes are included in one “.ini” file.

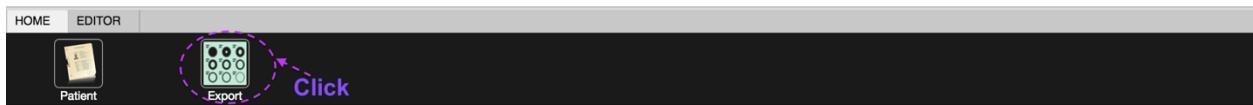


Figure 12.4
Click “Export”

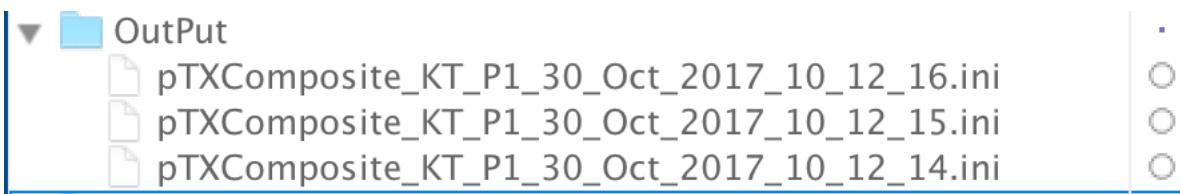


Figure 12.5
Output

2.5.4 MR Image Processing

To get an image as the Figure 13.2 (a) shown, there are two steps, find the phase and magnitude files from the imported folder respectively, since the user provided number of channel, and the layer can be calculated from the imported folder, hence these information is enough to compose such an image.

To describe in detailed. The first step is to divide the files by the phase and magnitude type. As the 2.3.4 mentioned, a folder of files needs to be divided by first half and last half parts. The normal cases are the files can be identified by the number in a file name. Since the number can be at any position in a filename, such like “file_01” and “01_filename”, there is a shared function, “sort_nat”, which made by Douglas Schwarz, on the MathWorks File Exchange platform. It helps sort files names by the number existed in a file name.

For example, there are 176 files, according to the sorted file names. The first 88 files are magnitude, from 89th file to the last file are belong to phase. And user will give the number of channels, 8. Then the number of magnate/phase files divided by the number of channels, the number of layers can be known as well, that is $88 / 8 = 11$ layers.

The second step is to process the image raw data by using the size of layers and channels. In the same case as mentioned above. For the first 88 magnitude data can be extracted from the magnitude files by using MATLAB built-in function, “dicomread”. For the 88-phase data, they also have the same file type as magnitude files, and can be extracted by “dicomread” as well. Because this built-in function is made for the digital image and communications in medical (DICOM) file. After obtaining the data of the 176 files, for magnitude and phase calculation, they may have different formulas which are provided by the MRI machine company. Once the calculated phase and magnitude data are prepared. The final image can be composed from formula 2 of image signal. In this example, there are 11 layers, 8 channels, hence the final image data which includes these 176-file data can be a $128 \times 128 \times 11 \times 8$ matrix, $128 \times 128 \times 11$ for each channel (Fig 13.1).

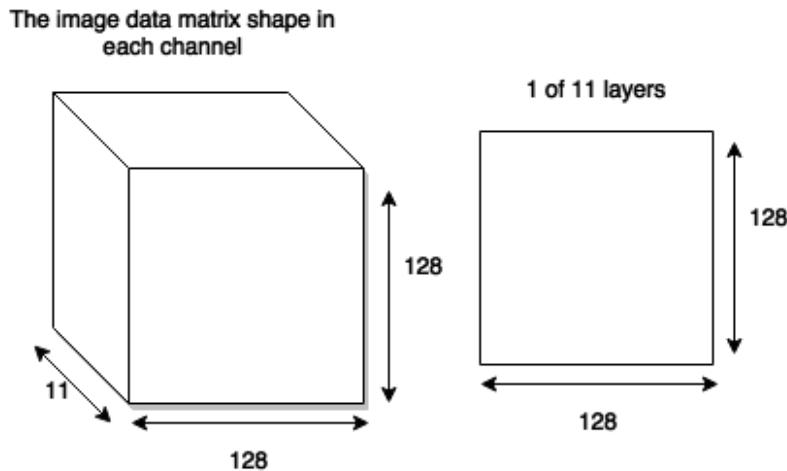
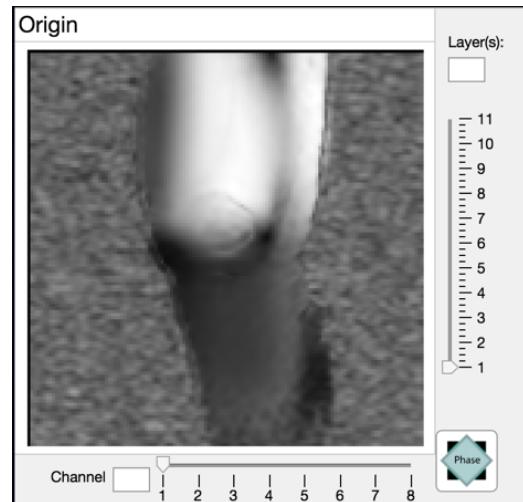


Figure 13.1
Matrix data structure

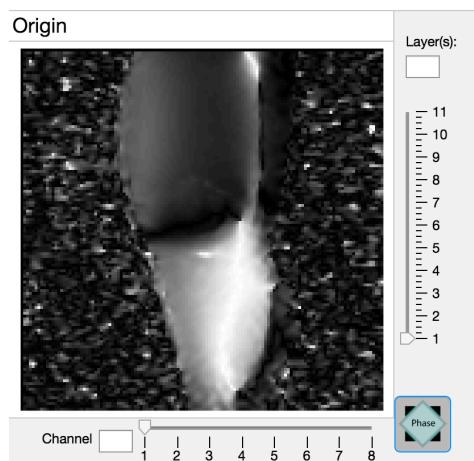
The default image displayed on the panel is layer 1 of channel 1. Like the image (Fig. 13.1) showed beside the cube, its size is $128 \times 128 \times 1$. “uiaxes” is an UI feature provided by MATLAB, it is always used to be a figure container, and the built-in function “imshow” is used to set image matrix on that axes container.

For the main image shown on the dashboard, users can slide the scales to locate the layer and channel. They can also change the numbers in the layer or channel boxes by typing. Once the number has been sensed as changed, corresponding image will be updated on the panel. The magnitude (Fig 13.2 (a)) showed as a default option. Users also can check the phase image (Fig

13.2 (b)) by clicking the green icon “Phase”. The “Phase” image is transferred from the magnitude image by using MATLAB built-in function “angle”.



(a) Magnitude image



(b) Phase image

Figure 13.2

```
function DisplayImage(this,name,img)
    p =
    uipanel(this.imagePanelGroup,'Title',name,'FontSize',18,'BackgroundColor','white','Position',[0 40 340 360]);
    this.mainEvent = [this.mainEvent,name];
    p_axes = uiaxes(p,'Box','off','Position',[-20 -50 390 390],'BackgroundColor','white');
    imshow(img,'Parent',p_axes);
end
```

Table 3
Matlab Code Fragments

“DisplayImage” is a shared function (Table 3), that is used by many call-back function of UI features to display image on this “Origin” panel.

2.5.5 Sliders

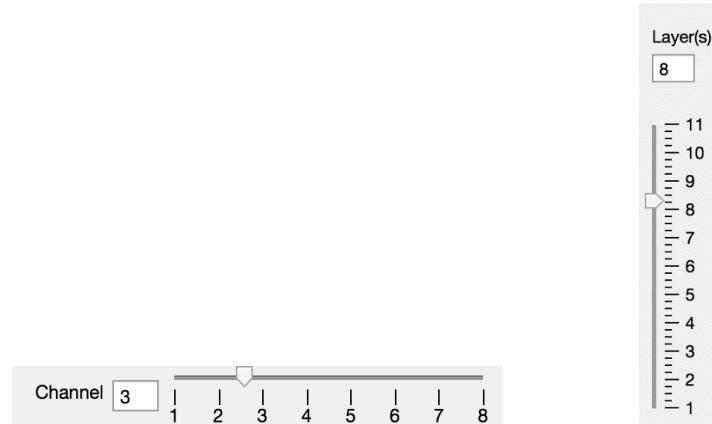


Figure 14.1
Sliders

These two sliders are made for querying the image which belong to channel 3 and layer 8. The ranges of both scales are got from image data. The text boxes changes text along with moving the scale. However, the App Designer does not allow too many users defined properties, such as the numbers picked on Figure 14.1, 2.5 and 8.2, the pointer cannot be changed for only choosing integer, the values have to be rounded to 3 and 8 respectively. When the text box is changed, the slider can sense it and change the position of pointer. If user inputs an out-of-range number, it will notify with a pop up window (Fig 14.2).

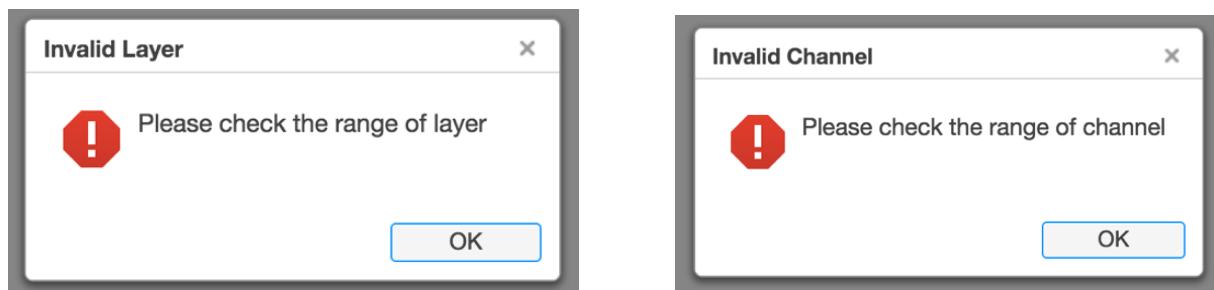


Figure 14.2
Pop up Window

2.5.6 ROI Toolbox

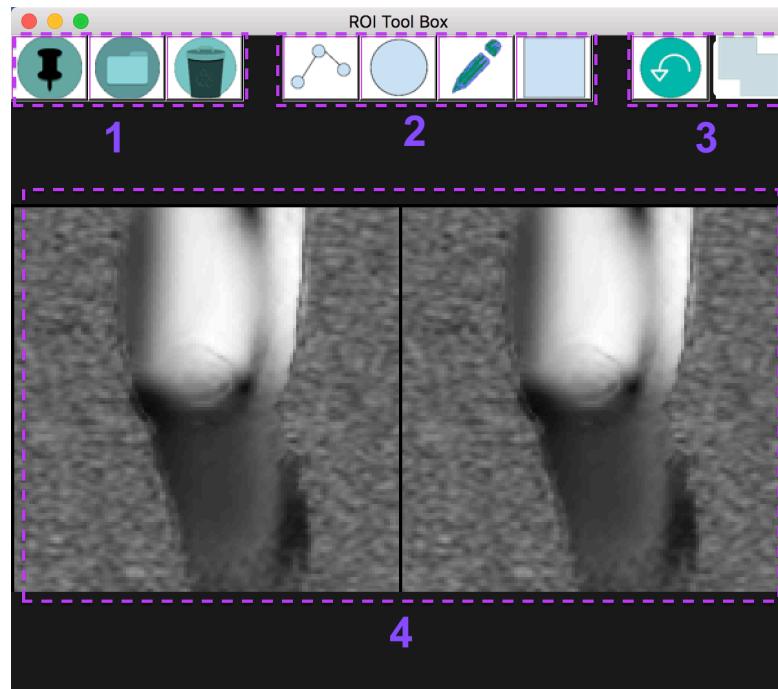


Figure 15.1

There are main 4 parts (purple frame) compose a ROI toolbox. In part 1 area, these three features can interact with Main GUI. There are 4 shapes of image plot in part 2. For the part 3, these two features can only be used in ROI toolbox. The part 4 is used to display images. Following part will introduce these 4 parts.

2.5.6.1 Part 1

Icon	Description
	This is used to save a ROI on the MATLAB workspace and draw this ROI on the “Origin” panel. The file names normally start with “roi_”, and a timestamp is followed.

	This is used to load all of “.m” files with “roi_” prefix from MATLAB workspace into a window (Fig. 15.2). User can only select single file and click “Confirm” to update ROI mask on the “Origin” panel.
	This is used to delete the ROI on the “Origin” panel.

Table 4
Icons

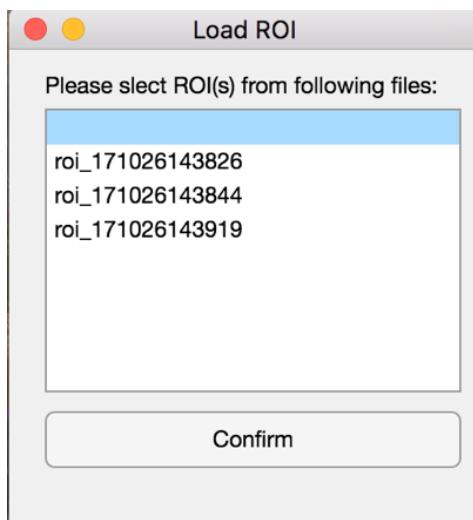


Figure 15.2
Load ROI screenshot

2.5.6.2 Part 2

These 4 shapes have their own built-in function of MATLAB. Except “imfreehand”, the other three shapes allow users to enlarge and diminish shape by dragging the node on the frame. And all shapes allow users drag to move shapes. These shapes also have their own properties that can be used by left click, the pop-up menu is look like the Figure 16.1. Once a shape is created, the shape will be saved as an object in a MATLAB cell array with a tag, that indicates if this shape is made for addition or subtraction. The tag form (“+imsel_1”) is illustrated in the Figure 16.2. As an object, the properties, such as the mask can be gotten from the MATLAB built-in function, “createMask”.

ROI Shape	Polygram	Ellipse	Free Hand	Rectangle
-----------	----------	---------	-----------	-----------

Icon				
Built-in Function	impoly	imellipse	imfreehand	imrect
Example				

Table 5

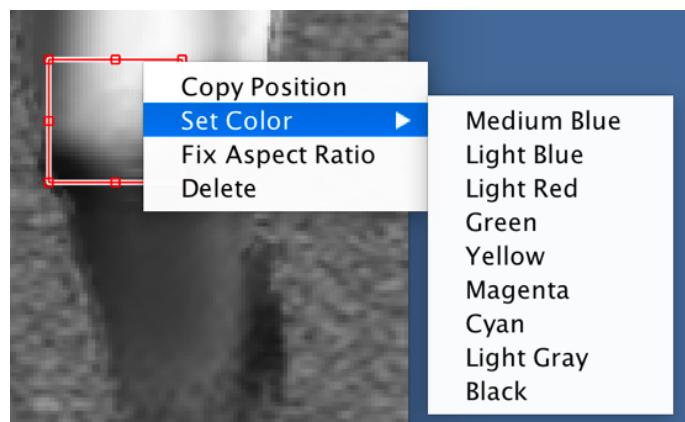


Figure 16.1

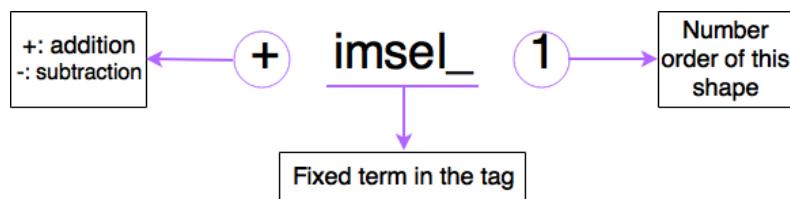


Figure 16.2

2.5.6.3 Part 3

Icon	Description
	This is used to delete shape that lasted created. And the shape can be deleted on both screens which show on the part 4.

	This is a toggle button for shape addition. The variable (isSubtract) in the “ROI_gui” class is 0.
	This is a toggle button for shape subtraction. The variable (isSubtract) in the “ROI_gui” class is 1.

Table 6

2.5.6.4 Part 4

The left figure is “this.roiax_axes” in the class, it is used to display the result of several ROI that drawn by users. The right figure is “this.imax_axes” in the class, it is used as a sketchpad that helps users draw a ROI. “linkaxes” is a built-in function that can let users plot the 2-dimentional ROI shapes on the both screens synchronously, but only the “GUIDE” GUI has this built-in function.

There are two types of ROI; one is for adding up the shapes, another one is for subtracting a shape. The shapes icons displayed on the top of Figure 17.1, are provided by MATLAB built-in functions. Table 2 introduces all of the icons. The Figure 17.2 (a) to (d) are the results can be gotten by using these icons.

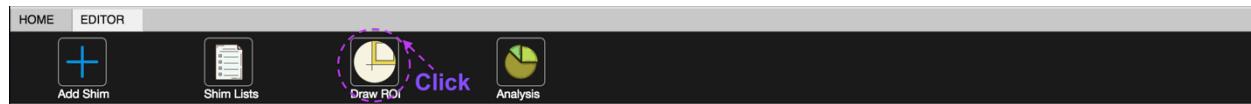
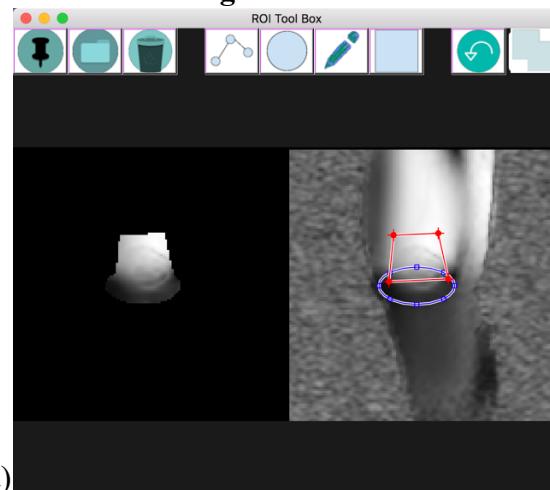
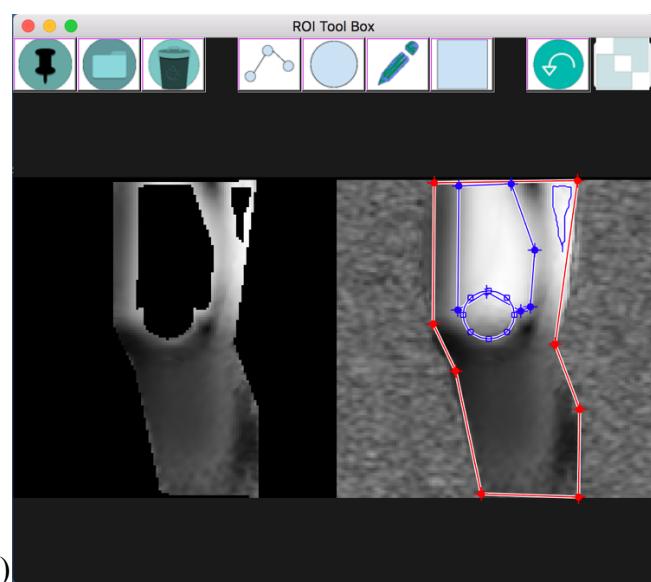
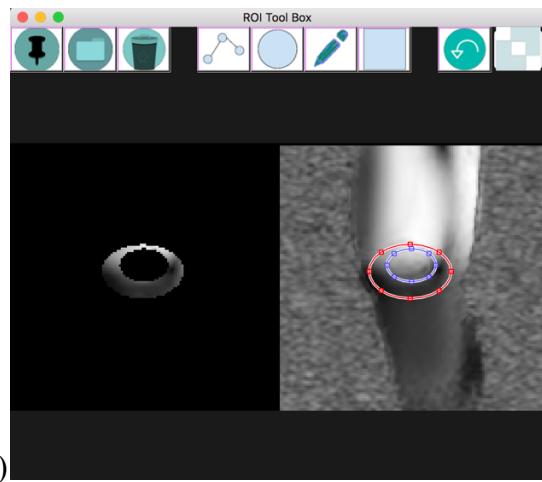


Figure 17.1





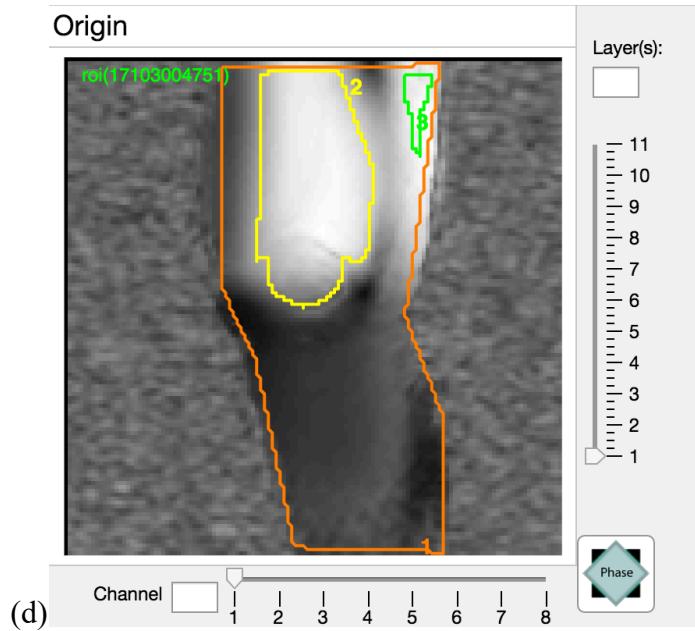


Figure 17.2

The mask is created according to the location and size of the shapes, it will a 2D logical type, which means the region of interest are 1, the color will be white, and remaining area is represented by 0, the color will be black. For example, there are two 12×12 logical matrixes, matrix A (Fig 17.3) and matrix B (Fig 17.4). When the two shapes are needed to be combined as one ROI mask, use $A | B$. When Figure B is made for subtracting Figure A, cannot use $A & B$ directly, because the area of original A is neglected, then the problem can be avoided by using $A & \sim B$. These solutions (Fig 17.5) can be used multiple times until getting the final ROI as users like.

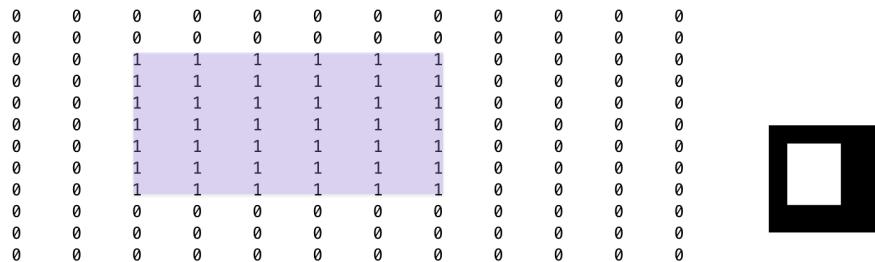


Figure 17.3
Matrix A

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	0	0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 17.4
Matrix B

A & B

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

A & ~B

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

A | B

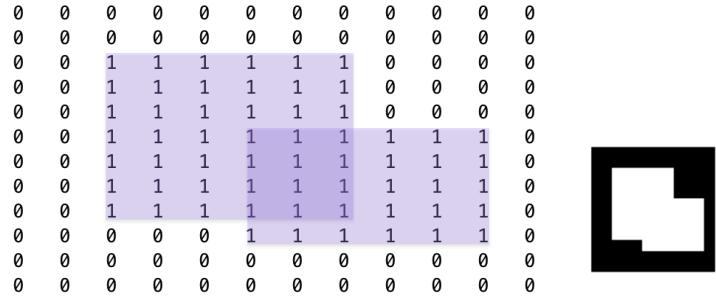
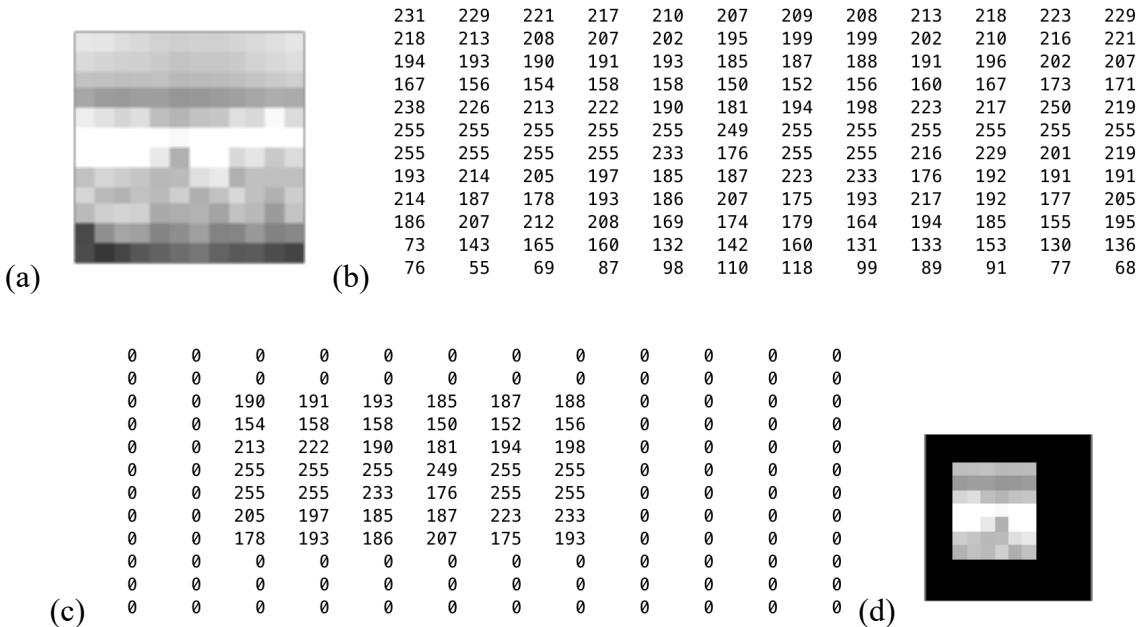


Figure 17.5

The images showed on the left panel of Figure 17.2 (c) and (d), are made by two steps. For example, there is a 12 x 12 pixel image as shown (Fig 17.6(a)(b)). When there are two shapes ROI, mask A (Fig 17.3) and B (Fig 17.4), the next step is to combine the Shape A and B ROI masks with the 12 x 12 pixel image. The matrix of an image is not logical, each integer number represents grey degree. To imply mask A (Fig 17.6 (c)(d)) to the image, the matrix of shape A and matrix of image will be multiplied. In the actual program code, the type of matrix A and B will be converted to be integer. When adding the mask B to the image, the matrix of the A&~B and matrix of image will be multiplied (Fig 17.6 (e)(f)).



0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	190	191	193	185	187	188	0	0	0	0	0
0	0	154	158	158	150	152	156	0	0	0	0	0
0	0	213	222	190	181	194	198	0	0	0	0	0
0	0	255	255	255	0	0	0	0	0	0	0	0
0	0	255	255	233	0	0	0	0	0	0	0	0
0	0	205	197	185	0	0	0	0	0	0	0	0
0	0	178	193	186	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
(e)	0	0	0	0	0	0	0	0	0	0	0	0



Figure 17.6

2.5.7 Shimming Simulation

“Add Shim” button is designed for doing shimming simulation for all layers. At this stage, only B_1 shimming schemes are supported. Before processing the shimming, at least one ROI must be applied. Users can select which layers that are used to do the B_1 shimming, via using the “Update ROIs” button without clicking the “Add Shim”.

2.5.7.1 Layer selection for shimming results

Because the binary sequence is easy to understand and store the layers/channels selected. In the code, the layers/Channels will be treated as a sequence binary number. When a layer/channel is selected, the number will be 1, otherwise 0. But they will be converted to the decimal number and saved in the “ImageData” object, this may help with further developing and debug, it is easier to identify differences. For example, there are 8 layers, but only layer 2 and layer 4 is selected (Fig 18.1). The binary sequence is 01010000, its decimal form is 80, and both variables are 8 bytes. This serves to apply shimming schemes on selected layers. After obtaining an ROI, the frame of ROI will be shown with the ROI id sequence. After selecting layers and click “Update ROIs” to confirm, the ROI will only show up on layer 2 and layer 4, otherwise, the “NaN” will replace the efficiency (Fig 18.2).

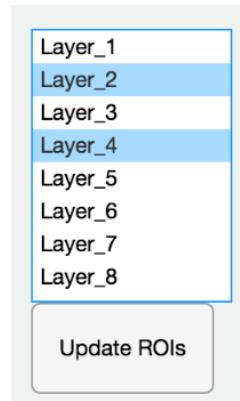
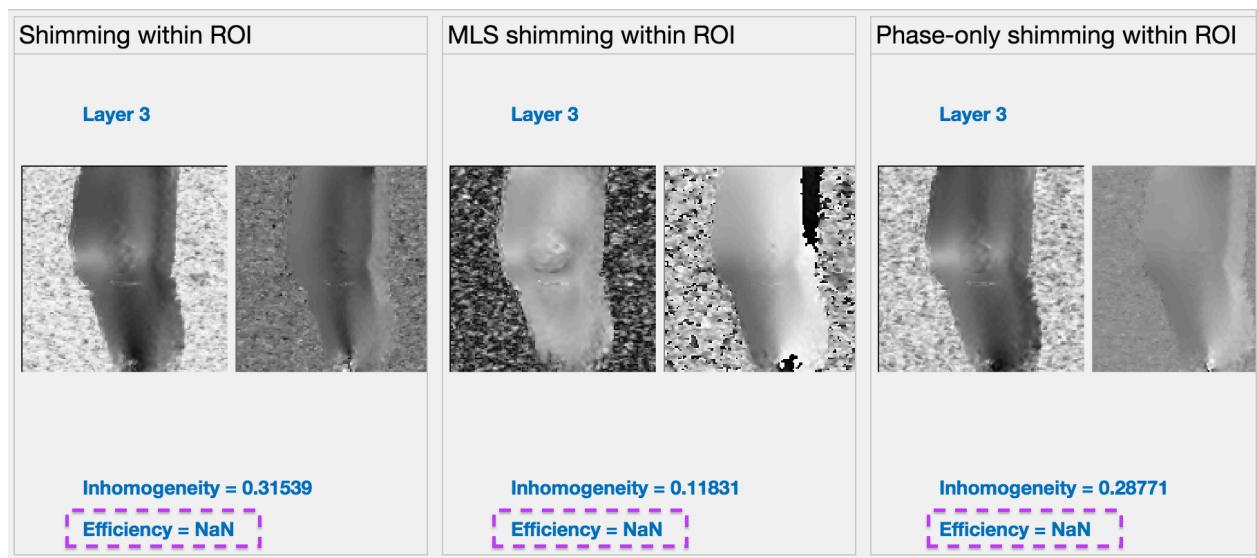
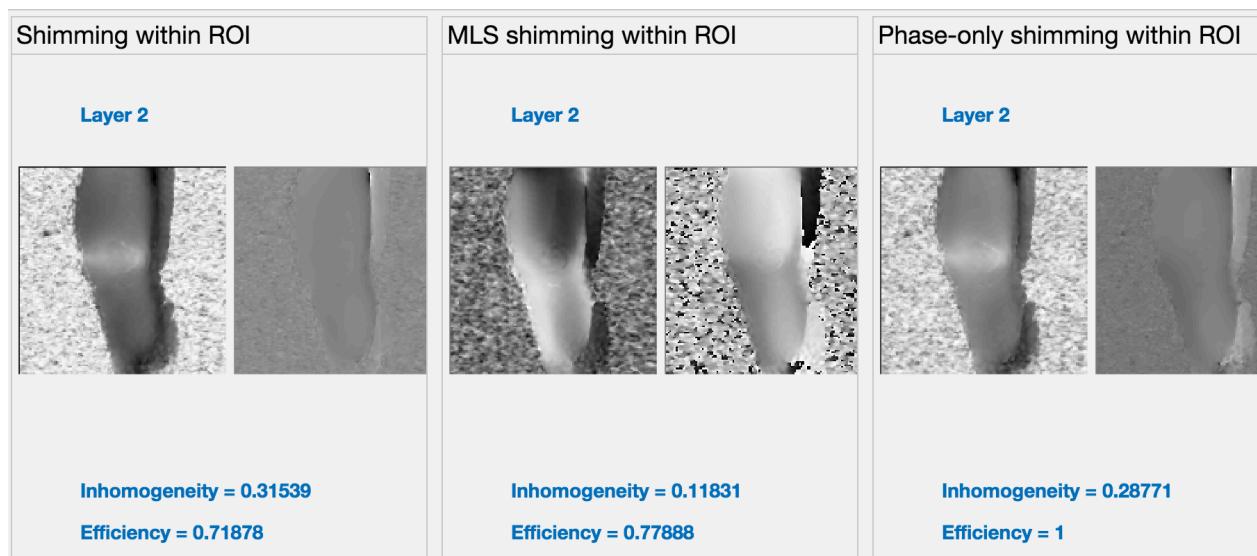


Figure 18.1



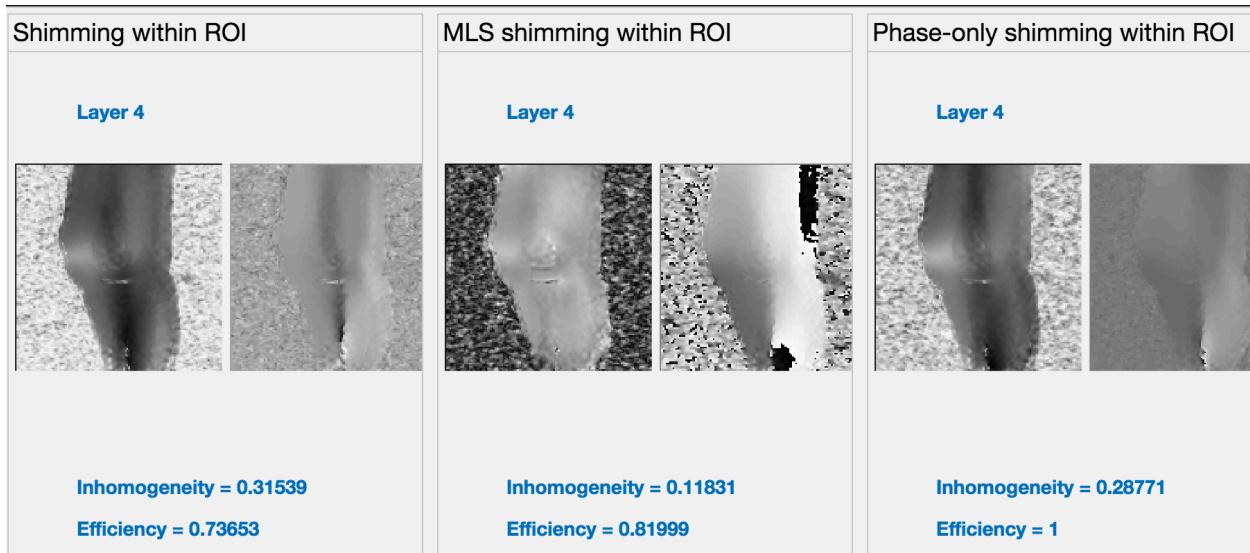


Figure 18.2

Finally, The B_1 shimming image and shimming schemes will be generated and displayed. In the application folder, the “Shim.m”, “GaussFilter.m” and “myMLS.m” are mainly used.

To get the final B_1 shimming image. There are 3 steps: filter image with ROI by Gaussian filter, apply the shimming formula and display the results.

2.5.7.2 Gaussian Filter

Gaussian will be applied on processing the image area within ROI (Fig 19.1 (a)). The Gaussian filter as an image correction method, it helps normalize pixels, and reduce the noise to achieve smoother image. Figure 19.1 (c) shows a blurred image after filtering with Gaussian, the top-left corner of Figure 19.1 (b) has been improved on the Figure 19.1 (c). In the MATLAB, the built-in function “fspecial” can be used to make a 2D Gaussian filter, the radius is set to be a default vector [5,5], the standard deviation sigma is set to be default (0.5). Another function, “imfilter” can be used to apply the Gaussian filter on an image. The pixels of a Gaussian image are much less than original pixel, the edges of ROI is also changed, original image pixel starts from row 46 and with only 4 non-zero image pixels.

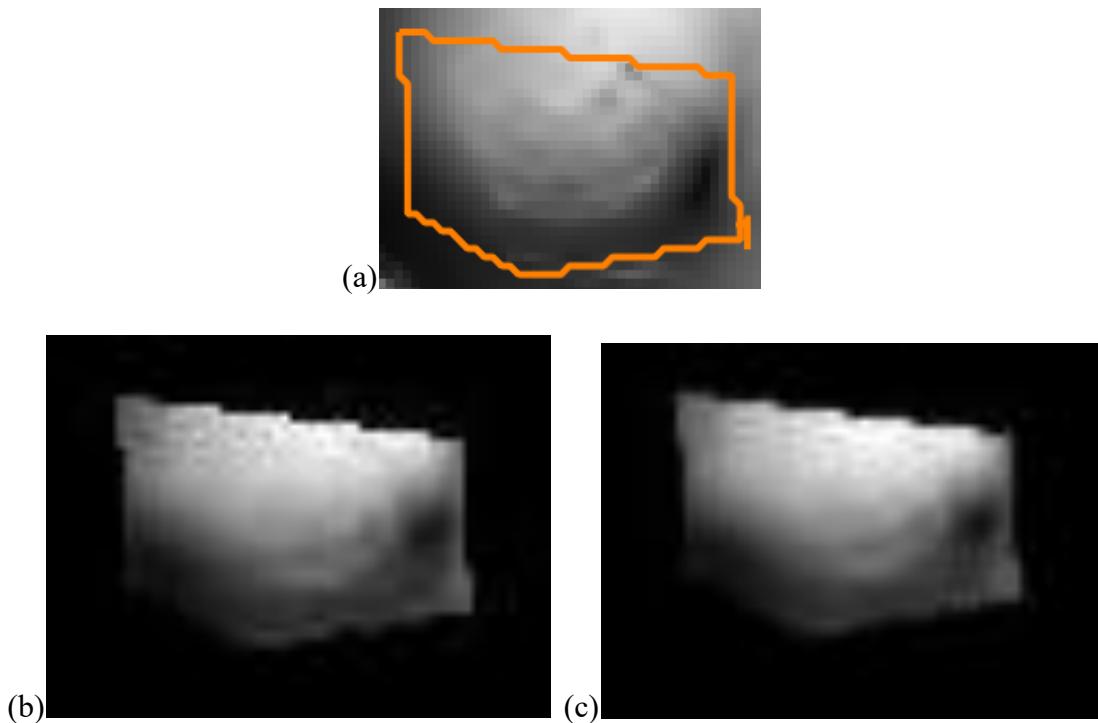


Figure 19.1

46	0	0	0	0.33148	0.3575	0.38994	0.43233	0	0	0	0
47	0	0	0	0.33645	0.36422	0.39638	0.43642	0.48495	0.53785	0.58842	0.63315
48	0	0	0	0.33557	0.36451	0.39696	0.43642	0.48465	0.53902	0.59222	0.63783
49	0	0	0	0.33704	0.36568	0.39608	0.4335	0.48319	0.5437	0.6007	0.64513
50	0	0	0	0.34493	0.37007	0.39638	0.43058	0.48027	0.54165	0.5969	0.64163
51	0	0	0	0.35545	0.37504	0.39784	0.43087	0.47881	0.53639	0.59164	0.63899
52	0	0	0	0	0.38176	0.40427	0.43788	0.48348	0.53727	0.59222	0.63987
53	0	0	0	0	0.38936	0.41567	0.45016	0.49284	0.54282	0.59369	0.63578

44	0	0	0	2.4109e-08	9.7523e-06	8.2386e-05	9.8708e-05	0.00010774	0.0001052	1.2714e-05	3.1444e-08	0
45	0	0	0	9.7507e-06	0.0039442	0.03332	0.039922	0.043575	0.042561	0.0052626	0.00016076	0.00016197
46	0	0	0	8.1764e-05	0.033074	0.27942	0.33489	0.36545	0.36233	0.091963	0.060007	0.065507
47	0	0	0	9.2545e-05	0.037434	0.31638	0.38007	0.41416	0.45047	0.45796	0.50165	0.54872
48	0	0	0	9.2563e-05	0.037443	0.31657	0.38115	0.41526	0.45647	0.50657	0.56292	0.61792
49	0	0	0	9.3088e-05	0.037655	0.3183	0.38245	0.4146	0.45406	0.50554	0.56662	0.62486
50	0	0	0	9.5126e-05	0.038479	0.32482	0.38666	0.41499	0.45167	0.50297	0.56473	0.62207
51	0	0	0	8.721e-05	0.035286	0.30178	0.38785	0.41716	0.45236	0.50183	0.56064	0.61763
52	0	0	0	1.0455e-05	0.0043229	0.07372	0.36306	0.42389	0.45925	0.50641	0.56173	0.61768
53	0	0	0	2.5852e-08	0.00011716	0.043253	0.36433	0.43329	0.46962	0.51439	0.5661	0.61789

Figure 19.2

However, the input image data is 4D, which has the case of $128 \times 128 \times 11 \times 8$ (128×128 image, 11 layers, and 8 channels). To ensure a 2D image will be displayed with Gaussian, according to the matrix information, the channel and layer can be a coordinator of it. There are 8 channels, and each channel has 11 layers, thus there are $11 \times 8 = 88$ layers in total. To get a new image data as the Figure 19.3 shown, the MATLAB build-in function “reshape” is used. The image data is used in shimming is $128 \times 128 \times 11 \times 8$. Therefore, the data will be reshaped back from the size $128 \times 128 \times 88$.

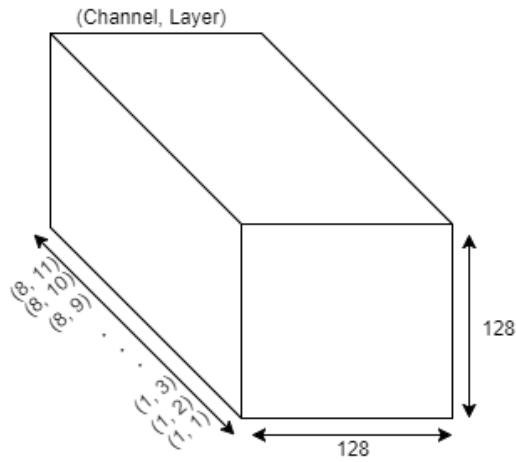


Figure 19.3

2.5.7.3 Implementation of shimming schemes

There are two outputs will be yield. The one is the channels shimming schemes; another one is the B_1 shimming mapping that uses the shimming schemes. The raw data of this step is the image data after Gaussian filter.

The result of schemes should be a matrix that is related to each channel used, and the ROI mask will be used to constrain the layer image data in each channel. To make a new image data that is about the channels, the MATLAB built-in function “squeeze” will make $128 \times 128 \times 11 \times 8$ image data be like the case shows on the Figure 19.3. For each channel, there are 11 layers, and the ROI mask is applied on each layer. Then only extract the data within in the ROI from each layer in one channel, the part is for eliminating the black area that is outside of the ROI (Fig 20). The extracted image data of one channel is the result of total single layer data ($128 \times 128 \times 11 = 180224$) subtract the zero (black) area (174278), which will be an array with size 10362 ($180224 - 174278 = 10362$).

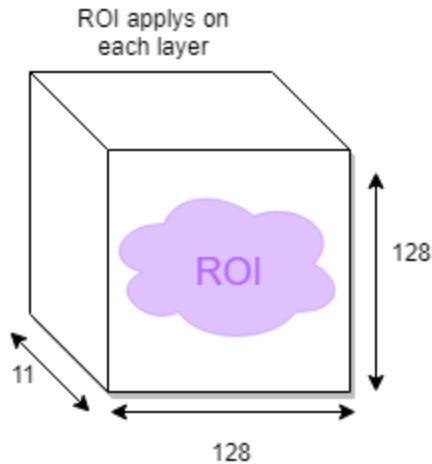


Figure 20

Section 2.2.1 has introduced three B_1 pulsing strategies. Its simulation will be implemented in this section. In the following calculation, the shared B_1 target is $1e - 6$. If there is an ROI mask, set it as a variable “mask_shm” in code, and the extracted image data matrix for each channel is B.

For the B_1 magnitude-phase shimming, according to the formula, the extracted image data need to be inverted. However, its size is not an invertible matrix, and needs apply the Moore-Penrose inverted, “pinv”, a built-in MATLAB function that can inverse a matrix with the size like 10362 x 8 (8 channels, each channel has data size 10362 x 1).

MATLAB code: `pinv(B)*(B1target.*ones(numel(find(mask_shm)),1))`

When B_1 magnitude-phase shimming used the MLS method, in this application, the MLS function is provided by my supervisor. There are 4 inputs of this function, A, b, tol, and MaxIter, the outputs are x, dif, iter [Table 3], where the tol and MaxIter are 0.01 and 1000.

MATLAB code: `[Shim_MagPha MLS, difMLS, TotalIter] = myMLS(B, B1target.*ones(numel(find(mask_shm)),1).*exp(1i.*pi/2), 0.01, 1000);`

Input		Output	
A	An extracted image data matrix.	x	The shimming scheme optimized from the MLS method.
b	The target amplitude, and the initial phase of target.	dif	The difference yield from the MLS method.

tol	The optimized target tolerance.	iter	The total iteration.
MaxIter	The maximum number of iteration.		

Table 7

For the phase-only shimming, according to the formula. An averaged phase will be applied on the extracted image data. There is no amplitude.

MATLAB code: `exp(-1i.*angle(mean(B, 1)));`

After getting the shimming results, they are needed to be normalized. The sudo code is:

`Shim_results ./ max (Shim_results)`

Where the “Shim_results” means the results of one of three shimming schemes.

After obtaining the results of each shimming schemes, the image mapping of magnitude-phase and MLS are similar. The image is made by the product of a reshaped matrix of image data and the results of shimming. For the phase-only, this image is made by the product of a reshaped matrix of image data and the shimming results that are transposed.

2.5.7.4 Visualization of shimming schemes

In the Main_gui.m file, there are two methods are mainly used for B_1 shimming schemes. B_1 shimming image results, top of Figure 21 is made by “DisplayShim”, and bottom of Figure 14 is made by “PlotShimBar” in the application code. For the shimming image, the default shimming results are from the first layer that user selected. The shimming images have two results. The left side is magnitude image; right side is phase image. The slider is for the user, to check shimming results of other layers, users can also operate it by sliding the layer scale beside the image. When the layer is not selected before applying the shim, there are no efficiency values. The inhomogeneity and efficiency for each B_1^+ shimming scheme shown below are calculated by the formula 9 and 10. In the MATLAB, it provides “abs” to get an absolute value of a result, and “mean” to get an average value of a result. Thus, in the formula, the $\|B_1^+\|^2$ in efficiency can be replaced by “abs” operation, and the “average” can be replaced by the “mean” operation.

On the bar charts, the dark-blue bars show values of amplitude, the light-blue bars show values of phase. The y-axis uses two different ranges, they are 0 – 1 and 0°- 360° for amplitude and phase

respectively. The x-axis uses range 1 – 8, shows the number of a channel. For example, as in the purple area shown, it means by using the B_1 Shimming within ROI, the number 1 channel has amplitude around 0.4, the phase is around 180° .

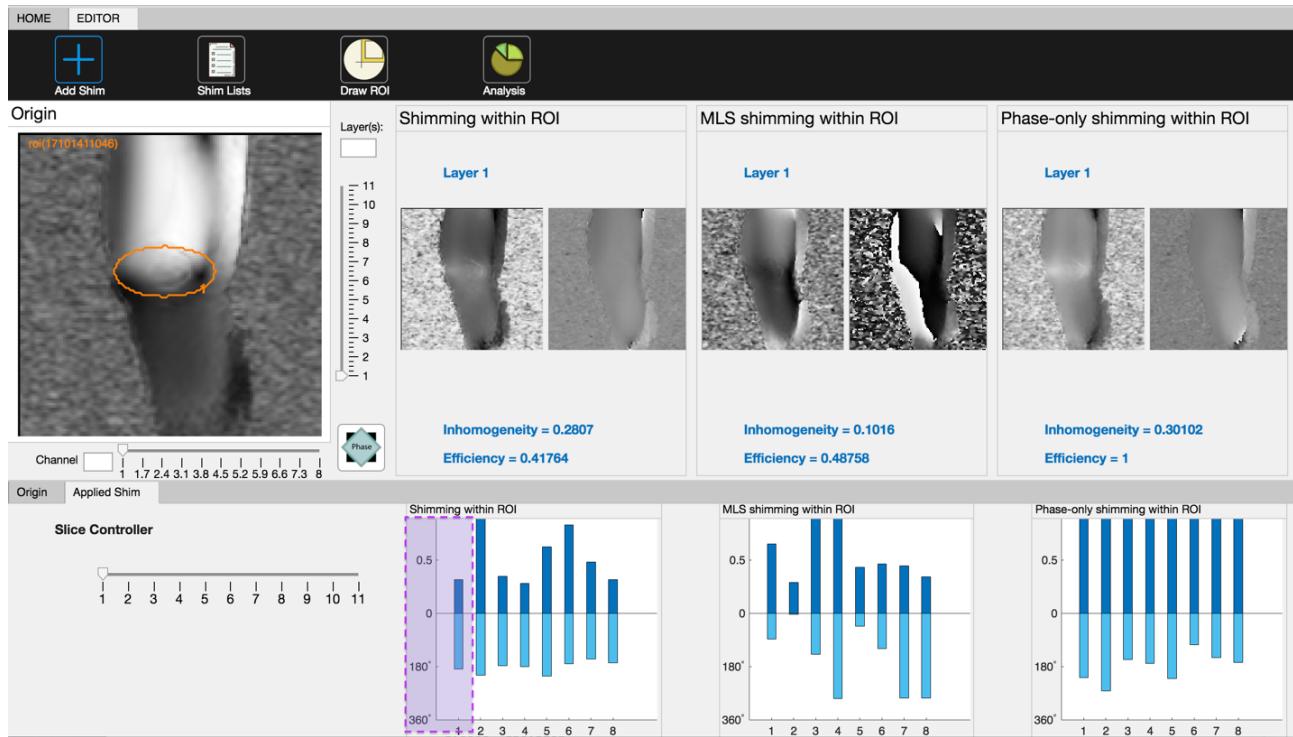


Figure 21

3. Results

In this section, knee, hip and spine are three human body parts which will be illustrated by experiments. The results are gotten by using the experimental methods, step 3 and step 4 from section 2.3. Due to these three body parts have similar operations on localiser and B_1 mapping as Knee, the localizer and mapping are only illustrated in the Section 2.3. All phase-only shimming efficiencies and phase are 1. Only Knee shows the three screenshots of exported files, the other two body part will show the file schemes in tables.

3.1 Simulation Results

3.1.1 Knee



Figure 22.1
Folder Information

Slice No.	6	7	8	9	10
Shimming Within ROIthin	0.53643	0.49817	0.4817	0.48098	0.48354
MLS Shimming Within ROI	0.56524	0.55073	0.53862	0.52238	0.50658
Phase-only Shimming Within ROIROI	1	1	1	1	1

Table 8

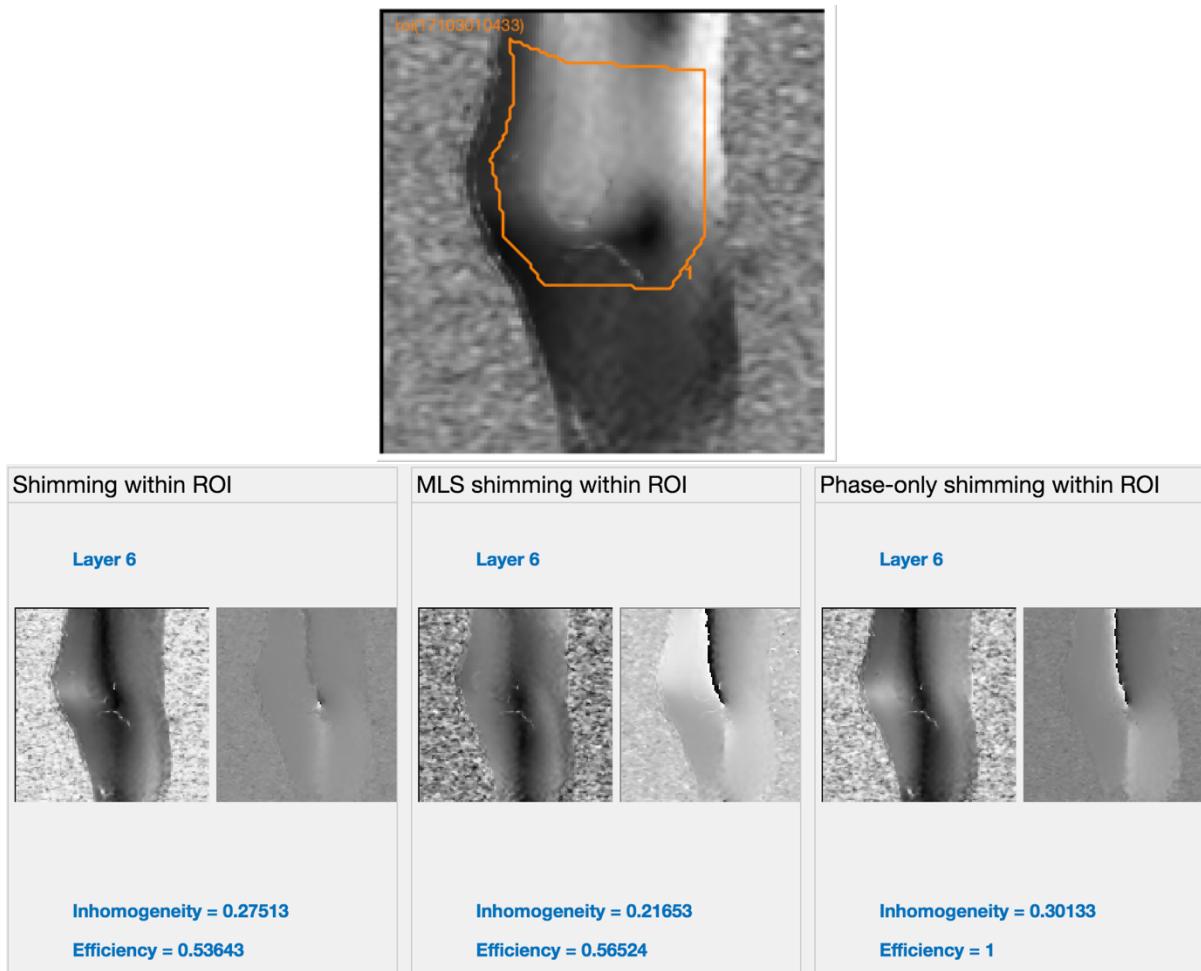
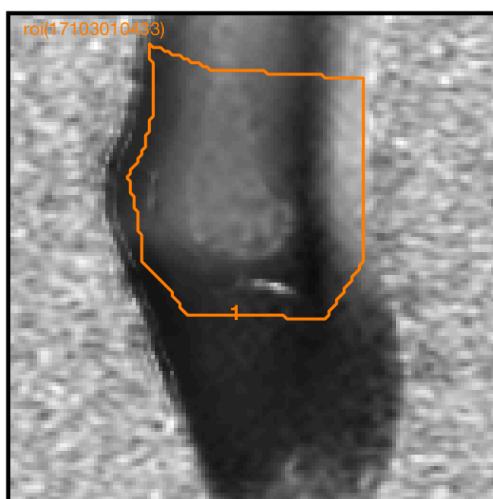


Figure 22.2
Slice 6



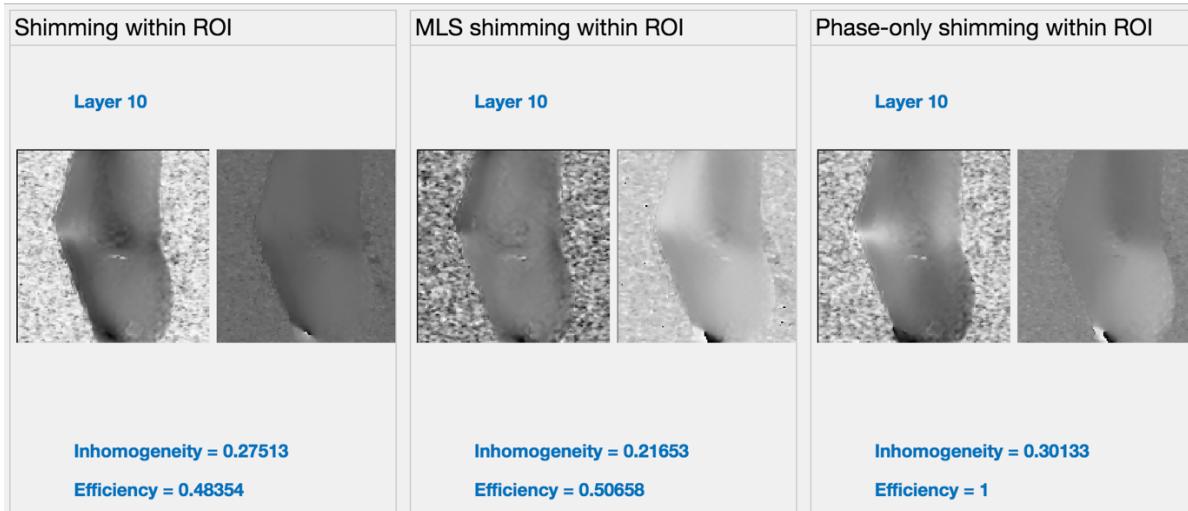


Figure 22.3
Slice 10

	Shimming Within ROI	MLS Shimming Within ROI	Phase-only Shimming Within ROI
Inhomogeneity	0.27513	0.21653	0.30133

Table 9
Inhomogeneity of 3 Shims

Above images are the knee shimming results with the orange ROI. The knee folder information is displayed in Figure 22.1. Table 8 illustrates that from slice 6 to 10, using MLS improved the shimming efficiency. Table 9 also shows the one used MLS has higher homogeneity. However, such ROI cannot be employed to every slice. Figure 22.3 shows the ROI is involved noise on slice 10. Thus, to improve it, smaller ROI is used. The results show slice 1 that has highest shim efficiency, and lower inhomogeneity than last ROI, however, Figure 22.3.1 still illustrated it is not homogeneity around the edge of knee. For the “Shimming within ROI” and “Phase-only shimming within ROI”, the inhomogeneity values indicate the new ROI showed in Figure 23.3.1 does improve the inhomogeneity slightly.

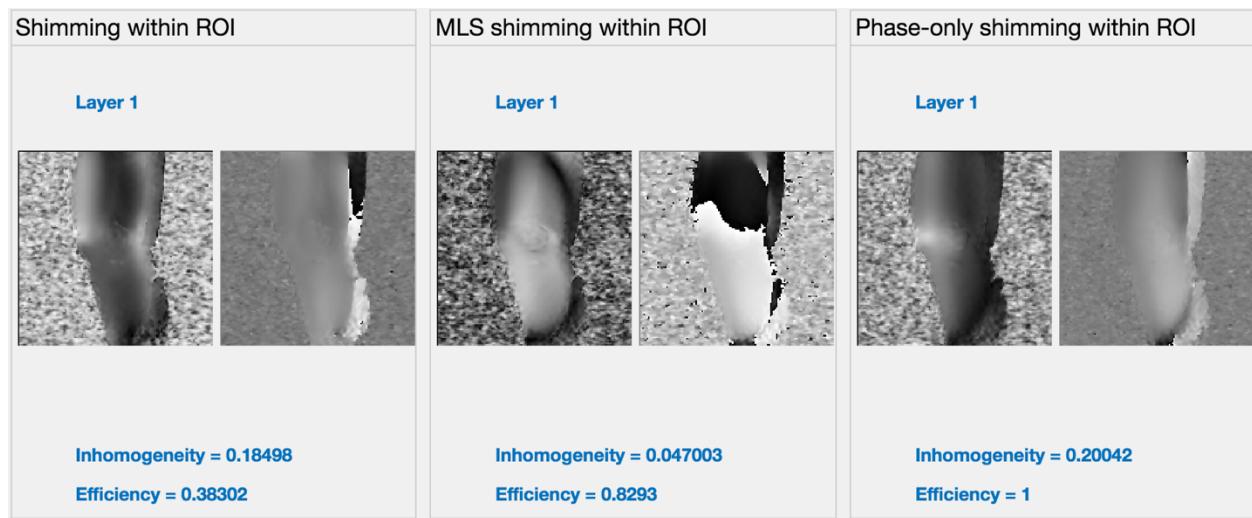
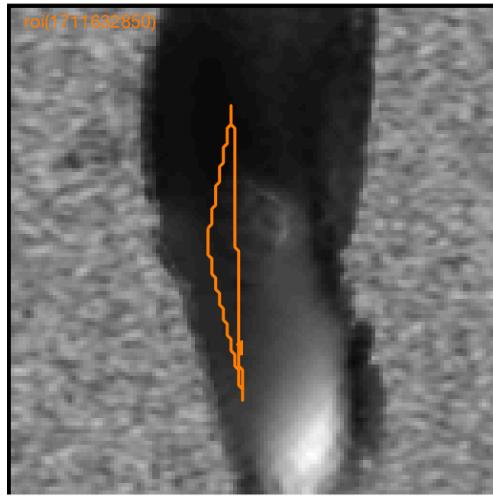


Figure 22.3.1
Slice 1

Following Figures shows the final schemes obtained from this MATLAB ToolBox. The final three “ini.” format documents screenshots can be used in the MRI software.

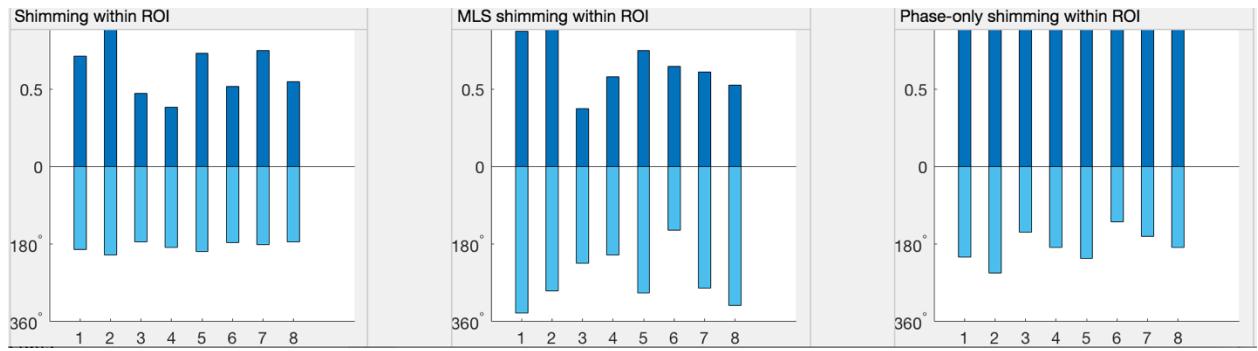


Figure 22.4
Amplitudes and phases of 8 channels

```

1  #pTXRFPulse
2  [pTXPulse]
3
4  NUsedChannels      = 8
5  MaxAbsRF            = 1
6  Samples              = 1
7  PulseName            = pTX composite pulse KT 1 30-Oct-2017_10:12:14
8  Comment               = OLS spokes
9
10 [pTXPulse_ch0]
11 RF[0]=  0.7182   3.364
12
13 [pTXPulse_ch0]
14 RF[0]=  1     3.567
15
16 [pTXPulse_ch0]
17 RF[0]=  0.4771   3.057
18
19 [pTXPulse_ch0]
20 RF[0]=  0.3851   3.271
21
22 [pTXPulse_ch0]
23 RF[0]=  0.7334   3.431
24
25 [pTXPulse_ch0]
26 RF[0]=  0.5219   3.065
27
28 [pTXPulse_ch0]
29 RF[0]=  0.7524   3.147
30
31 [pTXPulse_ch0]
32 RF[0]=  0.5527   3.042
33
34 #EOF

```

```
1 #pTXRFPulse
2 [pTXPulse]
3
4 NUsedChannels      = 8
5 MaxAbsRF           = 1
6 Samples             = 1
7 PulseName           = pTX composite pulse KT 1 30-Oct-2017_10:12:15
8 Comment              = OLS spokes
9
10 [pTXPulse_ch0]
11 RF[0]= 0.8759 5.922
12
13 [pTXPulse_ch0]
14 RF[0]= 1 5.036
15
16 [pTXPulse_ch0]
17 RF[0]= 0.3777 3.902
18
19 [pTXPulse_ch0]
20 RF[0]= 0.5817 3.575
21
22 [pTXPulse_ch0]
23 RF[0]= 0.7527 5.122
24
25 [pTXPulse_ch0]
26 RF[0]= 0.6481 2.578
27
28 [pTXPulse_ch0]
29 RF[0]= 0.6142 4.919
30
31 [pTXPulse_ch0]
32 RF[0]= 0.5281 5.626
33
34 #EOF
```

```
1 #pTXRFPulse
2 [pTXPulse]
3
4 NUsedChannels      = 8
5 MaxAbsRF           = 1
6 Samples             = 1
7 PulseName           = pTX composite pulse KT 1 30-Oct-2017_10:12:16
8 Comment              = OLS spokes
9
10 [pTXPulse_ch0]
11 RF[0]= 1 3.652
12
13 [pTXPulse_ch0]
14 RF[0]= 1 4.304
15
16 [pTXPulse_ch0]
17 RF[0]= 1 2.665
18
19 [pTXPulse_ch0]
20 RF[0]= 1 3.267
21
22 [pTXPulse_ch0]
23 RF[0]= 1 3.725
24
25 [pTXPulse_ch0]
26 RF[0]= 1 2.235
27
28 [pTXPulse_ch0]
29 RF[0]= 1 2.824
30
31 [pTXPulse_ch0]
32 RF[0]= 1 3.261
33
34 #EOF
```

Figure 22.5
Three shimming-scheme files which are exported

3.1.2 Hip

ID:736994_8103
Number Of Channel:8
Folder Path:/Users/remosy/Dropbox/toYangyang/UnilateralHip/GR__15_B1_rel_B0_11...
Total Number of File:176
Date:07-Sep-2017 15:49:49
Format:IMA

Figure 23.1
Folder Information

Layer No.	1	2	3	4	5	6	7	8
Shimming Within ROI	0.50892	0.48598	0.48596	0.46944	0.46691	0.49944	0.48717	0.46432
MLS Shimming Within ROI	0.59606	0.57773	0.57321	0.53233	0.5083	0.53399	0.53298	0.51982
Phase-only Shimming Within ROI	1	1	1	1	1	1	1	1

Layer No.	9	10	11
Shimming Within ROI	0.4699	0.44007	0.46136
MLS Shimming Within ROI	0.51729	0.491	0.50992
Phase-only Shimming Within ROI	1	1	1

Table 10

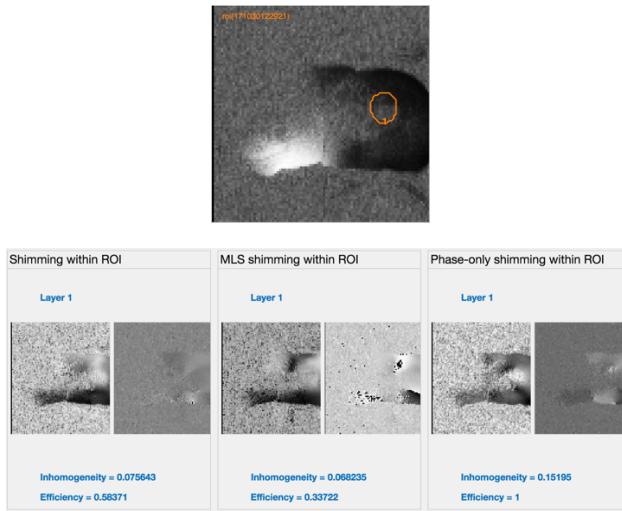


Figure 23.2
Slice 1

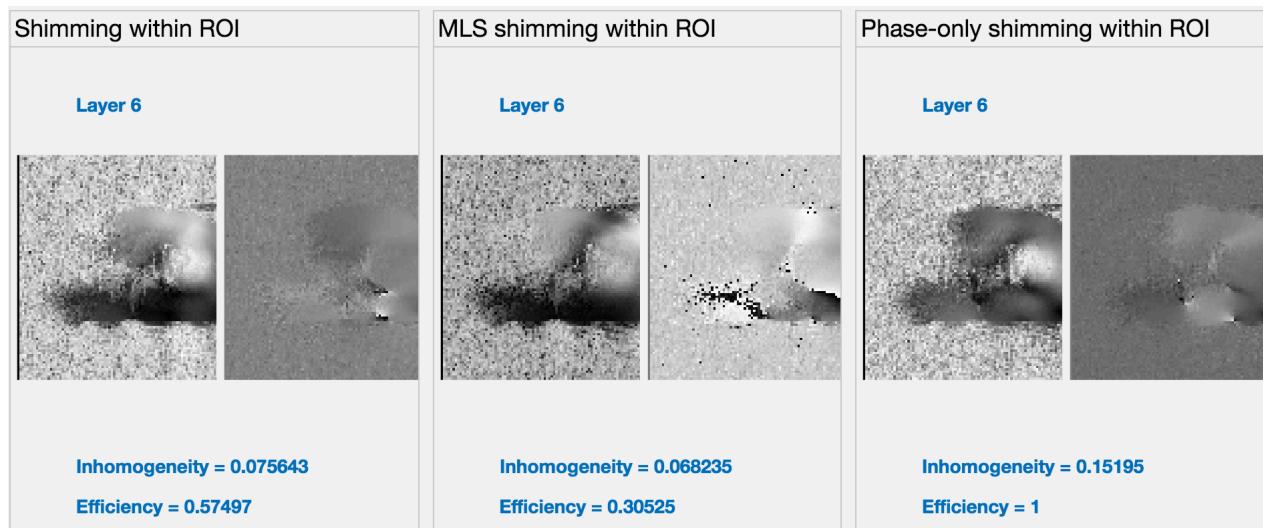
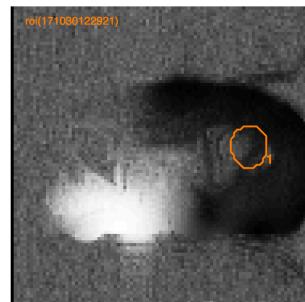
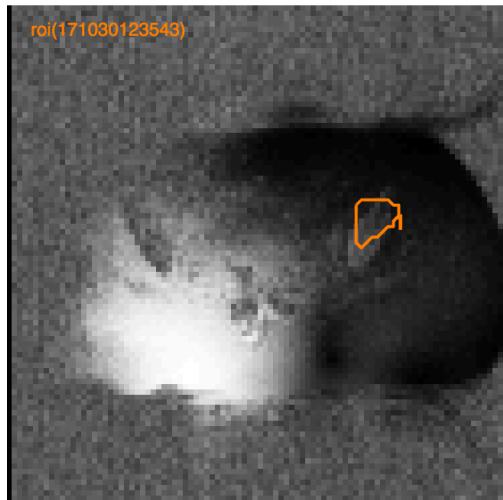


Figure 22.3
Slice 6



Shimming within ROI	MLS shimming within ROI	Phase-only shimming within ROI
Layer 1	Layer 1	Layer 1
Inhomogeneity = 0.108 Efficiency = 0.50892	Inhomogeneity = 0.070315 Efficiency = 0.59606	Inhomogeneity = 0.22684 Efficiency = 1

Figure 22.4
Slice 1

Inhomogeneity of 3 Shims

	Shimming Within ROI	MLS Shimming Within ROI	Phase-only Shimming Within ROI
Inhomogeneity	0.106	0.070315	0.22684

Table 10

Above images are the hip shimming results with the orange ROI. The hip folder information is displayed in Figure 23.1. The Hip has lower image quality than Knee. The first ROI is big; thus, the results are not good. Once the ROI size is reduced, the inhomogeneity is increased. MLS still make scheme be better. And when we compare the efficiency values (Table10) obtained from experiment, for some slices, this ROI is not accurate. In this experiment, the ROI was drawn on

the slice 1, the efficiencies of “Shimming with ROI” and “MLS Shimming with ROI” of slice 1 are the best among of whole table. And the efficiency in “Shimming with ROI” of a slice is higher than other slices also has similar trend in the “MLS Shimming with ROI” row.

Following Figures shows the final schemes obtained from this MATLAB app. The Table 11 shows the schemes from the “ini.” format documents of hip, the data can be used in the MRI software. For the phase-only Shimming, in both of Knee and Hips experiments, the top bars with dark blue are always same (amplitudes are 1), these amplitude values are also related to the efficiency.

	Channel	#Ch 1	#Ch 2	#Ch 3	#Ch 4	#Ch 5	#Ch 6	#Ch 7	#Ch 8
File 1	Magnitude	3.008	3.111	3.351	2.852	3.273	2.582	3.097	4.185
	Phase	0.245	0.5962	1	0.1323	0.2312	0.272	0.4445	0.3134
File 2	Magnitude	6.234	5.294	4.774	4.069	5.422	3.534	5.025	5.161
	Phase	0.1605	0.6125	1	0.4603	0.1868	0.05063	0.3041	0.5276
File 3	Magnitude	2.427	3.455	3.465	2.532	1.154	1.084	1.405	2.844
	Phase	1	1	1	1	1	1	1	1

Table 11

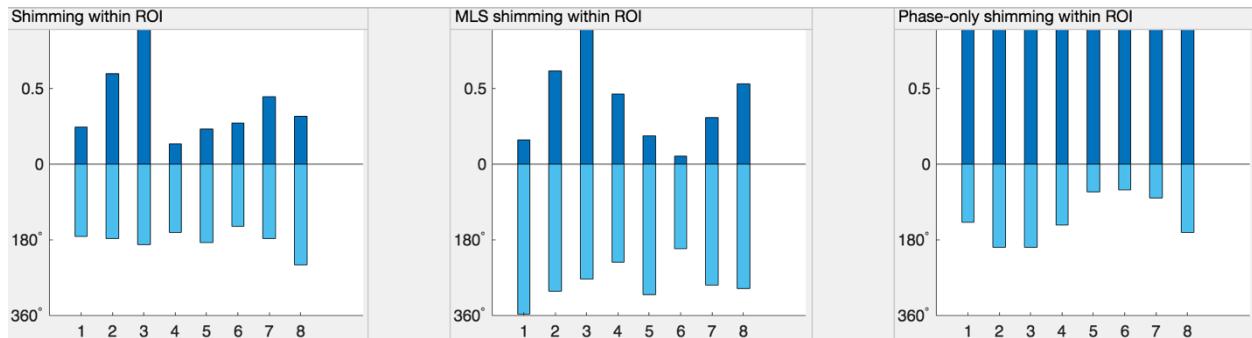


Figure 23.5
Amplitudes and phases of 8 channels

3.1.3 Spine

ID:736998_5983

Number Of Channel:8

Folder Path:/Users/remosy/Dropbox/toYangyang/LumbarSpine/GR__40_B1_rel_B0_2...

Total Number of File:176

Date:01-Aug-2017 16:38:53

Format:IMA

Figure 24.1
Folder Information

Layer No.	1	2	3	4	5	6	7	8
Shimming Within ROI	0.563	0.57303	0.58089	0.58685	0.58485	0.58191	0.5165	0.63002
MLS Shimming Within ROI	0.58442	0.59625	0.60724	0.62125	0.61617	0.60846	0.62236	0.69162
Phase-only Shimming Within ROI	1	1	1	1	1	1	1	1

Layer No.	9	10	11
Shimming Within ROI	0.59615	0.58229	0.59473
MLS Shimming Within ROI	0.63461	0.62134	0.63346
Phase-only Shimming Within ROI	1	1	1

Table 12

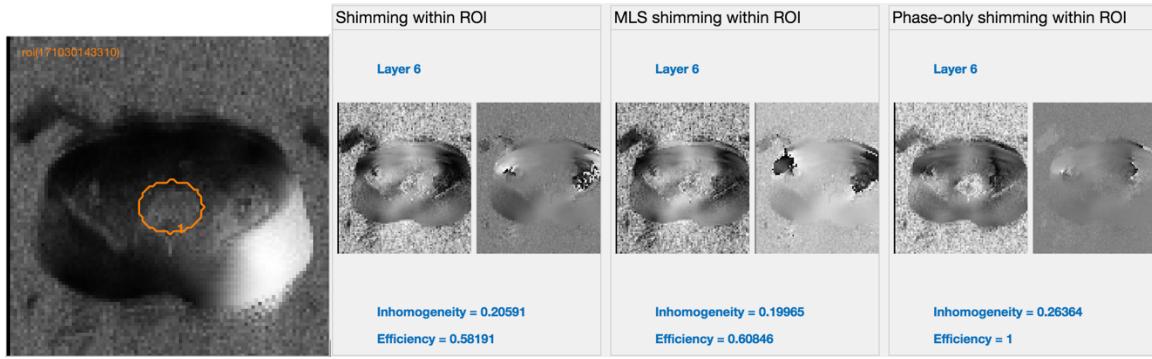


Figure 24.2
Slice 6

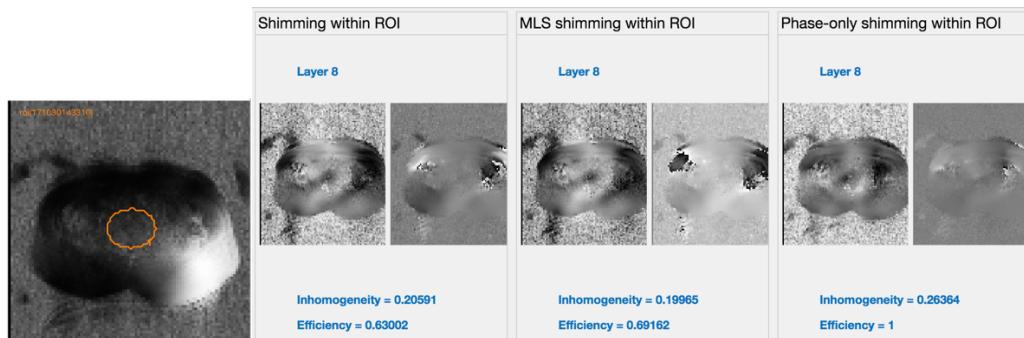


Figure 24.3
Slice 8

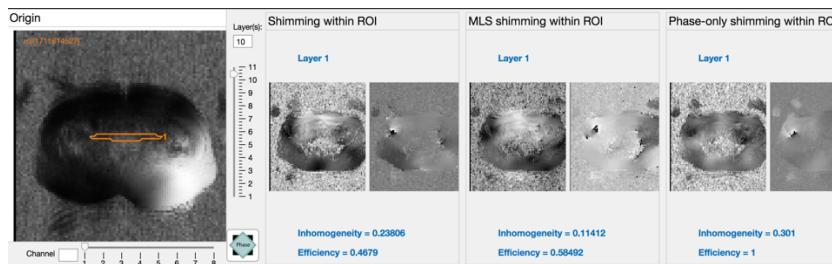


Figure 24.3.1
Slice 1

Above images are the spine shimming results with the orange ROI. The hip folder information is displayed in Figure 24.1. The Hip has lower image quality than Knee. The first ROI is also big, it was located at the centre of spine; the efficiency is around 0.69, which is higher than efficiencies obtained in hip experiment, but the inhomogeneity is still terrible. Once the ROI size is reduced, the inhomogeneity is increased. MLS still make scheme own better performance.

Following Figures shows the final schemes obtained from this MATLAB app. And these schemes (Table 13) can be used in Siemens MRI software. When the image has better homogeneity, the “ini.” Scheme documents can be exported.

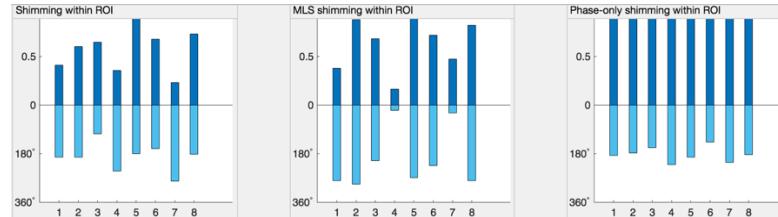


Figure 24.4
Amplitudes and phases of 8 channels

	Shimming Within ROI	MLS Shimming Within ROI	Phase-only Shimming Within ROI
Inhomogeneity	0.20591	0.19965	0.26364

Table 13

	Channel	#Ch 1	#Ch 2	#Ch 3	#Ch 4	#Ch 5	#Ch 6	#Ch 7	#Ch 8
File 1	Magnitude	3.381	3.389	1.858	4.284	3.145	2.817	4.931	3.19
	Phase	0.4082	0.5998	0.6463	0.3528	1	0.6746	0.2287	0.7281
File 2	Magnitude	4.897	5.117	3.614	0.3491	4.695	3.906	0.5279	4.898
	Phase	0.3793	0.8784	0.6809	0.1631	1	0.7154	0.4692	0.819
File 3	Magnitude	3.278	3.095	2.765	3.863	2.387	2.405	3.72	3.199
	Phase	1	1	1	1	1	1	1	1

Table 14

4. Discussion and Conclusion

4.1 Experiment Results Discussion

For most of layers, the MLS does help the efficiency of a shim. However, the efficiency depends on ROI size and area. Often, due to the presence of strong noise in B₁ maps, users need to draw several ROIs in a strategic manner for a better result in efficiency and homogeneity. When only compare these two magnitude-phase schemes, MLS typically has higher homogeneity and efficiency as phase profiles are neglected for better magnitude profiles. The inhomogeneity is also affected by the position of body, such like the hip area, only smaller ROI can achieve better results. All three experiments show the phase-only does not make images be more homogeneity.

In the real experiment, B₁ shims do help correct the image, and makes it be clearer and less inhomogeneity. Figure 24 shows the knee image, which has better image quality (right) after applying shimming.

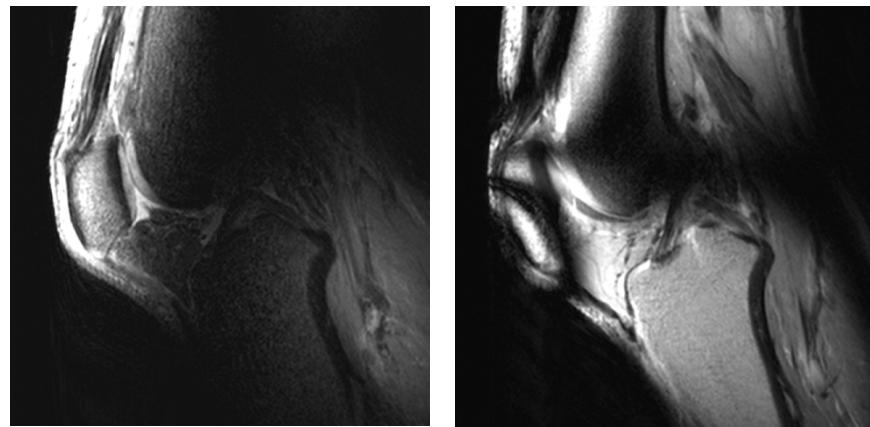


Figure 24 [29]

This thesis has introduced the MATLAB application for B₁ shimming experiment. The advantages and limits of using the ShimTool will be explained from two perspectives.

4.2 Users' Perspective

For researchers, because their experiments are about shimming schemes, at the current stage, the app can provide them three B₁ Shimming results with bar charts for quantitative comparison and

calculated images for visual inspection. The buttons on the GUIs can substitute “evaluate selection” in MATLAB, it saved time to look for code fragment and select same code repeatedly.

As an assistant application, ShimTool provides as much convenience as freedom to users. Such as the ROI toolbox, users can make the ROI mask by easily adding or subtracting any shapes they created. And the “.ini” format scheme files can be generated, users can import them to Siemens software directly.

The visualisations of ShimTool also help users check the results. Such as the blue slice (layer) number, efficiency, and inhomogeneity texts of a layer of channel, users can notice it and easily check the results. Corresponding shimming images are also illustrated between theses texts, this can help users debug their experiments by alter the position, size or shapes of a ROI.

The MATLAB GUI features are still under development. For example, the slider features does not support integer-only scales, which may reduce the quality of user experience.

4.3 Developers’ Perspective

This is an OOP application, if developers hope to improve or further develop the software program, since the basic framework is finished, it is easy to add more code. Developers can simply add a button with a function call to that button.

It is convenient to use the different MATLAB functions, however, along with developing the application, there are more methods duplicated and can be factorised.

4.4 Future Extension

At this stage, ShimTool is the MATLAB application that mainly servers for my supervisor’s project, and the shimming strategies only adopt B_1 shimming.

For the next stage, the kT shimming and its multiple ROIs functionalities will be added. The other functionalities can be developed or improved according to supervisors’ needs.

5. References

1. Dale, B.M., M.A. Brown, and R.C. Semelka, *Production of net magnetization*, in *MRI Basic Principles and Applications*. 2015, John Wiley & Sons, Ltd. p. 1-9.
2. K S Kwan, R., A. Evans, and B. Pike, *MRI simulation-based evaluation of image-processing and classification methods*. Medical Imaging, IEEE Transactions on. 1999. 1085-1097.
3. Dale, B.M., M.A. Brown, and R.C. Semelka, *Principles of magnetic resonance imaging - 2*, in *MRI Basic Principles and Applications*. 2015, John Wiley & Sons, Ltd. p. 39-64.
4. Jones, C.J. and J.R. Thornback, *Chapter 3 Diagnostic Medicine*, in *Medicinal Applications of Coordination Chemistry*. 2007, The Royal Society of Chemistry. p. 101-200.
5. Vaughan, J.T., et al., *Whole-body imaging at 7T: Preliminary results*. Magnetic Resonance in Medicine, 2009. **61**(1): p. 244-248.
6. Moser, E., et al., *7-T MR—from research to clinical applications?* NMR in Biomedicine, 2012. **25**(5): p. 695-716.
7. Balchandani, P. and T.P. Naidich, *Ultra-High-Field MR Neuroimaging*. AJNR. American journal of neuroradiology, 2015. **36**(7): p. 1204-1215.
8. Ugurbil, K., *Magnetic resonance imaging at ultrahigh fields.(Technical report)*. IEEE Transactions on Biomedical Engineering, 2014. **61**(5): p. 1364.
9. Truong, T.K., D. Chakeres, and P. Schmalbrock, *Effects of B 0 and B 1 Inhomogeneity in Ultra-High Field MRI*. 2004.
10. Wiesinger, F., et al., *Potential and feasibility of parallel MRI at high field*. NMR in Biomedicine, 2006. **19**(3): p. 368-378.
11. Vaughan, J.T., et al., *7T vs. 4T: RF power, homogeneity, and signal-to-noise comparison in head images*. Magnetic Resonance in Medicine, 2001. **46**(1): p. 24-30.
12. Jin, J., *Computational Electromagnetics-tailored Inverse Solution to Radiofrequency Issues of High Field MRI*. 2012, The University of Queensland, School of Information Technology and Electrical Engineering.
13. Schmitter, S., et al., *Design of parallel transmission radiofrequency pulses robust against respiration in cardiac MRI at 7 Tesla*. Magnetic Resonance in Medicine, 2015. **74**(5): p. 1291-1305.
14. Vaughan, T., et al., *9.4T human MRI: Preliminary results*. Magnetic Resonance in Medicine, 2006. **56**(6): p. 1274-1282.
15. Keong Li, B., F. Liu, and S. Crozier, *Focused, eight-element transceive phased array coil for parallel magnetic resonance imaging of the chest—Theoretical considerations*. Magnetic Resonance in Medicine, 2005. **53**(6): p. 1251-1257.
16. Mao, W., M.B. Smith, and C.M. Collins, *Exploring the limits of RF shimming for high-field MRI of the human head*. Magnetic Resonance in Medicine, 2006. **56**(4): p. 918-922.
17. van Den Bergen, B., et al., *7 T body MRI: B1 shimming with simultaneous SAR reduction*. Physics in medicine and biology, 2007. **52**(17): p. 5429.
18. Deniz, C.M., et al., *Maximum efficiency radiofrequency shimming: Theory and initial application for hip imaging at 7 tesla*. Magnetic Resonance in Medicine, 2013. **69**(5): p. 1379-1388.

19. Ma, C., et al., *Joint design of spoke trajectories and RF pulses for parallel excitation*. Magnetic Resonance in Medicine, 2011. **65**(4): p. 973-985.
20. Hoyos-Idrobo, A., et al., *On Variant Strategies to Solve the Magnitude Least Squares Optimization Problem in Parallel Transmission Pulse Design and Under Strict SAR and Power Constraints*. Medical Imaging, IEEE Transactions on, 2014. **33**(3): p. 739-748.
21. Cloos, M.A., et al., *Parallel-transmission-enabled magnetization-prepared rapid gradient-echo T1-weighted imaging of the human brain at 7 T*. NeuroImage, 2012. **62**(3): p. 2140-2150.
22. Yip, C.-y., J.A. Fessler, and D.C. Noll, *Advanced three-dimensional tailored RF pulse for signal recovery in T2*-weighted functional magnetic resonance imaging*. Magnetic Resonance in Medicine, 2006. **56**(5): p. 1050-1059.
23. Hsu, Y.-C., et al., *Mitigate B1+ inhomogeneity using spatially selective radiofrequency excitation with generalized spatial encoding magnetic fields.(Author abstract)*. Magnetic Resonance in Medicine, 2014. **71**(4): p. 1458.
24. Cloos, M.A., et al., *kT-points: Short three-dimensional tailored RF pulses for flip-angle homogenization over an extended volume*. Magnetic Resonance in Medicine, 2012. **67**(1): p. 72-80.
25. Grissom, W.A., et al., *Small -tip -angle spokes pulse design using interleaved greedy and local optimization methods*. Magnetic Resonance in Medicine, 2012. **68**(5): p. 1553-1562.
26. Cloos, M.A., et al., *Local SAR reduction in parallel excitation based on channel-dependent Tikhonov parameters*. Journal of magnetic resonance imaging : JMRI, 2010. **32**(5): p. 1209.
27. Panych, L.P. and B. Madore, *The physics of MRI safety*. Journal of Magnetic Resonance Imaging: p. n/a-n/a.
28. Wang, C. and G.X. Shen, *B1 field, SAR, and SNR comparisons for birdcage, TEM, and microstrip coils at 7T*. Journal of Magnetic Resonance Imaging, 2006. **24**(2): p. 439-443.
29. Jin, J., et al., *An open 8-channel parallel transmission coil for static and dynamic 7T MRI of the knee and ankle joints at multiple postures*. Magnetic Resonance in Medicine, 2017: p. n/a-n/a.
30. Yang, Q.X., et al., *Analysis of wave behavior in lossy dielectric samples at high field*. Magnetic Resonance in Medicine, 2002. **47**(5): p. 982-989.
31. Setsompop, K., et al., *Magnitude Least Squares Optimization for Parallel Radio Frequency Excitation Design Demonstrated at 7 Tesla With Eight Channels*. Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine, 2008. **59**(4): p. 908-915.

6. Appendix

<<Interface>>Main_gui	<<Interface>>ROI_gui
<pre>+ figh: double + figw: double + imageData: ImageData - f: matlab.ui.Figure - gui_ROI: ROI_gui - pickLayer_list: cell - mainEvent: char - isFirstShim: double - pos: double - CN: matlab.ui.control.NumericEditField - channel_fig: matlab.ui.Figure - current_layer: double - current_channel: double - shimList: double - shimList_B1p: double - mask_shm: logical - hwar: double - topGroup: matlab.ui.container.TabGroup - tab_home: matlab.ui.container.Tab - tab_editor: matlab.ui.container.Tab - tab_help: matlab.ui.container.Tab - imagePanelGroup: - channel: matlab.ui.container.Panel - slider_channel: matlab.ui.control.Slider - layerFrom: matlab.ui.control.TextArea - layerTo: double - slider_layer: matlab.ui.control.Slider - button_angleState: matlab.ui.control.StateButton - imageTabGroup: matlab.ui.container.TabGroup + Main_gui():Main_gui + delete(): void + set.figh(double: height); - helper_drawROI(matlab.ui.container.internal.UIContainer:axes): void - DisplayChannelImage(double:value): void - DisplayLayerImage(double:value): void - calbkLayer1(matlab.ui.control.TextArea: text): void - calbkLayer(matlab.ui.eventdata.ValueChangingData: event): void - calbkChannel1(matlab.ui.control.TextArea: text): void - calbkChannel2():void - updatePOOnLayers(matlab.ui.control.ListBox:src,matlab.ui.eventdata.ValueChangedData:event): void - calbkUpdateROOnLayers(): void - DisplayBasicInfo(char:name,char:str_folder,char:str_size,char:str_date,cell:str_format): void - helper_ShimSlider(double:val,matlab.ui.container.Panel:get:ShimPanels): void - calbkShimSlider(matlab.ui.eventdata.ValueChangingData: event): void - DisplayShim(logical:mask,double:shim,char: shim_name): void - PlotShimBar(double:shim,char:shim_name): void - RemovevImage(char:name): void - DisplayImage(char:name,double:img): void - calbkAngleImage(matlab.ui.eventdata.ValueChangedData: event): void - calbkInputCN(): void - exportHelper(double:shim): void - calbkExport(): void - calbkAddShim(): void - calbkPatient(): void - calbkShimLists(): void - calbkROIToolBox(): void - calbkAnalysis(): void - createWindow(): void</pre>	<pre>+ image: double + roi: double + labels: double + number: double + figw: double + roi_result: ROIMask - f: matlab.ui.Figure - imax_axes: matlab.graphics.axis.Axes - roiax_axes: matlab.graphics.axis.Axes - imag: matlab.graphics.primitive.Image - roifig: matlab.graphics.primitive.Image - figh: double - hwar: double - loadmask: double - mask: logical - current: double - roi_mask: cell - selectedROI: double - filename: char - currentGui: matlab.ui.container.Panel - buttons: double - isSubtract: double + ROI_gui(double: image, ROIMask: roiObj): ROI_gui + delete(): void + set.image(double: image): void + set.figh(double: height); + getROIData(int:varargin): - resetImages(): void - updateROI(): void - newShapeCreated(): void - winpressed(matlab.ui.Figure:h,matlab.ui.eventdata.WindowMouseData:e,char:type): void - closefig(): void - drawROI(logical:mask, char:id): matlab.graphics.primitive.Image, matlab.graphics.GraphicsPlaceholder, matlab.ui.container.internal.UIContainer, matlab.graphics.GraphicsPlaceholder - findImgGui(): matlab.ui.container.internal.UIContainer - calbkLoadROI(matlab.ui.control.ListBox: src, matlab.ui.eventdata.ValueChangedData: e): void - calbkConfirmROI(matlab.ui.eventdata.ValueChangedData: e): void - calbkSave(matlab.ui.eventdata.ActionData: e): void - calbkLoad(matlab.ui.eventdata.ActionData: e): void - calbkClear(matlab.ui.eventdata.ActionData: e): void - polyclick(matlab.ui.control.UIControl:h, matlab.ui.eventdata.ActionData: e): void - elliclick(matlab.ui.control.UIControl:h, matlab.ui.eventdata.ActionData: e): void - freeclick(matlab.ui.control.UIControl:h, matlab.ui.eventdata.ActionData: e): void - rectclick(matlab.ui.control.UIControl:h, matlab.ui.eventdata.ActionData: e): void - calbkUndo(matlab.ui.control.UIControl:h, matlab.ui.eventdata.ActionData: e): void - add_sub(matlab.ui.control.UIControl:h, matlab.ui.eventdata.ActionData: e): void - createWindow(): void</pre>