Technische Universität Wien

MASTERARBEIT

ΦUN

A Region Growing Phase Unwrapping Program Optimized for MR Images

ausgeführt an der

Universitätsklinik für Radiologie
der Medizinischen Universität Wien

und in der

Arbeitsgruppe Medizinische Physik des
Institutes für Diagnostische und Interventionelle Radiologie
der Friedrich Schiller Universität Jena

unter der Anleitung von

Univ. Doz. DI Dr. Markus Barth

durch

Stephan Witoszynskyj, Bakk.techn.
Zimmermanngasse 22
1090 Wien

Wien, 1. September 2007

Kurzfassung

Eine Besonderheit von Magnet Resonanz (MR) Daten ist, dass sie, im Gegensatz zu vielen anderen bildgebenden Verfahren, komplexwertig sind. Trotzdem nutzen die meisten Anwendungen nur die Magnitudeninformation. Nichtsdestotrotz gibt es aber Anwendungen, die auf der Phaseninformation basieren, da die Phase sehr sensitiv ist. In diesen Anwendungen stellt das Aliasing (auch Phase-Wrapping genannt), das auf Grund des beschränkten Definitionsbereiches der Phase auftritt, ein schwieriges Problem für die Nachbearbeitung dar. Die Beseitigung der resultierenden Mehrdeutigkeiten wird als Phase Unwrapping bezeichnet.

Im Rahmen dieser Masterarbeit wurde ein zweidimensionaler Region-Growing Phase-Unwrapping-Algorithmus für MR-Phasenbilder optimiert. Das Ziel war es eine Methode zu entwickeln, die besonders für suszeptibilitätsgwichtete Bildgebung (SWI) geeignet ist. SWI ist eine spezielle MR Technik, die auf hochaufgelösten Daten basiert, die oft ein niedriges Signal zu Rausch Verhältnis (SNR) aufweisen. Die Implementierung des Algorithmus, ΦUN, wurde ausgiebig an Phantom und in vivo Daten getestet und mit einer etablierten Methode verglichen. Gerade bei hochaufgelösten Daten mit niedrigem SNR war ΦUN sowohl in Bezug auf Geschwindigkeit als auch im Sinne der Verlässlichkeit der Ergebnisse überlegen.

Abstract

Magnetic Resonance Imaging (MRI) data, unlike the data of many other imaging modalities, are complex valued. Nevertheless, in most applications only the magnitude information is used. However, since the phase is very sensitive, there are techniques that are based on phase images. For data acquired with those techniques aliasing (also known as phase wrapping) due to the limited domain of the phase poses a difficult problem to post processing. Resolving the resulting ambiguities is known as phase unwrapping.

For this thesis a two-dimensional region growing phase unwrapping algorithm was optimized for MR phase images. The goal was to develop a method that is especially suited for Susceptibility Weighted Imaging (SWI). SWI is a special MR technique that is based on high-resolution data which often have a low signal-to-noise ratio (SNR). The implementation of the algorithm, ΦUN, was evaluated extensively on both phantom and in vivo data, and compared to an established method. ΦUN's performance, both in terms of speed and robustness, proved to be superior especially in the case of high resolution, low SNR data, such as SWI data.

Danksagung

Diese Arbeit, und damit auch der Abschluss meines Studiums, wäre mir ohne die Unterstützung durch eine Vielzahl von Personen nicht möglich gewesen. Ich bedanke mich bei all jenen, die mir auf die eine oder andere Art und Weise geholfen haben, auch wenn hier nicht alle namentlich Erwähnung finden.

Zuerst möchte ich mich bei meinem Betreuer Univ. Doz. DI Dr. Markus Barth, für seine Geduld, seine Unterstützung und die vielen hilfreichen Diskussionen und Hinweise bedanken.

Ein besonderes Danke gebührt DI Dr. Alexander Rauscher, der überhaupt erst mein Interesse für MR und für das Problem des Phase Unwrapping geweckt hat. Ich danke ihm für die vielen Gespräche und die gute Zusammenarbeit; auch ist er dafür verantwortlich, dass es mich nach einigen Irrfahrten über Urbana-Champaign und Genf nach Jena verschlagen hat.

In diesem Zusammenhang möchte ich mich ganz besonders bei Prof. Dr. Jürgen Reichenbach bedanken, der mich in seine Arbeitsgruppe in Jena aufgenommen hat. Seinen besonderen Führungsstil und seine Art seine Mitarbeiter zu motivieren habe ich sehr zu schätzen gelernt.

Allen Mitgliedern der Arbeitsgruppe medizinische Physik des Institutes für diagnostische und interventionelle Radiologie der Friedrich-Schiller Universität Jena danke ich für die freundschaftliche Aufnahme, die gute Zusammenarbeit und dass sie mir immer mit Rat und Tat zur Seite standen.

Zu besonderem Dank bin ich dem Interdisziplinären Zentrum für Klinische Forschung (IZKF) des Universtitätsklinikums Jena verpflichtet. Dieses ermöglichte mir meinen Forschungsaufenthalt an der Universität Jena im Rahmen eines Nachwuchsförderungsprogrammes (IZKF S-10 Projekt).

Dr. Petra Kullnig – für mich “die Beste aller Radiologinnen” – danke ich für die freundschaftliche Unterstützung und ihren Rat – und dafür, dass sie für mich das “grüne Herz Österreichs” mitten in Deutschland war.

Meinen Eltern, Mag. Elisabeth und Dr. Theodor Witoszynskyj, danke ich ganz besonders. Nicht nur für die Unterstützung meines Studiums – wofür ich ihnen danke möchte, lässt sich weder aufzählen noch in Worte fassen. Ich kann es vielleicht am besten so ausdrücken “ich danke ihnen dafür, dass sie meine Eltern sind”.

Contents

1 Introduction

1.1 Motivation

Magnetic resonance imaging (MRI) probably is the most powerful, but also the most complex, tool available for diagnostic imaging today. One reason for this is that MRI is sensitive to a large number of physical effects. This, on one hand, allows for an ever wider range of applications, but on the other hand makes it challenging to design robust data acquisition and image processing tools and protocols.

Although the data acquired by MRI consist of complex values, in clinical routine, only magnitude images are reconstructed most of the time. However, the phase is extremely sensitive and can thus be of great value. Phase information can be exploited to detect small inhomogeneities of the magnetic field caused by tissue interfaces [1, 2, 3], but also to measure large scale field inhomogeneities [4, 5]. This information can be applied to correct for distortions. Phase information can also be utilized to measure flow [6] or mechanic properties of tissue [7]. Phase images can also facilitate the segmentation of tissues [8].

Because of the limited domain of the phase, aliasing can occur. Aliasing itself is not only a problem if the underlying physical quantity is to be measured, but also if relative changes of the phase are of interest. The latter is the case in many applications, such as Susceptibility Weighted Imaging (SWI). In those applications phase aliasing causes these changes to appear larger than they actually are. Aliasing artifacts appear as sharp edges in phase images. They are commonly known as phase wraps. The removal of phase wraps in two or more dimensional data is a non-trivial problem known as phase unwrapping. It arises in fields ranging from optics, over radar interferometry to MRI.

In MRI, there exist several situations in which phase unwrapping can be circumvented. If phase images are used for field mapping, phase unwrapping often can be avoided by computing the complex division of two complex images instead of subtracting two unwrapped phase images [9, 10]. In SWI, homodyne detection [11] often is applied [12]. However, these approaches do not provide true unwrapping and can be subject to aliasing themselves. Latter usually occurs in the presence of strong field inhomogeneities.

A number of methods for true phase unwrapping have been described in the literature (e.g. [13, 14, 15, 16, 17, 18]). In the MRI community, PRELUDE [16] is a well established phase unwrapping program that is freely available on the internet. PRELUDE was developed for unwrapping of field maps for distortion correction of Echo Planar Imaging (EPI) data, which requires relatively low resolution phase images. However, its performance was found to be not sufficient for high resolution applications such as SWI. For those applications not only unwrapping speed is of concern, but also the performance in low Signal-to-Noise Ratio (SNR) areas.

The aim of the work described in this thesis was to develop a method that provides fast phase unwrapping for high resolution MRI applications such as SWI that often are affected by low SNR. The program that I have implemented is called ΦUN which stands for "Phase UNwrapper".

1.2 Overview

This master thesis is structured as follows:

Chapter 2 and 3 give an overview of the physical concepts and principles of MRI. An understanding of those foundations is necessary for appreciating difficulties and limitations of MRI data,

Chapter 4 explains how complex images are obtained from measuring physical quantities which are real valued. The effects of noise are described and problem caused by the limited domain of the phase – phase wraps – as well as approaches to solve this problem are discussed. Finally, the challenges posed by modern fast image acquisition techniques are discussed.

Chapter 5 describes the idea behind phase unwrapping by a region growing algorithm and the problems observed when this approach is applied to MRI data. Suggestions for optimizing this approach for MRI are introduced and evaluated.

Chapter 6 discusses some of the details of the implementation of the approach described in the previous chapter. The implementation was named ΦUN. The implementation will be referred to by this name throughout the thesis.

Chapter 7 describes the evaluation of ΦUN and the comparison with an established method, namely PRELUDE.

Chapter 8 discusses properties of the algorithm as well as potential applications, and, finally,

Chapter 9 summarizes the conclusions drawn from the work described in this thesis.

List of References

[1] Rauscher, A., *et al. Magnetic susceptibility-weighted MR phase imaging of the human brain.* American Journal of Neuroradiology, **26**(4), 736–742 (2005).

[2] Reichenbach, J. R., Venkatesan, R., Schillinger, D., and Haacke, E. M. *Small vessels in the human brain: MR-venography with deoxyhemoglobin as an intrinsic contrast agent.* Radiology, **204**, 272–277 (1997).

[3] Duyn, J. H., *et al. High-field MRI of brain cortical substructure based on signal phase.* Proceedings of the National Academy of Sciences, **104**(28), 11796–11801 (2007).

[4] Johnson, G. and Hutchison, J. M. S. *The limitations of NMR recalled-echo imaging techniques.* Journal of Magnetic Resonance, **63**(1), 14–30 (1985).

[5] Jezzard, P. and Balaban, R. S. *Correction for geometric distortion in echo planar images from B_0 field variations.* Magnetic Resonance in Medicine, **34**, 65–73 (1995).

[6] Bryant, D. J., Payne, J. A., Firmin, D. N., and Longmore, D. B. *Measurement of flow with NMR imaging using a gradient pulse and phase difference technique.* Journal of Computer Assisted Tomography, **8**(4), 588–593 (1984).

[7] Kruse, S. A., *et al. Tissue characterization using magnetic resonance elastography: preliminary results.* Physics in Medicine and Biology, **45**(6), 1579–1590 (2000).

[8] Bourgeat, P., *et al. MR image segmentation of the knee bone using phase information.* Medical Image Analysis, **11**(4), 325–335 (2007).

[9] Cusack, R. and Papadakis, N. *New robust 3–D phase unwrapping algorithms: Application to magnetic field mapping and undistorting echoplanar images.* NeuroImage, **16**, 754–764 (2002).

[10] Hutton, C., *et al. Image distortion correction in fMRI: A quantitative evaluation.* NeuroImage, **16**(1), 217–40 (2002).

[11] Noll, D. C., Nishimura, D. G., and Makovski, A. *Homodyne detection in magnetic resonance imaging.* IEEE Transactions on Medical Imaging, **10**, 154–163 (1991).

[12] Reichenbach, J. R. and Haacke, E. M. *High-resolution BOLD venographic imaging: a window into brain function.* NMR in Biomedicine, **14**(7-8), 453–467 (2001).

[13] Hedley, M. and Rosenfeld, D. *A new two-dimensional phase unwrapping algorithm for MRI images.* Magnetic Resonance in Medicine, **24**(1), 177–181 (1992).

[14] An, L., Xiang, Q.-S., and Chavez, S. *A fast implementation of the minimum spanning tree method for phase unwrapping.* IEEE Transactions on Medical Imaging, **19**(8), 805–808 (2000).

[15] Chavez, S., Xiang, Q.-S., and An, L. *Understanding phase maps in MRI: a new cutline phase unwrapping method.* IEEE Transactions on Medical Imaging, **21**(8), 966–977 (2002).

[16] Jenkinson, M. *Fast, automated, N-dimensional phase-unwrapping algorithm.* Magnetic Resonance in Medicine, **49**(1), 193–197 (2003).

[17] Volkov, V. V. and Zhu, Y. *Deterministic phase unwrapping in the presence of noise.* Optics Letters, **28**(22), 2156–2158 (2003).

[18] Xu, W. and Cumming, I. *A region-growing algorithm for InSAR phase unwrapping.* IEEE Transactions on Geoscience and Remote Sensing, **37**(1), 124–134 (1999).

2

Theory of Nuclear Magnetic Resonance

Nuclear magnetic resonance (NMR) and magnetic resonance imaging (MRI) are complex techniques that exploit a number of physical effects and properties. Thus, a description of the theory of NMR has to include quantum mechanics, electrodynamics and statistical mechanics. As the name NMR implies the very starting point of such a description has to be the magnetic behaviour of nuclei. This behaviour is a result of a quantum mechanical property called spin that can be attributed to all kind of particles. Nevertheless, due to the Ehrenfest theorem which states that classical physics can be applied to the expectation value of a quantum variable, and the large number of nuclei present in any NMR experiment, a large fraction of NMR can be described in terms of classical mechanics which will be done most of the times throught these sections.

2.1 Spin, Angular Momentum and Magnetic Moment

Angular Momentum in Classical Mechanics

Classical mechanics distinguishes between two different types of rotation of a rigid body: rotation about an external axis and rotation about the center of mass. A quantity named angular momentum can be attributed to each type of rotation (see for example [1]). In the first case it is called orbital angular momentum $\boldsymbol{L}_{\text{orbital}}$ and is calculated as

$$\boldsymbol{L}_{\text{orbital}} = \boldsymbol{r} \times \boldsymbol{p} \,, \tag{2.1}$$

where $\boldsymbol{r}$ is the distance of the center of mass from the axis of rotation and $\boldsymbol{p}$ the momentum. In the second it is called spin $\boldsymbol{L}_{\text{spin}}$ and is given by

$$\boldsymbol{L}_{\text{spin}} = I\boldsymbol{\omega} \ , \tag{2.2}$$

with I being the moment of inertia and $\boldsymbol{\omega}$ the angular frequency. In principle, there is no difference between the two angular momenta. For a given object and a given axis of rotation, the spin angular momentum (eq. 2.2) can be derived from the formula for the orbital angular moment (eq. 2.1) by integrating over the object.

Orbital angular momentum $\boldsymbol{L}_{\text{orbital}}$ and spin angular momentum $\boldsymbol{L}_{\text{spin}}$ can be added up to form a total angular momentum $\boldsymbol{L}_{\text{total}}$

$$\boldsymbol{L}_{\text{total}} = \boldsymbol{L}_{\text{orbital}} + \boldsymbol{L}_{\text{spin}} \ . \tag{2.3}$$

The total angular momentum $\boldsymbol{L}_{\text{total}}$ is a conserved quantity.

Magnetic Moment

The magnetic (dipole) moment $\boldsymbol{\mu}$ [2] is defined as

$$\boldsymbol{\mu} = \frac{1}{2} \int \mathrm{d}^3 r \ \boldsymbol{r} \times \boldsymbol{j}(\boldsymbol{r}) \tag{2.4}$$

with $\boldsymbol{j}(\boldsymbol{r})$ being a current density. It relates the torque $\boldsymbol{\tau}$ acting upon a current distribution in a magnetic field $\boldsymbol{B}$ to this field:

$$\boldsymbol{\tau} = \boldsymbol{\mu} \times \boldsymbol{B} \tag{2.5}$$

For a simple loop current, equation 2.4 can be simplified to

$$\boldsymbol{\mu} = j\boldsymbol{A} \ , \tag{2.6}$$

where $\boldsymbol{A}$ is the area enclosed by the loop and j the current.

It can be shown easily that, in case of a rotating and electrically charged object, the magnetic moment is related to the angular momentum: let's consider an infinitely thin ring of radius r, mass m that has a charge q distributed around the whole circumference. The ring rotates about its center with an angular velocity of $\boldsymbol{\omega}$. For simplicity the ring is assumed to lie within the x-y plane and to be centered about the z-axis (figure 2.1). As a result the vector equations can be simplified to scalar equations.

The moment of inertia I can be calculated to be

$$I = mr^2 \ . \tag{2.7}$$

The angular momentum hence is

$$L = I\omega = mr^2\omega \ . \tag{2.8}$$

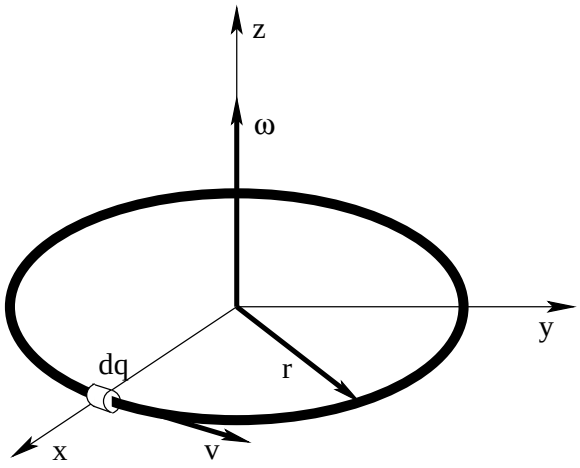


Figure 2.1: A ring that has a charge q, a mass m and is rotating about the z-axis has a magnetic moment μ that is parallel to the axis of rotation and proportional to the ring's charge, mass and angular momentum.

The rotation of the charged ring corresponds to an electric current j circulating in a loop. The current is given by

$$j = \frac{q\omega}{2\pi} \, . \tag{2.9}$$

Using equation 2.6, the ring's magnetic moment μ can be thus written as

$$\mu = \pi r^2 j = r^2 \frac{q\omega}{2} \, . \tag{2.10}$$

ω can then be substituted using equation 2.8 leading to

$$\mu = \frac{r^2 q L}{2I} = \frac{q}{2m} L = \gamma L \, . \tag{2.11}$$

γ is called the gyromagnetic ratio. It can be shown that the relation $\gamma = \frac{q}{2m}$ holds for any object as long as both charge and mass are distributed uniformly.

Angular Momentum in Quantum Mechanics

As in classical mechanics, in quantum mechanics, one has to distinguish between orbital and spin angular momentum. The operators for the orbital angular momentum can be derived from the classical orbital angular momentum (eq. 2.1) [3] and are given by

$$L_x = \frac{\hbar}{i}\left(y\frac{\partial}{\partial z} - z\frac{\partial}{\partial y}\right), \;\; L_y = \frac{\hbar}{i}\left(z\frac{\partial}{\partial x} - x\frac{\partial}{\partial z}\right) \;\text{ and }\; L_z = \frac{\hbar}{i}\left(x\frac{\partial}{\partial y} - y\frac{\partial}{\partial x}\right) \, . \tag{2.12}$$

	particle(s)	spin
bosons	photon	1
	8 gluons	1
	$W^{\pm}$, Z^0	1
	graviton	2
fermions	ν_e, ν_μ, ν_τ	$\frac{1}{2}$
	e^-, μ^-, τ^-	$\frac{1}{2}$
	u, d, s, c, b, t	$\frac{1}{2}$

Table 2.1: Fundamental particles and their spins.

It can be shown easily that the angular momentum's components L_x, L_y and L_z do not commute

$$[L_x, L_y] = i\hbar L_z, \quad [L_y, L_z] = i\hbar L_x, \quad [L_z, L_x] = i\hbar L_y \; . \tag{2.13}$$

This means that L_x, L_y and L_z are incompatible observables and, because of Heisenberg's uncertainty principle, cannot be measured at the same time. As a consequence $\boldsymbol{L}$ cannot be known. Nevertheless, each component L_x, L_y and L_z commutes with the squared total angular momentum

$$L^2 = L_x^2 + L_y^2 + L_z^2 \tag{2.14}$$

$$\left[L^2, \boldsymbol{L}\right] = 0 \; . \tag{2.15}$$

Thus, L^2 and one of the components can be measured at the same time. This means that functions exist that are eigenfunctions of both L^2 and L_z (or any other component).[1] The eigenequations can be derived to be

$$L^2|\psi\rangle = \hbar^2 l\,(l+1)\,|\psi\rangle \quad \text{and} \quad L_z|\psi\rangle = \hbar m_l|\psi\rangle \tag{2.16}$$

with

$$l = 0, 1, 2, \ldots \quad \text{and} \quad m_l = -l, -l+1, \ldots, l-1, l \; . \tag{2.17}$$

This means that both L^2 and L_z are quantized and can assume discrete values only.

The mathematical theory of the intrinsic angular momentum – the spin $\boldsymbol{S}$ – basically is the same as the one of the orbital angular momentum. As in the case of the angular momentum, the components (S_x, S_y and S_z) do not commute

$$[S_x, S_y] = i\hbar S_z, \quad [S_y, S_z] = i\hbar S_x, \quad [S_z, S_x] = i\hbar S_y \tag{2.18}$$

and, thus, cannot be measured at the same instant. The eigenvectors of S^2 and S_z satisfy

$$S^2|sm_s\rangle = \hbar^2 s\,(s+1)\,|sm_s\rangle \quad \text{and} \quad S_z|sm_s\rangle = \hbar m_s|sm_s\rangle \tag{2.19}$$

[1] In text books, the z-component is used as a "synonym" for the components of the angular momentum.

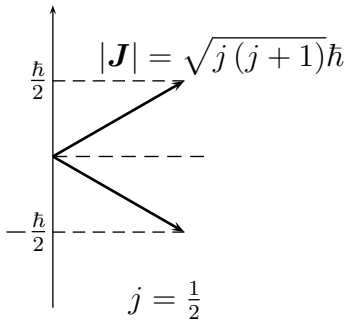


Figure 2.2: The schematic relation between $\boldsymbol{J}$, $|J|$ and J_z for a particle with a total angular momentum number of $j = \frac{1}{2}$ (e.g. $l = 0$ and $s = \frac{1}{2}$).

with s and m_s having values of

$$s = 0, \frac{1}{2}, 1, \frac{3}{2}, \ldots \quad \text{and} \quad m_s = -s, -s+1, \ldots, s-1, s\,. \tag{2.20}$$

The value of s is an immutable property of fundamental particles. All fundamental particles have either $s = \frac{1}{2}$, $s = 1$ or $s = 2$ (tab. 2.1). Fundamental particles having integer spin are called gauge bosons and are subject to the Bose-Einstein statistics. They act as mediator of the four fundamental forces. Particles having spin $s = \frac{1}{2}$ are called fermions. They are subject to the Fermi statistics. An important consequence of the Fermi statistics is Pauli's exclusion principle.

Spins of two or more particles can be added up to form a total angular momentum. For a two particle system the resulting total spin quantum number s can assume values from $|s_1 - s_2|$ to $(s_1 + s_2)$:

$$s = |s_1 - s_2|\,, |s_1 - s_2| + 1, \ldots, (s_1 + s_2) - 1, (s_1 + s_2) \tag{2.21}$$

This equation can be generalized for systems consisting of more than two particles. In case of a combination of three quarks, for example u, u and d, the possible total angular momentum (assuming all quarks have zero orbital angular momentum) can be either $s = \frac{1}{2}$ or $s = \frac{3}{2}$. These two possibilities are considered to be two distinct particles: the proton in case of $s = \frac{1}{2}$ and Δ^+ in case of $s = \frac{3}{2}$.

Spin $\boldsymbol{S}$ and angular momentum $\boldsymbol{L}$ can couple to form the total angular momentum

$$\boldsymbol{J} = \boldsymbol{L} + \boldsymbol{S}\,. \tag{2.22}$$

Just as in the case of angular momentum $\boldsymbol{L}$ and spin $\boldsymbol{S}$, J^2 and J_z fulfill the eigenvalue equations

$$J^2|jm_j\rangle = \hbar^2 j\,(j+1)\,|jm_j\rangle \quad \text{and} \quad J_z|jm_j\rangle = \hbar m_j|jm_j\rangle \tag{2.23}$$

with

$$j = |l - s|\,, |l - s| + 1, \ldots, l + s - 1, l + s \quad \text{and} \quad m_j = -j, -j+1, \ldots, j-1, j\,. \tag{2.24}$$

Z	A	J	$^{A}_{Z}$Nucleus
even	even	0	$^{12}_{6}$C, $^{16}_{8}$O
odd	odd	$\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \ldots$	$^{1}_{1}$H, $^{15}_{7}$N, $^{19}_{9}$F, $^{31}_{15}$P
even	odd	$\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \ldots$	$^{13}_{6}$C, $^{17}_{8}$O
odd	even	$1, 2, 3, \ldots$	$^{2}_{1}$H, $^{14}_{7}$N

Table 2.2: Angular momentum quantum numbers of some important nuclei (taken from [6]).

The schematic relation between $\boldsymbol{J}$, $|J|$ and J_z for a particle having $s = \frac{1}{2}$ and $l = 0$ and thus $j = \frac{1}{2}$ is shown in figure 2.2. In case of atoms, the total angular momentum $\boldsymbol{J}$ gives rise to the anomalous ZEEMAN effect. This effect is a consequence of different orientations of $\boldsymbol{J}$ with respect to an external magnetic field $\boldsymbol{B}$ having slightly differing energies.

In case of nuclei, the total angular momentum can be calculated using the nuclear shell model (see for example [4] or [5]). According to this model, the spins of protons and neutrons have to be summed up separately. The total spin $\boldsymbol{S}$ and the total angular momentum $\boldsymbol{L}$ are then coupled to form a total angular momentum $\boldsymbol{J}$ (the values of $\boldsymbol{J}$ for several selected nuclei is given in table 2.2).[2] Often, especially in MR literature, the total angular momentum $\boldsymbol{J}$ of a nucleus is referred to as spin. This nomenclature will be used in the following sections.

Spin and Magnetic Moment

Just as there is a magnetic dipole moment $\boldsymbol{\mu}$ associated with the classical angular momentum, a magnetic dipole moment $\boldsymbol{\mu}$ can be attributed to the spin of a particle.[3] The proportionality is given by an equation similar to equation 2.11:

$$\boldsymbol{\mu} = \gamma \boldsymbol{S} \tag{2.25}$$

where γ is the gyromagnetic ratio. The magnetic moment is quantized in the same way as the spin $\boldsymbol{S}$.[4]

[2] This scheme does not hold for heavier nuclei, because these exhibit jj coupling. In this case, the spin $\boldsymbol{S}_i$ and the angular momentum $\boldsymbol{L}_i$ of each nucleon have to be coupled first to form a $\boldsymbol{J}_i$. The $\boldsymbol{J}_i$ then couple together to form the total angular momentum $\boldsymbol{J}$.

[3] Particles do not necessarily have to be electrically charged to have a magnetic dipole moment. For example the neutron has no net charge, but a magnetic dipole moment $\mu = -1.9130427 \pm 0.0000005 \mu_N$ [7]. The reason for this is that the neutron is composed out of three quarks, which all have a non-zero charge.

[4] Historically, spin was introduced by GOUDSMIT and UHLENBECK to explain the anomalous ZEEMAN effect. According to their theory, spin is associated with a magnetic momentum. The quantization of spin and the magnetic momentum was shown experimentally by STERN and GERLACH in the famous experiment named after them.

The gyromagnetic ratio γ is an intrinsic property of each particle. For the electron it is given by

$$\gamma_\mathrm{e} = g_\mathrm{e}\frac{\mu_\mathrm{B}}{\hbar} \ . \tag{2.26}$$

μ_B is called BOHR magneton and is defined as

$$\mu_\mathrm{B} = \frac{e\hbar}{2m_\mathrm{e}} = 927.400915(23) \times 10^{-26}\,\mathrm{JT}^{-1} \quad [8]. \tag{2.27}$$

g_e is the electron's LANDÉ factor. The precise prediction of its value

$$g_\mathrm{e} = -2.0023193043622(15) \quad [8] \tag{2.28}$$

is one of the big successes of quantum electrodynamics (QED).

In case of nuclei, the magnetic moment is expressed in units of nuclear magnetons

$$\mu_N = \frac{e\hbar}{2m_\mathrm{p}} = 5.05078324(13) \times 10^{-27}\,\mathrm{JT}^{-1} \quad [8]. \tag{2.29}$$

The magnetic moment of a nuclei is thus given by

$$\boldsymbol{\mu} = \gamma\boldsymbol{J} = g_J\frac{\mu_N}{\hbar}\boldsymbol{J} \ . \tag{2.30}$$

g_J is characteristic for each nucleus.

2.2 Free Spins in a Magnetic Field

By solving the SCHRÖDINGER equation using the HAMILTONIAN of a spin $\frac{1}{2}$ particle at rest in a magnetic field $\boldsymbol{B} = B_0\hat{\boldsymbol{z}}$ and calculating the expectation value of the spin $\langle\boldsymbol{S}\rangle$ one obtains following values for the individual components:

$$\begin{aligned}
\langle S_x\rangle &= \frac{\hbar}{2}\sin\alpha\cos(\gamma B_0 t) \\
\langle S_y\rangle &= -\frac{\hbar}{2}\sin\alpha\sin(\gamma B_0 t) \\
\langle S_z\rangle &= \frac{\hbar}{2}\cos\alpha
\end{aligned} \tag{2.31}$$

This means that the expectation value precesses about the direction of the field at the LARMOR frequency

$$\omega = \gamma B_0 \ . \tag{2.32}$$

The angle α by which $\langle\boldsymbol{S}\rangle$ is tilted from the direction of the magnetic field depends only on the initial conditions and is constant (figure 2.3).

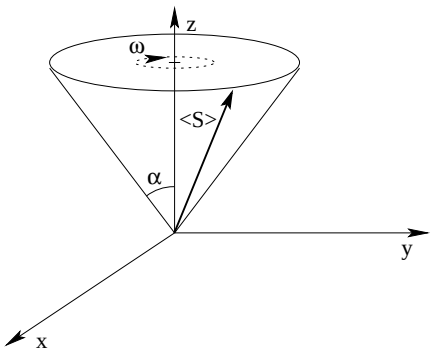


Figure 2.3: Precession of the expectation value $\langle \boldsymbol{S} \rangle$ in a uniform magnetic field.

As stated by EHRENFEST's theorem, the same result should be obtained if the classical equations of motions are solved. An angular moment $\boldsymbol{L}$ on which a torque $\boldsymbol{\tau}$ is acting evolves according to

$$\frac{\mathrm{d}\,\boldsymbol{L}}{\mathrm{d}\,t} = \boldsymbol{\tau} \ . \tag{2.33}$$

Substituting equation 2.5 and equation 2.11[5] into this equation yields

$$\frac{\mathrm{d}\,\boldsymbol{\mu}}{\mathrm{d}\,t} = \boldsymbol{\mu} \times \gamma \boldsymbol{B} = \boldsymbol{\omega} \times \boldsymbol{\mu} \ . \tag{2.34}$$

The solution is a time-dependent magnetic moment of the form

$$\boldsymbol{\mu}(t) = \begin{pmatrix} \mu_{xy} \cos(\omega t) \\ \mu_{xy} \sin(\omega t) \\ \mu_z \end{pmatrix} \tag{2.35}$$

which has exactly the same form as the equation of the expection value (eq. 2.31). This solution also holds for a magnetization vector $\boldsymbol{M}$ which is the sum of a number of moments $\boldsymbol{\mu}_i$

$$\begin{aligned} \boldsymbol{M} &= \sum_i \boldsymbol{\mu}_i \\ \frac{\mathrm{d}\,\boldsymbol{M}}{\mathrm{d}\,t} &= \boldsymbol{\omega} \times \boldsymbol{M} \ . \end{aligned} \tag{2.36}$$

2.3 Spin Ensembles in a Magnetic Field

If an object having a magnetic moment $\boldsymbol{\mu}$ is placed in a magnetic field $\boldsymbol{B}$ not only a torque $\boldsymbol{\tau}$ acts upon it, but it also has a potential energy E_{pot} depending on its orientation.[6] This potential energy is given by

$$E_{\mathrm{pot}} = -\boldsymbol{\mu}\boldsymbol{B} \ . \tag{2.37}$$

[5] Of course, the quantum mechanical value for γ has to be used in equation 2.11.

[6] Since the potential energies are the eigenvalues of the Hamiltonian, this relation was implicitly used in the derivation of the equation for the expectation value in the previous section.

nucleus	"nuclear spin" J	magnetic moment μ [μ_N]	LARMOR frequency $\frac{\nu}{B}$ [MHz/T]	rel. abundance in biol. systems
$^{1}_{1}\mathrm{H}$	$\frac{1}{2}$	2.793	42.58	1
$^{23}_{11}\mathrm{Na}$	$\frac{3}{2}$	2.216	11.27	10^{-3}
$^{31}_{15}\mathrm{P}$	$\frac{1}{2}$	1.131	17.25	1.4×10^{-3}
$^{13}_{6}\mathrm{C}$	$\frac{1}{2}$	0.702	10.71	2.5×10^{-4}
$^{19}_{9}\mathrm{F}$	$\frac{1}{2}$	2.627	40.08	6.3×10^{-5}

Table 2.3: Selected nuclei and their "nuclear spin" J, magnetic moment, LARMOR frequency, as well as their abundance in biological systems relative to $^{1}_{1}\mathrm{H}$ (adapted from [6, 9]).

If the magnetic moment is quantized the potential energy E_pot will also have discrete values only

$$E_m = -\gamma\hbar B m_J \quad \text{with} \quad m_J = -J, -J+1, \ldots, J-1, J \; . \tag{2.38}$$

A spin $\frac{1}{2}$ particle will thus have two energy levels with an energy difference

$$\Delta E = \gamma\hbar B \quad \text{with} \quad B = |\boldsymbol{B}| \, . \tag{2.39}$$

Transitions between the two energy levels are possible by the emission or absorption of photons. The frequency of the photons is given by

$$\Delta E = h\nu = \hbar\omega \; . \tag{2.40}$$

This frequency is called the LARMOR frequency and can also be written as

$$\omega = \gamma B \; . \tag{2.41}$$

For the hydrogen nucleus the LARMOR frequency is $\frac{\nu}{B} = \frac{\gamma}{2\pi} \approx 42.58\,\mathrm{MHz/T}$. Table 2.3 lists a few selected nuclei and their spin J, magnetic moment μ, LARMOR frequency ν and relative abundance in biological systems such as the human body.

If an ensemble of non-interacting spin $\frac{1}{2}$ particles is placed in an external magnetic field $\boldsymbol{B}$ both energy levels ($\gamma\hbar B$ and $-\gamma\hbar B$) will be populated equally.[7] If, however, the ensemble interacts with a reservoir the population distribution in thermal equilibrium is given by the BOLTZMANN statistics

$$\frac{n_{-\frac{1}{2}}}{n_{\frac{1}{2}}} = \mathrm{e}^{-\frac{\Delta E}{k_B T}} = \mathrm{e}^{-\frac{\gamma\hbar B}{k_B T}} \; , \tag{2.42}$$

where $k_B = 1.3807 \times 10^{-23}\,\mathrm{J/K}$ is the BOLTZMANN constant and T the temperature. This means that the state of higher energy will be less populated than

[7] This, of course, is only true if the particles were not polarized before they were placed into the magnetic field.

the state of lower energy. However, in general, $\gamma\hbar B$ will be much smaller than $k_B T$, and thus the ratio $\frac{n_{-\frac{1}{2}}}{n_{\frac{1}{2}}}$ will be close to one.

Using equation 2.42, the population difference can be calculated to be

$$\Delta n = N \tanh\left(\frac{\gamma\hbar B}{2k_B T}\right) \approx \frac{N\gamma\hbar B}{2k_B T} \quad (\gamma\hbar B \ll k_B T) \tag{2.43}$$

At room temperature $T \approx 300\,\mathrm{T}$ and a magnetic field $B = 3\,\mathrm{T}$, this results in the lower energy level having an excess of

$$\Delta n \approx 10^{-5} N \tag{2.44}$$

particles. This population difference leads to a net magnetization of

$$M = \Delta n \cdot \mu \approx \frac{N\gamma\hbar\mu B}{2k_B T} \tag{2.45}$$

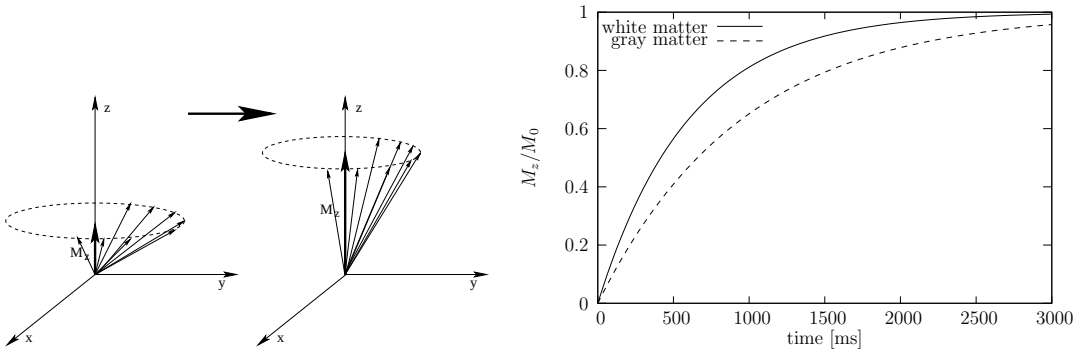
in thermal equilibrium. Due to the large number of nuclei involved in a typical MR experiment (N is in the order of Avogadro's number $N_A = 6.022 \times 10^{23}\,\mathrm{mol}^{-1}$), M is of detectable size.

2.4 Relaxation Mechanisms

In the previous two sections, the magnetization of a spin ensemble in thermal equilibrium and the behavior of a non-interacting magnetic moment with an arbitrary initial condition were discussed. In this section these two aspects shall be fused. The question on how an initial condition that differs from thermal equilibrium can be obtained will be discussed in a later section.

The transition from a given magnetic moment $\boldsymbol{M}$ to the equilibrium moment $\boldsymbol{M}_0$ is called relaxation. The z component of $\boldsymbol{M}_0$ is given by equation 2.45. Since in thermal equilibrium there is no energy difference between a magnetic moment's orientation perpendicular to the magnetic field, each possible orientation is equally probable. The transverse component of the total magnetic moment $\boldsymbol{M}_0$ must thus be 0. Because of this, and for reasons that will become clear soon, it makes sense to split the discussion of relaxation mechanisms into the relaxation of the longitudinal moment M_z and the relaxation of the transverse moment $\boldsymbol{M}_\perp$.[8] The first process is called "spin-lattice relaxation", the second "spin-spin relaxation".

[8] Strictly speaking, this is allowed only if the time constant of the process leading to the recovery of the longitudinal magnetic moment is much longer than the process leading to the decay of the transverse component.



(a) (b)

Figure 2.4: Spin-lattice relaxation: the longitudinal magnetization M_z recovers with time due to spin flip processes. Spin-spin relaxation (which is shown in figure 2.5) takes place at the same time independent of spin-lattice relaxation.

Spin-Lattice Relaxation

Changes in M_z can only occur by a spin's transition from one of its quantized orientations to another. Since both energy and angular momentum have to be conserved such a spin flip requires energy and angular momentum exchange with the surroundings (referred to as lattice) of the nucleus.

Using MAXWELL's theorem of relaxation, the spin-lattice relaxation can be expressed mathematically as a differential equation

$$\frac{\mathrm{d}\, M_z}{\mathrm{d}\, t} = -\frac{M_z - M_0}{T_1} \ . \tag{2.46}$$

The solution of this equation is given by

$$M_z(t) = M_0 \left(1 - \mathrm{e}^{-\frac{t}{T_1}}\right) + M_z(0)\, \mathrm{e}^{-\frac{t}{T_1}} \ , \tag{2.47}$$

where $M_z(0)$ is the initial longitudinal magnetization (i.e. immediately after a RF pulse). The time constant T_1 is referred to as longitudinal or spin-lattice relaxation time (fig. 2.4). In solid insulators, T_1 can be as long as 1000 s, while in conductors it can be in the range of ms.[9] Typical values for T_1 of human tissues are given in table 2.4.

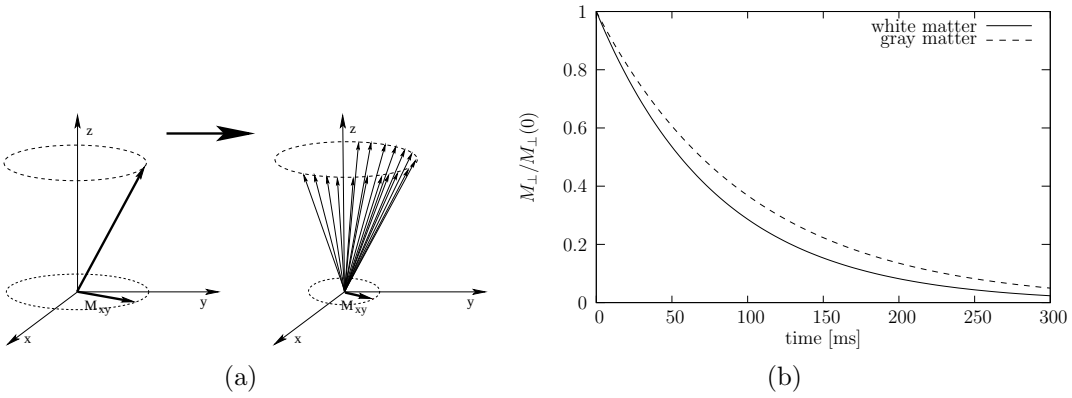
Spin-Spin Relaxation

The decay of the transverse magnetic moment $\boldsymbol{M}_\perp$ (referred to as spin-spin relaxation) occurs independent of the spin-lattice relaxation. Furthermore, it is not a result of energy exchange, but of a mechanism called "dephasing".

[9] In metals, the electrons in the conduction band form an additional energy reservoir. Thus, the energy exchange can occur at a faster rate.

Tissue	T_1 [ms]	T_2 [ms]
white matter	600	80
gray matter	950	100
cerebrospinal fluid	4500	2200
fat	250	60
muscle	900	50
blood	1200	100-200

Table 2.4: Spin-lattice relaxation times (T_1) and spin-spin relaxation times (T_2) of several tissues at a field strength of $B_0 = 1.5\,\mathrm{T}$ and body temperature (adopted from [9]).



(a) (b)

Figure 2.5: Spin-spin relaxation: because spins experience different local field strengths they precess at different frequencies. This leads to dephasing of the spins and a decay of the transverse magnetization $\boldsymbol{M}_\perp$.

Within any sample, not all spins will experience the same magnetic field. The reason can be static inhomogeneities of the magnetic field, or stochastic processes such as local random fluctuations of the magnetic field. As a result not all spins will precess with the same LARMOR frequency. $\boldsymbol{M}_\perp$ will thus be an average over a large number of spin isochromats precessing at different frequencies. In a frame of reference rotating with $\boldsymbol{M}_\perp$ it will appear as if the magnetic moments would smear out over the whole transverse plane (figure 2.5). The averaged magnetic moment will hence approach zero. If only random processes are taken into account the time constant of this process is T_2, and $T_2^\star$ if static inhomogeneities of the magnetic field are considered as well. The time constants of these processes can usually be separated

$$\frac{1}{T_2^\star} = \frac{1}{T_2} + \frac{1}{T_2'} \,. \tag{2.48}$$

The time dependency of the magnitude of the transverse magnetization $\boldsymbol{M}_\perp$ is given by

$$|\boldsymbol{M}_\perp(t)| = |\boldsymbol{M}_\perp(0)|\, \mathrm{e}^{-\frac{t}{T_2^\star}} \,. \tag{2.49}$$

Typical values for T_2 of human tissues are given in table 2.4. The differences in relaxation times T_1 and T_2 between various tissues types give rise to image contrast in anatomical MRI.

BLOCH's Equations

By introducing the relaxation mechanisms into equation 2.34 (or equation 2.36, respectively) one obtains a set of differential equations known as BLOCH's equations:

$$\begin{aligned}
\frac{\mathrm{d}\, M_x}{\mathrm{d}\, t} &= (\boldsymbol{M} \times \gamma \boldsymbol{B})_x - \frac{M_x}{T_2} = \gamma\,(M_y B_z - M_z B_y) - \frac{M_x}{T_2} \\
\frac{\mathrm{d}\, M_y}{\mathrm{d}\, t} &= (\boldsymbol{M} \times \gamma \boldsymbol{B})_y - \frac{M_y}{T_2} = \gamma\,(M_z B_x - M_x B_z) - \frac{M_x}{T_2} \\
\frac{\mathrm{d}\, M_z}{\mathrm{d}\, t} &= (\boldsymbol{M} \times \gamma \boldsymbol{B})_z - \frac{M_z - M_0}{T_1} = \gamma\,(M_x B_y - M_y B_x) - \frac{M_z - M_0}{T_1}
\end{aligned} \tag{2.50}$$

From these equations it is obvious that while spin-lattice relaxation also has an effect on the transverse magnetization $\boldsymbol{M}_\perp$, spin-spin relaxation has no influence on the longitudinal magnetization. In biological systems spin-spin relaxation always occurs at a much faster rate than spin-lattice relaxation ($T_1 > T_2$) and thus the influence of spin-lattice relaxation on the transverse magnetic moment $\boldsymbol{M}_\perp$ can be neglected.

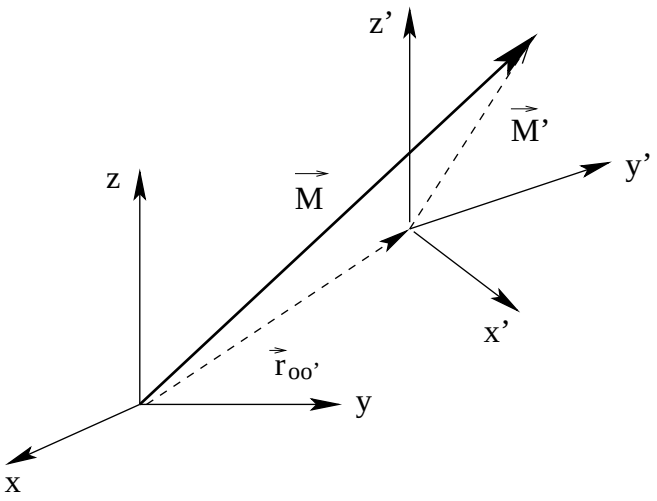


Figure 2.6: General relationship between two frames of reference. The primed coordinates denote the moving frame, x, y and z the resting frame. In case of the description of the magnetic moment as seen from a rotating frame of reference both $\boldsymbol{r}_{OO'}$ and $\frac{\mathrm{d}\,\boldsymbol{r}_{OO'}}{\mathrm{d}\,t}$ vanish (adopted from [10]).

2.5 Magnetic Moments and Radio Frequency Fields

In the previous sections the behavior of a magnetic moment $\boldsymbol{M}$ in a static magnetic field $\boldsymbol{B}$ and the mechanisms leading to the return of a given magnetic moment to thermal equilibrium were discussed. This section will deal with the question on how a deviation of a magnetic moment $\boldsymbol{M}$ from thermal equilibrium can be obtained.

Let's assume that in addition to a static magnetic field $\boldsymbol{B}_0 = B_0\hat{\boldsymbol{z}}$ that causes a magnetic moment to precess at a LARMOR frequency ω_0 a rotating radio frequency (RF) field $\boldsymbol{B}_1$ with the frequency ω is applied:

$$\boldsymbol{B}_1 = \begin{pmatrix} B_1 \cos \omega t \\ B_1 \sin \omega t \\ 0 \end{pmatrix} . \tag{2.51}$$

This leads to a total magnetic field

$$\boldsymbol{B} = \boldsymbol{B}_0 + \boldsymbol{B}_1 = \begin{pmatrix} B_1 \cos \omega t \\ B_1 \sin \omega t \\ B_0 \end{pmatrix} . \tag{2.52}$$

The solution of the resulting equations of motion can be simplified if a transformation into a coordinate system (figure 2.6) rotating with the RF field is

performed [6]:

$$\frac{\mathrm{d}\,\boldsymbol{M}'}{\mathrm{d}\,t} = -\gamma \boldsymbol{B}' \times \boldsymbol{M}' = \omega' \times \boldsymbol{M}' \; . \tag{2.53}$$

Transforming the time derivative to the rotating frame and taking into account that the origins of both frames coincide yields

$$\frac{\mathrm{d}'\,\boldsymbol{M}'}{\mathrm{d}\,t} = \begin{pmatrix} \omega M'_y \\ -\omega M'_x \\ 0 \end{pmatrix} + \begin{pmatrix} \gamma B_0 M'_y \\ -\gamma B_0 M'_x + \gamma B_1 M'_z \\ -\gamma B_1 M'_y \end{pmatrix} = -\gamma \begin{pmatrix} B_1 \\ 0 \\ B_0 + \frac{\omega}{\gamma} \end{pmatrix} \times \begin{pmatrix} M'_x \\ M'_y \\ M'_z \end{pmatrix} \; . \tag{2.54}$$

The last term in eq. 2.54 now contains an expression for an effective field

$$\boldsymbol{B}'' = \begin{pmatrix} B_1 \\ 0 \\ B_0 + \frac{\omega}{\gamma} \end{pmatrix} \; . \tag{2.55}$$

Resonance

It is intuitively obvious that for

$$\omega_0 - \omega = \Delta\omega = 0 \tag{2.56}$$

the effective field $\boldsymbol{B}''$ (eq. 2.55) becomes

$$\boldsymbol{B}'' = \begin{pmatrix} B_1 \\ 0 \\ 0 \end{pmatrix} \; . \tag{2.57}$$

Substituting this effective field into the equation of motions leads to

$$\begin{aligned} \frac{\mathrm{d}'\,M'_x}{\mathrm{d}\,t} &= 0 \\ \frac{\mathrm{d}'\,M'_y}{\mathrm{d}\,t} &= -M'_z \omega_1 \\ \frac{\mathrm{d}'\,M'_z}{\mathrm{d}\,t} &= M'_y \omega_1 \; . \end{aligned} \tag{2.58}$$

The resulting motion in the rotating frame of reference is thus a rotation about the x'-axis. The angle depends on the strength of the RF Field B_1 and its duration t:

$$\phi(t) = \omega_1 t = -\gamma B_1 t \tag{2.59}$$

Thus, by choosing the amplitude and/or duration of B_1 an arbitrary inclination can be obtained. In MR two angles are of special importance:

$$B_1 t = -\frac{\pi}{2\gamma} \quad \rightarrow \quad \phi = \frac{\pi}{2} \tag{2.60}$$

and

$$B_1 t = -\frac{\pi}{\gamma} \quad \rightarrow \quad \phi = \pi \; . \tag{2.61}$$

The first is called a $90\,^\circ$ pulse and the second a $180\,^\circ$ pulse.

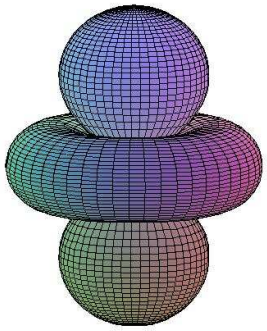


Figure 2.7: Iso-surface of the magnitude of the magnetic field $\boldsymbol{B}_{\text{dipole}}$ of a point dipole. Note that the torus and the handles have opposite signs.

2.6 The NMR Signal

The magnetic field $\boldsymbol{B}_{\text{dipole}}$ that is generated by a (point) dipole $\boldsymbol{M}$ is given by

$$\boldsymbol{B}_{\text{dipole}}(\boldsymbol{r}) = \frac{\mu_0}{4\pi r^3}\left(3\left(\boldsymbol{M}\hat{\boldsymbol{r}}\right)\hat{\boldsymbol{r}} - \boldsymbol{M}\right) + \frac{2\mu_0}{3}\boldsymbol{M}\delta^3(\boldsymbol{r}) \;, \tag{2.62}$$

where $\boldsymbol{r}$ is the distance from the dipole to the point of measurement, $\hat{\boldsymbol{r}}$ the unity vector pointing in the same direction as $\boldsymbol{r}$ and μ_0 the magnetic vacuum permeability (figure 2.7 displays an iso-surface of the magnetic field strength $\boldsymbol{B}_{\text{dipole}}$). Combining this expression with equations 2.35 and 2.36 leads to a time varying magnetic field $\boldsymbol{B}_{\text{dipole}}(\boldsymbol{r}, t)$.

If a loop is placed somewhere in the vicinity of the sample, the time varying field $\boldsymbol{B}_{\text{dipole}}(\boldsymbol{r}, t)$ gives rise to an electromotive force $\mathcal{E}$ in the loop. According to Faraday's law [2] of in induction the electromotive force $\mathcal{E}$ is given by

$$\mathcal{E} = -\frac{\mathrm{d}\,\Phi}{\mathrm{d}\,t} = -\frac{\mathrm{d}}{\mathrm{d}\,t}\int_{\text{coil}} \mathrm{d}\,\boldsymbol{A}\cdot\boldsymbol{B}_{\text{dipole}}(\boldsymbol{r}, t) \;. \tag{2.63}$$

As a result a current will be induced that, according to Lenz's law, will oppose the flux changes. In MR experiments, this current is the measured signal[10]. From this equation it is obvious that only variations of the magnetic field with time contribute to the signal and, thus, the static field can be ignored. Furthermore the contribution of the relaxation processes can be ignored when calculating the differential, since they happen on a time scale that is much slower than the precession. Thus, only the rotation of the transverse magnetization $\boldsymbol{M}_{\perp}$ leads to the NMR signal.

Although, in principle, equation 2.63 is the general form of Faraday's law of induction, there is an implicit assumption in the way it is written: $\boldsymbol{B}_{\text{dipole}}(\boldsymbol{r}, t)$

[10] For reasons that will be explained later, usually a voltage is measured and the current is kept small by using circuits with a high impedance.

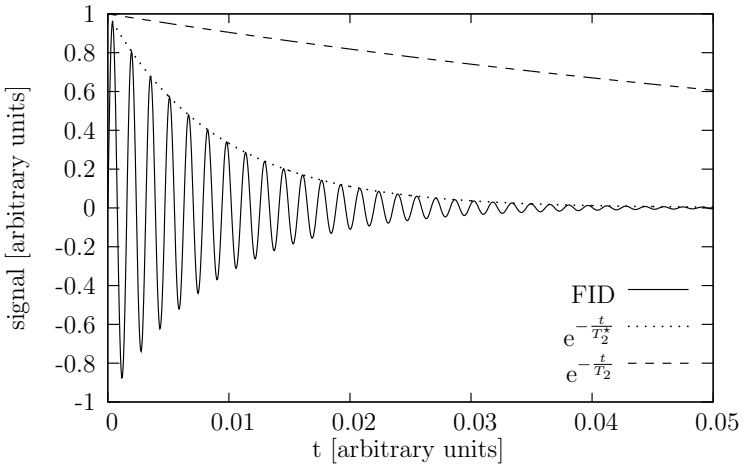


Figure 2.8: Schematic signal generated by a free induction decay (FID). The curves corresponding to T_2 and $T_2^\star$, respectively, are also shown.

was assumed to be a dipole field. This holds only if the sample is very small compared to the coil or at a large distance from the coil. In NMR experiments this is not fulfilled. Nevertheless, this does not pose a substantial problem since one can split up the magnetic dipole moment $\boldsymbol{M}$ into a distribution of magnetic moments $\boldsymbol{\mu}_i$ and compute the contribution of each to the total signal. This can be done by applying the principle of reciprocity.[11] The magnetic field $\boldsymbol{B}_{\text{receive}}(\boldsymbol{r})$ that would be generated by a current in the loop is used as spatial weighting factor called coil sensitivity. The induced electromotive force $\mathcal{E}$ is then given by [9]

$$\mathcal{E} = -\frac{\mathrm{d}}{\mathrm{d}\,t}\Phi(t) = -\frac{\mathrm{d}}{\mathrm{d}\,t}\int_{\text{sample}} \mathrm{d}^3\, r\boldsymbol{\mu}(\boldsymbol{r},t)\cdot\boldsymbol{B}_{\text{receive}}(\boldsymbol{r})\ . \tag{2.64}$$

If we replace $\boldsymbol{\mu}(\boldsymbol{r},t)$ in equation 2.64 with the solution of the equation of motions for a magnetic moment in a magnetic field (eq. 2.35) it becomes apparent that $\mathcal{E}$ oscillates with the LARMOR frequency ω. If relaxation is taken into account as well one obtains a waveform that is known as free induction decay (FID) (fig. 2.8).

List of References

[1] Demtröder, W. *Experimentalphysik 1.* Springer, 4 edition (2006).

[2] Demtröder, W. *Experimentalphysik 2.* Springer, 4 edition (2006).

[3] Griffiths, D. J. *Introduction to Quantum Mechanics.* Prentice Hall (1994).

[11] The principle of reciprocity states that the signal induced in a loop by a localized source is proportional to the magnetic field generated by the loop if a current is applied.

[4] Beiser, A. *Concepts of Modern Physics*. McGraw-Hill, 6 edition (2003).

[5] Demtröder, W. *Experimentalphysik 4*. Springer, 2 edition (2005).

[6] Badurek, G. *Biological and Medical Applications of Nuclear Physics 2*. Lecture Notes.

[7] Yao, W. M., *et al. Review of Particle Physics*. Journal of Physics G, **33** (2006).

[8] Committee on Data for Science and Technology. *2006 CODATA recommended values*. http://physics.nist.gov/cuu/Constants/.

[9] Haacke, E. M., Brown, R. W., Thompson, M. R., and Venkatesan, R. *Magnetic Resonance Imaging – Physical Principles and Sequence Design*. Wiley-Liss (1999).

[10] Rauscher, A. *Phase information in magnetic resonance imaging*. Ph.D. thesis, Technische Universität Wien (2005).

3
Magnetic Resonance Imaging

The physical principles described in the previous chapter are exploited in an imaging modality known as magnetic resonance imaging (MRI) [1]. MRI probably is the most important application of nuclear magnetic resonance (NMR) today. This chapter starts with a simple derivation of the basic concepts that allow imaging using the signal created by the precession of magnetic moments. During this discussion most of the physics derived in the previous chapter is neglected. By adding excitation and the relaxation mechanisms to the concepts described in the beginning contrast mechanisms and two basic experimental protocols for imaging are derived. A discussion of the properties of experimental data would be incomplete if the measurement process itself was left out. The data acquisition has some very fundamental consequences on the images obtained with MRI, it is thus discussed briefly. Finally some sources of image artifacts are mentioned to illustrate the complexity of MRI and MRI data.

3.1 Basic Concepts of MRI

MRI is based on the resonance frequency's dependency on the external magnetic field $\boldsymbol{B}_0$ (eq. 2.41). Thus, if one manages to generate a magnetic field $\boldsymbol{B}_0$ that is a function of space, the LARMOR frequencies will also be a function of space, and, therefore, space will be encoded by frequencies. The basic principles of this encoding will be derived starting with a one-dimensional problem. The two and three dimensional cases will be discussed afterwards. For lucidity, several simplifications will be introduced: first of all signal decay will be neglected, secondly, in each situation the problem will be treated as truly one dimensional, or two dimensional, respectively. The fact that space is three dimensional and,

thus, signals originating from all dimensions contribute to the total signal will be ignored. Furthermore it is assumed that the sensitivity to signals coming from the whole measured volume is uniform.

The One-Dimensional Problem

Let's consider a one dimensional distribution of magnetic moments $\boldsymbol{\mu}(x)$ in a magnetic field $\boldsymbol{B}_0$ (fig. 3.1). $\boldsymbol{B}_0$ shall be $\boldsymbol{B}_0 = B_0\hat{\boldsymbol{z}}$, the precession of all moments shall be in phase and their longitudinal component shall be zero. With this assumptions, $\boldsymbol{\mu}(x)$ can be written as

$$\boldsymbol{\mu}(x,t) = \mu(x) \begin{pmatrix} \cos\omega t \\ \sin\omega t \\ 0 \end{pmatrix} , \tag{3.1}$$

where $\omega = \gamma B_0$. Substituting this expression into equation 2.64 under the assumption that $\boldsymbol{B}_{\text{receive}} = B_{\text{receive}_x}\hat{\boldsymbol{x}}$ will yield an electromotive force $\mathcal{E}$ oscillating with ω. $\mathcal{E}$ is proportional to the integral of $\mu(x)$:

$$\mathcal{E}(t) = -\frac{\mathrm{d}}{\mathrm{d}\,t}\int \mathrm{d}x B_{\text{receive}_x}\mu(x)\cos\omega t \;\propto\; \sin\omega t \int \mathrm{d}x\mu(x) \tag{3.2}$$

The measured signal $s(t)$ is proportional to $\mathcal{E}(t)$.

Let's assume that it is possible to modulate the z component of the magnetic field $\boldsymbol{B}_0$ without changing any of the other components.[1] The variation shall be a linear function of x so that

$$B_z(x) = B_0 + G_x x \; . \tag{3.3}$$

Since applying the nabla operator to this function yields

$$\nabla B_z = \begin{pmatrix} \frac{\mathrm{d}\,B_z}{\mathrm{d}\,x} \\ \frac{\mathrm{d}\,B_z}{\mathrm{d}\,y} \\ \frac{\mathrm{d}\,B_z}{\mathrm{d}\,z} \end{pmatrix} = \begin{pmatrix} G_x \\ 0 \\ 0 \end{pmatrix} , \tag{3.4}$$

G_x is called x gradient.

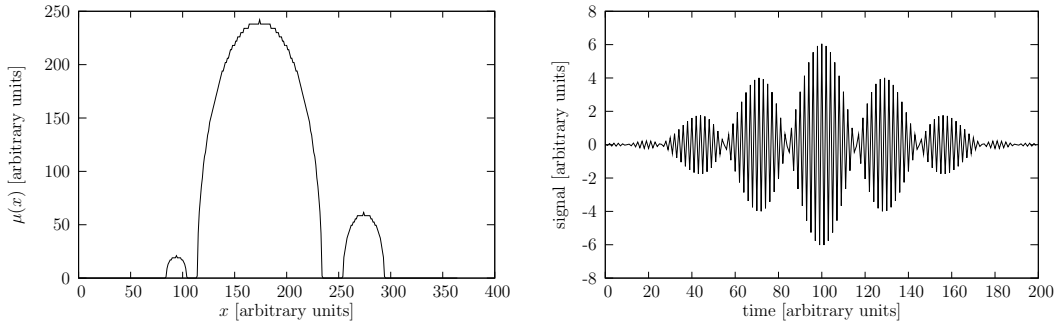
Because of equation 2.32 and 3.3 the Larmor frequency is now a function of x

$$\omega(x) = \gamma B_z(x) = \gamma\left(B_0 + G_x x\right) \; . \tag{3.5}$$

The signal given by equation 3.2 is thus modified to

$$\begin{aligned} s(t) &\propto \frac{\mathrm{d}}{\mathrm{d}\,t}\int \mathrm{d}x\mu(x)\cos\left(\omega_0 + \gamma G_x x\right)t \\ &\propto \int \mathrm{d}x\mu(x)\left(\omega_0 + \gamma G_x x\right)\sin\left(\omega_0 + \gamma G_x x\right)t \; . \end{aligned} \tag{3.6}$$

[1] Strictly speaking it is not possible to modify a single component of the magnetic field $\boldsymbol{B}_0$ without altering any of the other components because of Maxwell's equations [2]. Nevertheless, if the modification is small a change of the axis of precession can be neglected.



(a) (b)

Figure 3.1: An one dimensional distribution of magnetic moments $\mu(x)$ (a) and an example of a signal that results from those moments precessing in a magnetic field to which a linear gradient is superimposed (b).

Using the coordinate transform

$$\begin{aligned} x' &= \omega_0 + \gamma G_x x \\ \mathrm{d}\, x' &= \gamma G_x \,\mathrm{d}\, x \end{aligned} \tag{3.7}$$

and taking into account that, in general, $\gamma G_x x$ is very small compared to ω_0 equation 3.6 can be written as

$$s(t) \propto \int \mathrm{d}x' \mu(\frac{x' - \omega_0}{\gamma G_x}) \sin x' t \propto \int \mathrm{d}x' \mu'(x') \sin x' t \; . \tag{3.8}$$

From this expression it is obvious that the measured signal (fig. 3.1(b)) is nothing but the sine transform of the magnetic moments' distribution (fig. 3.1(a))! Thus, the distribution can be obtained from the measured signal by using the inverse transform.

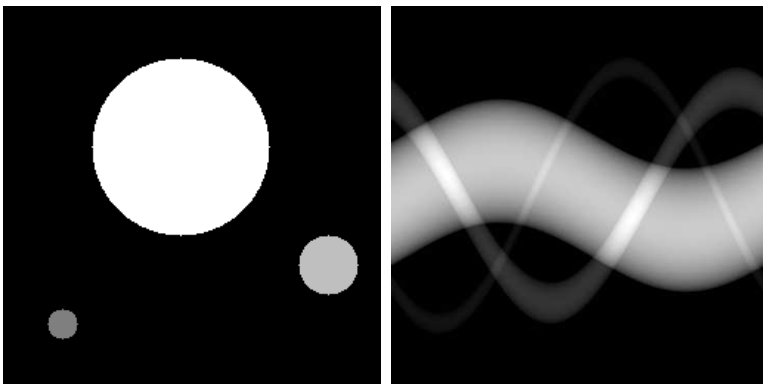
The Two-Dimensional Problem

The problem of encoding a second dimension can be solved using the same assumptions as for the one dimensional problem. The only exception shall be that the distribution of the magnetic moments is a function of not only x but also of y. This leads to

$$\boldsymbol{\mu}(x, y, t) = \mu(x, y) \begin{pmatrix} \cos \omega t \\ \sin \omega t \\ 0 \end{pmatrix} \; . \tag{3.9}$$

If no field gradient is applied the signal can be written as

$$s(t) \propto \sin \omega t \iint \mathrm{d}x \,\mathrm{d}y \, \mu(x, y) \; . \tag{3.10}$$



(a) (b)

Figure 3.2: A two-dimensional distribution of magnetic moments $\mu(x, y)$ (a) and profiles acquired from 0 ° to 360 ° in steps of 1 ° plotted as a sinogram (b).

Just as before, it is possible to apply a gradient. Once again, this gradient shall be along the x axis. Substituting the one dimensional distribution in equation 3.8 by a two dimensional distribution and using equation 3.11 leads to

$$s(t) \propto \iint \mathrm{d}x' \, \mathrm{d}y \, \mu(\frac{x' - \omega_0}{\gamma G_x}, y) \sin x't \propto \int \mathrm{d}x' \sin x't \int \mathrm{d}y \, \mu'(x', y) \; . \quad (3.11)$$

It is obvious that only a single direction can be encoded by this approach.

The Projection Reconstruction Method

So far it has been assumed that the field gradient is applied along the x direction and thus encodes along the x axis. Of course, instead of applying a gradient G_x, one could use a gradient G_y along the y axis. This would yield an encoding in the y direction. It is trivial to show that using linear combinations of G_x and G_y allows for encoding in arbitrary directions. The transform of the signal yields a profile of the density $\mu(x, y)$ perpendicular to the encoding direction. Such a profile is equivalent to an absorption profile obtained with X-ray computer aided tomography (CAT). The series of steps required for imaging can thus be summarized as:

1. measure a set of density profiles of $\mu(x, y)$ by varying the angle of the encoding direction step by step (figure 3.2b displays a set of profiles of the distribution shown in figure 3.2a as a sinogram),

2. reconstruct the image by back-projection.

Although, this method was the one proposed for MRI originally [1], today the Fourier reconstruction method is used in most MRI applications.

The Fourier Reconstruction Method

One of the many assumptions that were made previously was that the magnetic moments had a no phase shift with respect to $t = 0$. This was done because the mathematical treatment is much easier, and therefore, the result is more intuitive. In general, this will not be fulfilled. In the most general case (but still using a many of the simplifications introduced in the treatment of the one dimensional problem) equation 3.2 can be written as

$$s(t) \propto \iint \mathrm{d}x\,\mathrm{d}y\;\mu(x,y)\sin\left(\omega(x,y)t + \phi(x,y)\right)\ . \tag{3.12}$$

To solve the one dimensional case, it was assumed that a gradient G_x is applied along the x direction. Thus, the previous equation can be rewritten into the form known from before:

$$s(t) \propto \iint \mathrm{d}x\,\mathrm{d}y\;\mu(x,y)\sin\left(\omega_0 t + \gamma G_x x t + \phi(x,y)\right)\ . \tag{3.13}$$

No statement about the history of the magnetic moments has been made so far. It was assumed implicitly that $t \geq 0$, that the signal is measured continuously and that the gradient G_x is constant with respect to time. It shall be now assumed that all magnetic moments were forced to a coherent precession by an RF pulse at some point $t < 0$.[2] Whatever happened between this excitation and the start of the measurement is of importance now, since it might have changed the phase $\phi(x,y)$. If, for example, a gradient $\boldsymbol{G}$ of the form

$$\boldsymbol{G} = \begin{pmatrix} 0 \\ G_y \\ 0 \end{pmatrix} \tag{3.14}$$

had been applied for a period Δt, a magnetic moment at (x,y) would have acquired a phase shift of

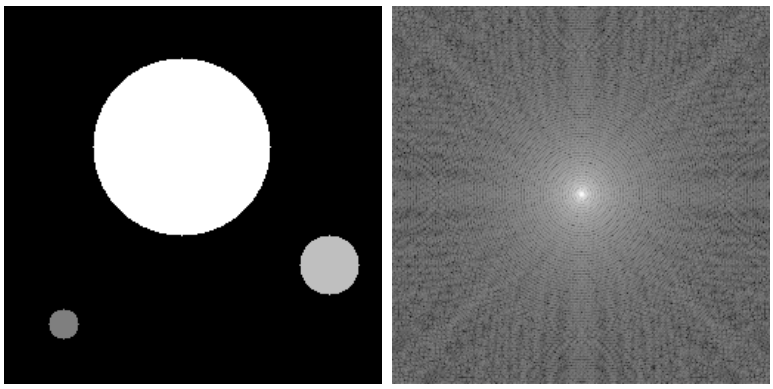
$$\phi(x,y) = \Delta\omega\Delta t = \gamma G_y y \Delta t \tag{3.15}$$

compared to a magnetic moment at $(x,0)$. The total signal would thus be

$$s(t) \propto \iint \mathrm{d}x\,\mathrm{d}y\;\mu(x,y)\sin\left(\omega_0 t + \gamma G_x x t + \gamma G_y \Delta t y\right)\ . \tag{3.16}$$

Since $\gamma G_y \Delta t$ is constant during a single experiment it does not change the fact that only the x direction is encoded by this measurement. Nevertheless, assuming that the experiment is repeated with different values of $\gamma G_y \Delta t$ the

[2] The excitation is assumed to happen at $t < 0$ because the signal is measured from $t = 0$ onwards. Thus, a RF pulse at $t = 0$ would result in the situation described previously.



(a) (b)

Figure 3.3: A two-dimensional distribution of magnetic moments $\mu(x, y)$ (a) and the corresponding k-space (the logarithm of the magnitude is shown) (b). Each line corresponds to the signal acquired after a certain phase encoding gradient G_y was applied.

signal $s(t)$ can be written as a function not only of the time t, but also of $\gamma G_y \Delta t$. Using $k_y = \gamma G_y \Delta t$ simplifies the notation:

$$s(t, k_y) \propto \iint \mathrm{d}x\, \mathrm{d}y\, \mu(x, y) \sin\left(\omega_0 t + \gamma G_x x t + k_y y\right) \tag{3.17}$$

It can be shown that the density $\mu(x, y)$ can be calculated from a set of signals equally spaced along k_y by a two dimensional FOURIER transform (fig. 3.3).

The procedure for two dimensional imaging using the FOURIER reconstruction method can be summarized as:

- acquire a set of signals. During each acquisition the G_x gradient is applied. This results in an encoding along the x direction.
- before each acquisition a gradient G_y is applied for a period Δt. The strength of the gradient varies from acquisition to acquisition. The variation of the G_y gradient provides for encoding in the y direction.
- perform a two dimensional FOURIER transform on the full data-set. This transform yields the density $\mu(x, y)$.

The gradient that is applied while the signal is acquired (in the previous treatment this was G_x) is often called “read-out gradient”. The gradient that is used to encode the other direction (G_y) is often named “phase encoding gradient”. The encoding directions are thus referred to as “read-out direction” and “phase-encoding direction”, respectively.

k-Space

To achieve a more consistent notation another coordinate transform shall be introduced[3]:

$$\gamma G_x t = k_x \quad \Leftrightarrow \quad t = \frac{k_x}{\gamma G_x} \tag{3.18}$$

The signal equation thus becomes

$$s(k_x, k_y) \propto \iint \mathrm{d}x\, \mathrm{d}y\, \mu(x, y) \sin\left(k_x \left(\frac{\omega_0}{\gamma G_x} + x\right) + k_y y\right) \tag{3.19}$$

or

$$s(k_x, k_y) \propto \iint \mathrm{d}x'\, \mathrm{d}y\, \mu'(x', y) \sin\left(k_x x' + k_y y\right) \ , \tag{3.20}$$

with $x' = \frac{\omega_0}{\gamma G_x} + x$ and $\mu'(x', y) = \mu(x' - \frac{\omega_0}{\gamma G_x}, y)$. k_x and k_y span the so called k-space [3]. The goal of an MRI acquisition is to sample k-space in a way that k-space is covered completely. Equation 3.20 allows a description that is basically independent of how the sampling and gradient application is performed. For example, the sampling scheme for the projection reconstruction method can also be expressed in terms of k_x and k_y. It thus can be shown that both methods are equivalent.

The Three-Dimensional Problem

There are two methods for performing imaging of a volume. The first one is imaging the volume slice by slice. For this it is necessary that only magnetic moments within a certain slice are excited by an RF pulse. This can be achieved by applying a gradient along the z direction during the duration of the pulse. Thus, the frequency of the pulse will only match the LARMOR frequency of magnetic moments at a certain z position. Only these magnetic moments will be turned towards the x–y plane by the desired angle. After such an excitation both methods for two dimensional imaging, projection and FOURIER imaging, can be applied.

Truly three dimensional imaging can be achieved by extending the principle of two dimensional FOURIER imaging to three dimensions. This is done by encoding along the z axis in the same manner as along the y axis. Thus, for each y encoding step a number of z encoding steps are performed. In practice, both y and z gradients can be applied at the same time.

3.2 Pulse Sequences

A number of assumptions that are violated in nature were made for the derivation of the fundamental concepts that enable imaging by NMR. One, for ex-

[3] For reasons of lucidity the definition of k_x differs from the one usually found in the literature by a factor of 2π.

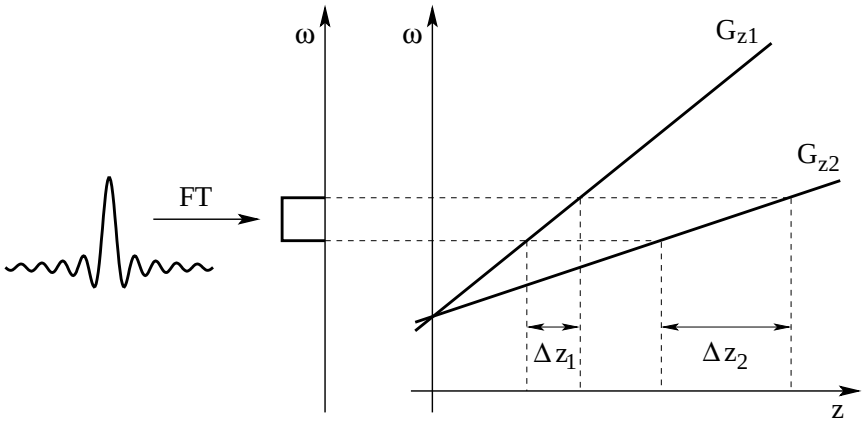


Figure 3.4: Illustration on the effects of different slice selection gradient strengths. A stronger slice gradient (G_{z1}) yields a thiner slice (adopted from [4])

ample, was to exclude both relaxation and excitation. This made it possible to treat the signal as if it was infinite in time. Nevertheless it was mentioned already that the acquistion of an magnetic resonance image requires repeated excitation, encoding and read-out steps. Such a sequence is called a pulse sequence. Within this section a more realistic view of MRI shall be given and the two most important basic sequence types shall be discussed.

RF Pulses and Slice Selection

In section 2.5, the properties a radio frequency must have to change the orientation of a magnetic moment by a desired amount were derived. It was shown that the frequency has to match the LARMOR frequency. Nevertheless, the fact that a delta function like frequency spectrum corresponds to an infinite sine wave, and thus to a pulse of infinite length, was neglected. The effects of limiting the pulse length will be described in the next paragraphs.

Furthermore, it was shown in the previous section how magnetic field gradients can be used for encoding space by magnetic field gradients for imaging. For most of the derivation it was neglected that any sample stretches out in all three dimensions. Only, in the last paragraphs, it was mentioned that by applying a gradient along the third dimension while the sample is excited by an RF pulse, images of single slices can be obtained. In the next paragraphs the properties of such RF pulses and the corresponding gradients shall be discussed.

In most applications, one is interested to excite all magnetic moments within a slab of finite thickness uniformly. The thickness might vary depending on whether a two dimensional or a three dimensional sequence is performed, on how much signal can be expected from a certain volume, or on how much the volume might be affected from sources of signal loss (e.g. due to field inhomogeneities).

The relationship between location and resonance frequency in a situation

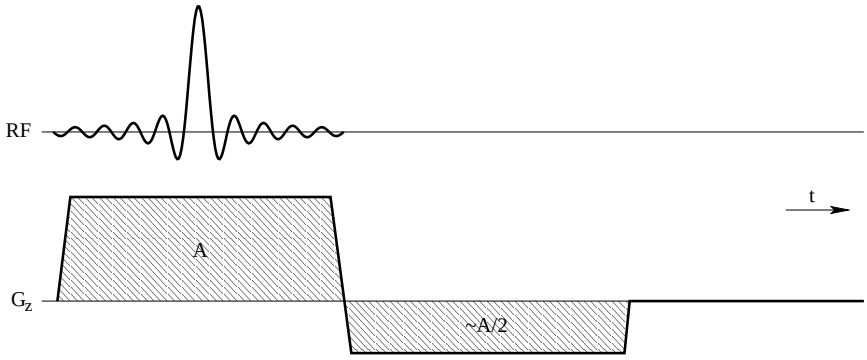


Figure 3.5: A slice selective truncated RF pulse and a slice selection gradient with an area A. The slice selection gradient is followed by a rephasing gradient with an area $\approx \frac{A}{2}$.

where a linear gradient is applied holds not only for precessing magnetic moments but also during excitation. It is given by equation 3.5. Thus, an RF pulse that is supposed to excite magnetic moments at a certain location has to have a frequency that fulfills this equation. According to equation 2.59, the angle by which a magnetic moment is turned into the x-y plane depends on the magnitude and duration of the RF pulse. From these two equations it is obvious that the pulse's frequency spectrum has to resemble a boxcar function with frequencies ranging from $\omega_0 - \gamma G_z \frac{\Delta z}{2}$ to $\omega_0 + \gamma G_z \frac{\Delta z}{2}$ to excite all magnetic moments within a slab of finite thickness. Since the FOURIER transform of a rectangle centered around zero in frequency space is a sinc function in time domain, the envelope of the pulse has to be a sinc function. Figure 3.4 displays the relationship between the thickness of the excited slice and the strength of the slice selection gradient for a given pulse.

Unfortunately, also the sinc function spans the whole time domain. But unlike, for example, the sine function it approaches zero asymptotically. Thus, in principle, it can be truncated. Nevertheless, a truncation causes imperfection (especially ringing) that give rise to an inhomogeneous excitation profile. The shorter the pulse will be the stronger these artifacts will get. In practice, the undesired effects of truncation can be mitigated if other windowing functions than the boxcar function which corresponds to a simple truncation are used.

Another issue is that the slice selection gradient will cause magnetic moments to precess at different frequencies and thus to build up a phase difference with respect to the magnetic moments at the center of the slice:

$$\phi(z, t_p) = \mathrm{e}^{i\gamma G_z (z - z_o) \frac{t_p}{2}} \,, \tag{3.21}$$

where z_0 is the center of the slice and t_p the duration of the slice selection gradient. If this phase shift is not compensated for signal loss occurs. A compensation can be achieved by applying the slice selection gradient with opposite

sign and approximately half the area after excitation.[4] Figure 3.5 shows both a slice selective RF pulse and the scheme for switching the slice selection gradient to compensate for dephasing.

Relaxation

The basic mechanisms leading to a decay of the transverse magnetic moment and the build up of the equilibrium longitudinal magnetic moment were described in section 2.4. This section will deal with the consequences of various relaxation mechanisms for MRI. Although these processes are exploited to generate different contrasts (i.e. proton density, T_1 weighting, T_2 weighting) by changing the acquisition parameters, the contrasts themselves shall not be discussed.

It was pointed out before that the time constant T_1 governing the recovery of the magnetic moment to thermal equilibrium is much longer than the time constant T_2 describing the decay of the transverse component of the magnetic moment. Thus, the two relaxation mechanisms can be treated independently. T_1 relaxation has following importance for imaging: the size of the transverse magnetic moment after a pulse depends not only on the flip angle, but also on the magnitude of the longitudinal magnetic before the pulse was applied. Thus, if subsequent pulses are used to sample k-space the magnitude of the moment that can be exploited by a pulse depends on how much the longitudinal moment has recovered from the previous pulse because of T_1 relaxation. The time between excitations is referred to as repetition time T_R. If T_R is long T_1 differences between tissues will not affect the image contrast. If T_R is short T_1 differences will be a contrast mechanism.

The relaxation of the transverse magnetic moment $\boldsymbol{M}_\perp$, on the other hand, is a result of a multitude of causes. Dephaseing, in principle, is caused by a variation of precession frequencies within a certain volume or by movement of magnetic moments in an inhomogeneous field. Differing precession frequencies can be either a result of time constant field inhomogeneities or by time dependent field variations. It should be therefore obvious that a discussion of the decay of the transverse magnetic moment $\boldsymbol{M}_\perp$ is rather complex.

Although the intrinsic processes leading to T_2 decay are spin-spin interactions which are random processes, for reasons of lucidity, the effects of time constant field inhomogeneities are described first. This is followed by a discussion of time varying field inhomogeneities and of magnetic moments moving in an inhomogeneous field.

[4] The area under the refocusing gradient is exactly half the area of the slice selection gradient only under the assumption that spins are flipped instantaneously and that no spin diffusion or motion (flow) occurs [5].

Static Field Inhomogeneities

Ideally, the magnetic field within a sample is homogeneous and of the same strength as the applied field. This can be fulfilled only if the sample has the same magnetic properties as vacuum, which is almost never the case. Materials alter the magnetic field depending on their magnetic susceptibility χ. The magnetic susceptibility χ is a material property that describes the relation between a material's magnetization and an applied field. The magnetic flux within the material is given by

$$\boldsymbol{B} = \mu_0 \left(1 + \chi\right) \boldsymbol{H} \ , \tag{3.22}$$

where $\mu_0 = 4\pi \times 10^{-7}\,\mathrm{VsA^{-1}m^{-1}}$ is the vacuum's magnetic permeability. Depending on their magnetic susceptibility χ, materials are classified as being diamagnetic ($\chi \leq 0$), paramagnetic ($\chi \geq 0$) or ferromagnetic (in this case the magnetic susceptibility is not a scalar). Most tissues in the human body are diamagnetic.

The field within an object placed into an external field depends not only on the sample's susceptibility, but also on its geometry. The field outside of an object is modified as well to satisfy the MAXWELL equations at the boundaries. Within biological samples, many areas with differing susceptibilities exist. Thus the field within such a sample will never be homogeneous. As a result the local LARMOR frequencies deviate from the expected values. This causes the total transverse magnetic moment to decay due to dephasing. The time constant for this process is named T_2'.

The field variations caused by susceptibility differences are constant in time (at least within the time frame of the experiment). This means that the frequency difference $\Delta\omega$ between a magnetic moment at $\boldsymbol{r}$ and one at $\boldsymbol{r'}$ stays constant. If it was possible to change the direction of precession without altering the absolute value of the precession frequency there would be a point in time at which the moments have zero phase difference again and thus the total magnetic moment would regain its maximum value. This rephasing of magnetic moments is called an echo. It is obvious that the echo would occur after exactly the same time span after sign reversal that the spins had for dephasing. Although, reversing the sign of precession is impossible, methods that achieve a similar behavior will be discussed in the next sections.

Time Varying Field Inhomogeneities

The magnetic field experienced by a spin depends not only on the applied field and macroscopic material properties, but also on the microscopic environment. Or in other words, on the atoms and molecules in its vicinity. Since these are subject to thermal motions such as vibration or BROWNIAN motion, which are stochastic processes, the field that the spin is exposed to fluctuates randomly. The phase shift ϕ a spin accumulates because of the random changes of the magnetic field resembles a random walk [2]. The expectation value of ϕ^2 in-

creases with time. This means that the distribution of phases of an ensemble will spread out and thus the total magnetic moment will decrease. This relaxation is intrinsic to the sample. It is the primary mechanism for T_2 relaxation.

Magnetic Moments Moving in an Inhomogeneous Field

Let's consider a homogeneous sample in which some sort of transport process, for example diffusion, takes place. The effect of this transport process on signal decay can be explained by a simple classical model. For this model we assume a homogeneous sample placed in an inhomogeneous magnetic field.

Depending on its location, a proton contributes to the magnetic moment of a certain isochromat. If this proton moves to a different location, it will contribute to the isochromat that corresponds to the magnetic field at the new location. Since the proton was precessing at a different frequency before, it will have a phase difference with respect to the magnetic moment of the isochromat. If a large number of protons are moving into and out of the isochromat, the total number of protons contributing to the isochromat will stay constant (assuming a zero net flux). Nevertheless, the magnetic moment of the isochromat will change. In the most general case, the phase and the magnitude of the magnetization vector will change. However, how those two are affected depends on the transport process.

One can distinguish between two transport processes: diffusion and flow. The first one, diffusion, is a stochastic process. Protons will move in and out of a location randomly. Each proton will have a phase depending on its history, which can be described by a random walk. As a result the expectation value of the isochromat's phase will stay constant over time, but the magnitude of its magnetization vector will decrease. As in the case of random field fluctuations, this loss cannot be compensated for. Nevertheless, the impact of diffusion depends on the inhomogeneity of the applied field. This is exploited by a method called diffusion weighted imaging.

The second process, flow, leads not only to signal loss, but also to a phase shift. This can be illustrated by considering the most extreme scenario in which all protons move with the same velocity in the same direction. In this case the movement would not lead to a signal loss, but only to a phase shift.

All stochastic processes, diffusion and random fluctuations of the local magnetic field, are subsumed under a single time constant - T_2. The resulting decay of the transverse magnetic moment is irreversible. The time constant for the effective relaxation of the transverse magnetic moment includes the relaxation caused by static field inhomogeneities and is called $T_2^\star$ and is given by

$$\frac{1}{T_2^\star} = \frac{1}{T_2} + \frac{1}{T_2'} \,. \tag{3.23}$$

The effect caused by $T_2^\star$ relaxation is a signal that decays quickly after the excitation pulse.

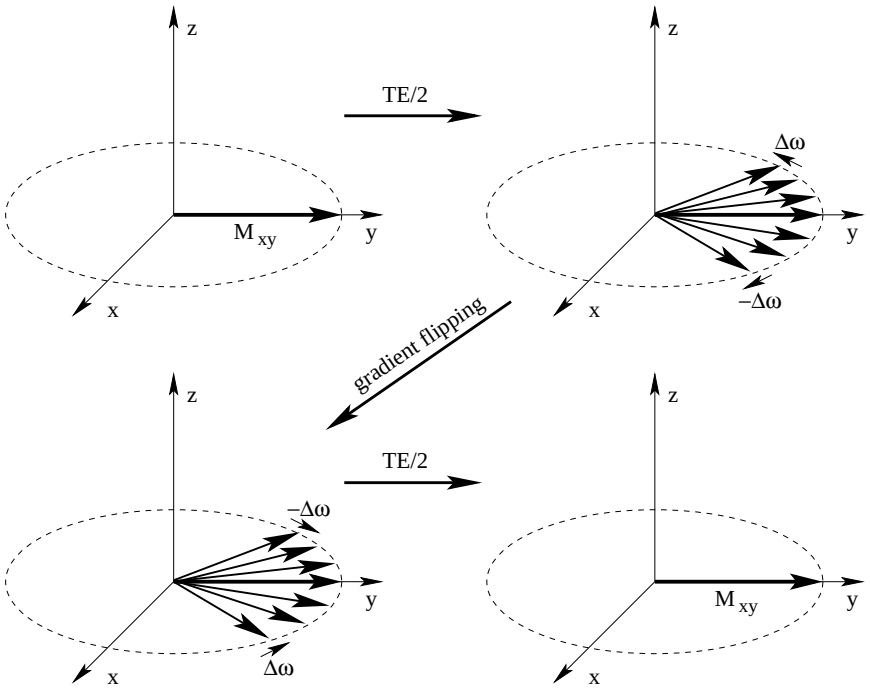


Figure 3.6: Schematic illustration of the principle of a gradient recalled echo as seen from the rotating frame of reference. After forcing the magnetic moments to precess within the x-y plane by an RF pulse the moments will dephase due to an inhomogeneous field caused by the so called (read)dephasing gradient (top row). Flipping the sign of the gradient will reverse this effect. Spins that precessed faster before, will be slower and vice versus. This results in a rephasing of the magnetic moments (bottom row).

Gradient-Echo Imaging

Since the decay of the signal starts immediately after excitation and occurs rapidly there is little time for complicated gradient switching schemes as required by imaging. It was mentioned before that the fraction of the transverse magnetic moment that is lost due to T_2' decay could, at least theoretically, be recovered if it was possible to reverse the direction of rotation. Although, in practice, it is not possible to change the direction of rotation in the laboratory frame, one can modify the velocity of rotation. In the rotating frame this amounts to reversing the motion of magnetic moments relative to each other (fig. 3.6). This can be achieved by reversing the sign of the gradient. This kind of sequence is called "gradient-echo sequence"(GE) or "gradient recalled echo sequence" (GRE).

The echo is generated when the integral of the gradient after gradient reversal reaches the same size as before (fig. 3.7). Dephasing caused by field inhomogeneities (i.e. T_2' decay) cannot be compensated using this technique. Thus, the signal is not subject to pure T_2 decay and is sensitive to field inhomogeneities

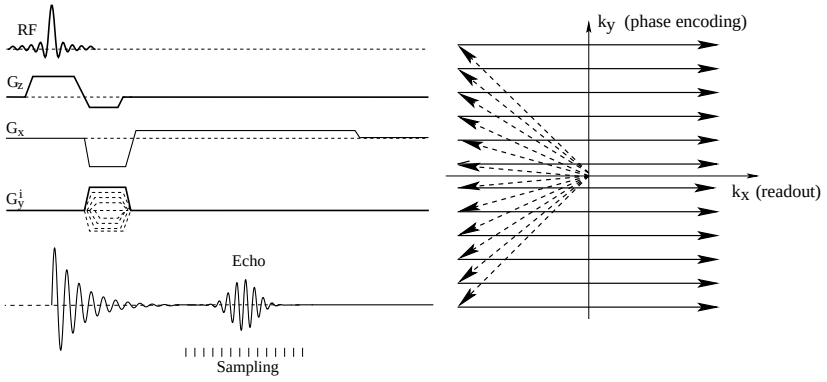


Figure 3.7: Schematic pulse diagram of a gradient-echo sequence (a) and the corresponding k-space trajectory (b). The sequence is repeated for each phase encoding step. The different strengths of the phase encoding gradient for each phase encoding step are indicated by the dashed lines.

and susceptibility differences.

Spin-Echo Imaging

The sensitivity towards field inhomogeneities and susceptibility differences can be avoided by using a so-called spin echo sequence. Figure 3.8 displays the underlying principle as seen from the rotating frame of reference. After the initial pulse the magnetic moments dephase due to their different relative angular velocities. After a while a 180 ° (or π) pulse is applied. In the rotating frame of reference this pulse corresponds to a reflection with respect to a line passing through the origin in the x-y plane (e.g. the x axis). As a result magnetic moments with a small deviation of their angular velocity from the angular velocity of the transverse magnetic moment will be put ahead of those with a larger difference. Thus the faster magnetic moments will "catch up" and an echo will be generated. A precondition is, of course, that the angular velocity of each magnetic moment stays constant. This condition can be fulfilled exactly only as long as the magnetic moments are just subject to static field inhomogeneities. Since in practice stochastic fluctuations of the magnetic field always are present the maximum amplitude of the echo is affected by T_2 decay (but not by T_2' relaxation).

A schematic pulse diagram and the corresponding k-space trajectories are shown in figure 3.9. The sequence starts with a 90 ° pulse that rotates the relaxed magnetic moment into the x-y plane. During the pulse, the slice selection gradient G_z is applied. After the pulse, a refocusing gradient is employed along the z-axis. Phase encoding is performed in the y direction. An additional dephasing gradient is used along the x-axis. The purpose of this gradient is

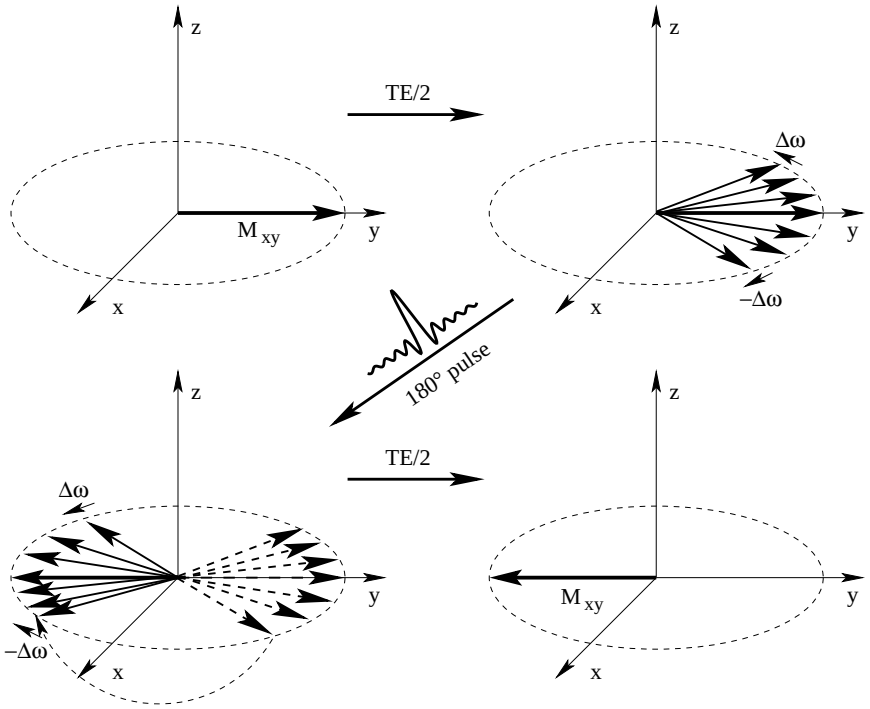
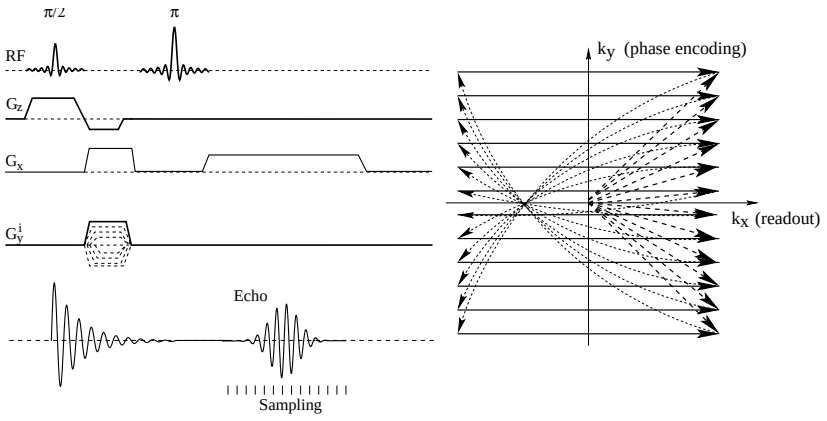


Figure 3.8: Schematic illustration of the principle of a spin echo as seen from the rotating frame of reference. After forcing the magnetic moments to precess within the x-y plane by a RF pulse the moments will dephase due to field inhomogeneities (top row). A 180 ° pulse sets the slow magnetic moments in front of the faster moments. Thus the faster moments will "catch up" and an echo will be generated (bottom row). Note that T_2 decay has been neglected in this illustration.



(a) (b)

Figure 3.9: Schematic pulse diagram of a spin-echo sequence (a) and the corresponding k-space trajectory (b). The sequence is repeated for each phase encoding step. The different strengths of the phase encoding gradient for each phase encoding step are indicated by the dashed lines.

to ensure that the echo is centered in k-space. Although the gradient causes a movement in positive k_x direction, the refocusing pulse reflects the spin system back to negative k_x. Thus, k-space will be sampled from $-\frac{\Delta k_x}{2}$ to $\frac{\Delta k_x}{2}$ during readout. Note that also the sign of k_y is flipped by the refocusing gradient.

3.3 Data Acqusition

Up to now, the signal has been implicitly treated as a infinite continuous function. However, data is not only acquired during a finite time window, but also sampled in discrete time steps. This has some important consequences for imaging.

Perfect time discrete sampling can be described by multiplying the signal $s(t)$ with a sample function $w(t)$. The sampling function is defined as

$$w(t) = \Delta t \sum_{m=-\infty}^{\infty} \delta(t - m\Delta t) \text{ [6]}, \tag{3.24}$$

where $\delta(t)$ is the KRONECKER delta function. The sampled signal $\hat{s}(t)$ can thus be written as

$$\hat{s}(t) = s(t)w(t) = \Delta t \sum_{m=-\infty}^{\infty} s(t)\delta(t - m\Delta t) \, . \tag{3.25}$$

The FOURIER transform of this function yields

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathrm{d}\, t\ \hat{s}(t)\, \mathrm{e}^{-ixt} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathrm{d}\, t\ \Delta t \sum_{lm=-\infty}^{\infty} s(t)\delta(t - m\Delta t)\, \mathrm{e}^{-ixt} =$$
$$= \frac{1}{\sqrt{2\pi}} \Delta t \sum_{m=-\infty}^{\infty} s(m\Delta t)\, \mathrm{e}^{-ixm\Delta t} \tag{3.26}$$

which is nothing but a FOURIER series. $s(m\Delta t)$ corresponds to the FOURIER coefficients. However, since the function computed by a FOURIER series is periodic always, $f(x)$ is a periodic function. The periodicity interval L is given by

$$L = \frac{2\pi}{\Delta t}\ . \tag{3.27}$$

This means the closer the samples are placed the larger is the periodicity interval L.[5] Or, in other words, the higher the sampling frequency the larger the length of the period.

The notion of k-space was introduced previously to take into account that the MR signal is a function of both gradient strength and time. By using k-space, a formulation of the sampling that is independent of the actual gradient strength can be given. The distance in k-space between two samples is given by

$$\Delta k = \gamma G \Delta t\ . \tag{3.28}$$

The sampled points of the signal are thus $s(m\Delta k)$ and equation 3.26 becomes

$$f(x) = \frac{1}{\sqrt{2\pi}} \Delta k \sum_{m=-\infty}^{\infty} s(m\Delta k)\, \mathrm{e}^{-ixm\Delta k} \tag{3.29}$$

Since the samples $s(m\Delta k)$ are nothing but the FOURIER coefficients they can be written as

$$s(m\Delta k) = s_m = \frac{\sqrt{2\pi}}{L} \int_{-\frac{L}{2}}^{\frac{L}{2}} \mathrm{d}\, x f(x)\, \mathrm{e}^{ixm\Delta k}\ . \tag{3.30}$$

Let's assume that $f(x)$ corresponds to the density of, for example, magnetic moments $\mu(x)$ (without considering the physics of the measurement process). In general, of course, $\mu(x)$ is not a periodic function, and the integral in equation 3.30 can not be limited to a finite interval since the precession of all magnetic moments contribute to the signal. If, however, $\mu(x)$ is limited to $-\frac{L}{2} \leq x < \frac{L}{2}$ because of, for example, the size of the sample, the signal will satisfy the equation for the FOURIER coefficients and $f(x)$ will be an exact estimate for $\mu(x)$. If this condition is not fulfilled aliasing will occur and $f(x)$ will not represent $\mu(x)$ correctly. L is usually called the field of view.

[5] It should be noted that both x and L have the dimension of a frequency.

So far it has been assumed that an infinite number of samples is taken. This, of course, is never the case. If a limited number of samples (e.g. $2n$ samples) is taken, equation 3.29 is transformed to

$$\hat{f}(x) = \frac{1}{\sqrt{2\pi}} \Delta k \sum_{m=-n}^{n-1} s(m\Delta k)\, \mathrm{e}^{-ixm\Delta k} \quad . \tag{3.31}$$

$\hat{f}(x)$ is now an estimation for $f(x)$ that approaches $f(x)$ in case the signal is sampled for an infinite period. In the image, the error that is made by the estimate appears as blurring and limits the resolution. Another result of the finite number of samples is GIBB's ringing which worsens with a decreasing number of samples.

If a finite number of samples of $\hat{f}(x)$ is taken and the number of samples is the same as the number of signal points $\hat{f}(m\Delta x)$ and $s(m\Delta k)$, form a pair of the discrete FOURIER transform. For a given field of view L the resolution Δx is thus given by the number of sampled signal points

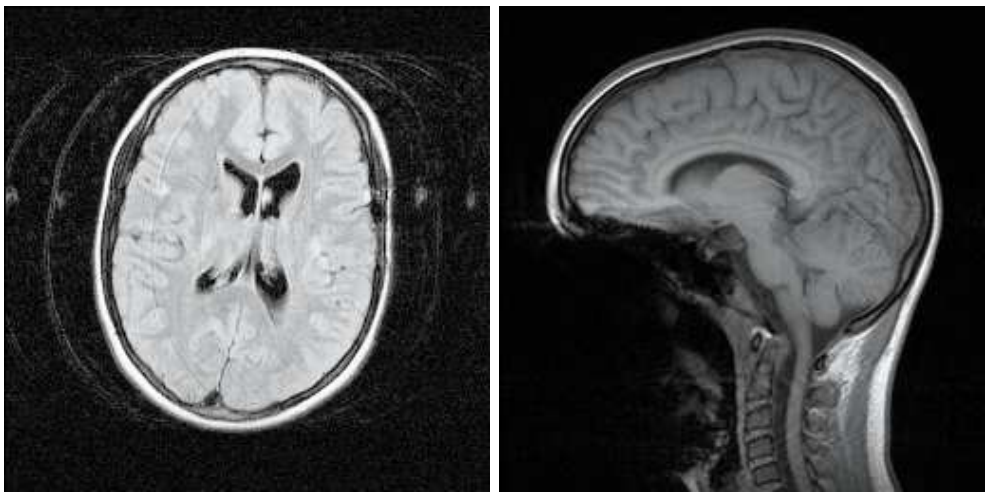
$$\Delta x = \frac{L}{N} \tag{3.32}$$

Together with equation 3.27, this equation relates the resolution to the distance Δk between two k-space points, or, if the gradient strengths are taken into account, to the sampling frequency $\frac{1}{\Delta t}$. If instead of the k-space formulation, the formulation in time space is used, Δx has the dimension of a frequency. $\frac{1}{N\Delta t}$, which is the inverse of the length of the readout interval, is called bandwidth per pixel. This measure quantifies the frequency difference between the centers of two neighboring pixels.

3.4 Artifacts

Considering the complex theory of MRI, it is not very surprising that MR imaging can be affected by a large number of different artifacts. Although the MR scanner itself can be the source of a number of artifacts, starting from imperfect RF pulses, to imperfections of both the static magnetic field and the gradient system to subtle effects like heating of the gradient system due to the strong currents applied, many artifacts are caused by the underlying physics. This on one hand requires a careful choice of measuring parameters, but on the other hand makes MRI extremely powerful. It can be summarized in following quote *one man's artifact is another man's signal*.

While it is out of the scope of this thesis to describe all kind of possible artifacts and their sources, a few examples of artifacts shall be given to illustrate the complexity of MR data. The large number of possible artifacts makes it very hard to develop image processing techniques that can be applied to all sort of MR data.



(a) (b)

Figure 3.10: Illustration of selected MRI artifacts. (a) displays ghosts caused by motion, (b) a susceptibility artifact (note that the mouth and the nose are not visible at all!) caused by metal brackets.

In clinical applications, the probably most common artifacts are motion artifacts. Motion artifacts are a result of the considerable time that often is required to sample the full k-space. For example, acquiring a high resolution SWI dataset of the brain typically takes 10-15 min. Since the image is generated from k-space using the FOURIER transform, any error in k-space propagates to the full image. In images, motion artifacts appear as "ghosts" (fig. 3.10(a)). How much an image is affected by motion depends on the type of motion, on the time at which motion occurred during k-space sampling and on the k-space sampling scheme. Periodic motions have the strongest impact on images. Also, if motion occurs while the center of k-space is sampled the effects on the images are considerable.

Motion artifacts are caused by any kind of motion, not only by bulk motion. For example breathing or even blood flow cause motion artifacts. Flow artifacts are special because they can be mitigated by carefully designed sequences.

Another important source of image artifacts are susceptibility differences in the measured object. As discussed before, the magnetic flux in matter depends on a property called susceptibility χ (see equation 3.22). While materials with different susceptibility usually have well defined boundaries and interfaces, the effect on the magnetic field extends well beyond those boundaries. Magnetic field inhomogeneities caused by such susceptibility differences can lead to signal loss because of dephasing. Figure 3.10(b) shows a sagittal image of a patient with metal brackets obtained with a two-dimensional gradient echo sequence. Mouth and nose are not visible because of the strong distortions of the magnetic field.

On the other hand susceptibility differences can be exploited by carefully designed techniques such as SWI to image structures that have dimensions smaller than the voxel size. For example, small venous vessels can be detected because the susceptibility of deoxygenated blood differs from parenchyma.

Not only susceptibility can cause the magnetic field strength to deviate from the expected values. Another source of artifacts caused by magnetic field imperfections are eddy currents. Eddy currents are induced by fast changing gradients and thus typically occur in fast k-space acquisition techniques only. The field distortions caused by eddy current can cause severe distortions of the image.

The previous artifacts were caused by the LARMOR frequency deviating from the expected value because of macroscopic inhomogeneities of the magnetic field. Nevertheless, such an aberration of the LARMOR frequency can be caused not only by the macroscopic magnetic field differing from the presumed value, but also by microscopic effects. The microscopic magnetic field, and thus the LARMOR frequency, depends on the chemical environment. For example the LARMOR frequency of a Hydrogen nucleus that is bound to a lipid differs from one in a water molecule by 214 Hz at 1.5 T [2]. This can cause some sort of an aliasing artifact if the bandwidth per pixel is too small. In this case, the image attributed to protons bound to fat will be shifted from the water image. This shift can be quite large. Even if the bandwidth is chosen such that no shift occurs, the signal in a given voxel can vary depending on the echo time, since depending on the acquired phase shift the water and fat signals can add up constructively or destructively.

So far the discussion of artifacts has only mentioned artifacts that are caused by a deviation of the evolution of the magnetic moments from the ideal case. Nevertheless it is obvious that if the initial magnetization is imperfect the resulting image differs from the expected one. There are a multitude of reasons why an excitation might be less then perfect. It starts with distortions of the frequency spectrum of a pulse caused by truncation. In principle, all the mechanisms described previously can also cause imperfect magnetization. In addition to these inhomogeneities of the B_1 field might be present. These inhomogeneities can be caused not only by the design of the RF coil, but also by the interaction of the sample with the RF field. This interaction might not only cause damping but also resonances. Thus, because of B_1 inhomogeneities, even an image of a perfectly homogeneous sample, that is not effected by any other source for artifacts (e.g. susceptibility or flow), might be inhomogeneous.

List of References

[1] Lauterbur, P. *Image formation by induced local interaction: Examples employing nuclear magnetic resonance.* Nature, **242**, 190–191 (1973).

[2] Haacke, E. M., Brown, R. W., Thompson, M. R., and Venkatesan, R. *Mag-*

netic Resonance Imaging – Physical Principles and Sequence Design. Wiley-Liss (1999).

[3] Twieg, D. B. *The k-trajectory formulation of the NMR imaging process with applications in analysis and synthesis of imaging methods.* Medical Physics, **10**(5), 610–621 (1983).

[4] Liang, Z. P. and Lauterbur, P. *Principles of Magnetic Resonance Imaging.* IEEE Press (2004).

[5] Rauscher, A. *Phase information in magnetic resonance imaging.* Ph.D. thesis, Technische Universität Wien (2005).

[6] Bracewell, R. N. *The Fourier Transfrom and Its Applications.* McGraw-Hill, 3 edition (1999).

4
MR Phase Images

In the previous chapter, an important property of MRI signals and images was omitted: namely that MRI signals and images consist of complex numbers. This might seem strange because measured physical quantities such as voltage always are real values. This chapter starts with an explanation of the complex nature of MRI signals. Afterwards the properties of noise in MRI images and especially in phase images is discussed. This is followed by an analysis of the effects of the limited domain of the phase, namely phase wraps, and the problem of phase unwrapping. Finally, the problem of reconstructing phase images from data acquired with modern phased array coils is touched briefly.

4.1 The Origin of Complex Signals in MRI

In section 2.6 it was explained how the precession of magnetic moments induces a signal in a coil. In general, the voltage of this signal is measured. Of course, the measured value is a real value. Since the signal depends only on the rotation of $\boldsymbol{M}_\perp$ its frequency is given by the LARMOR frequency ω_0. At 1.5 T and for the hydrogen atom it is thus 63.72 MHz.

A derivation of how linear field gradients can be used for encoding space with frequencies and how this can be exploited to use NMR as an imaging modality was given in the previous chapter. Typical field gradients are of the order of a few 10 mT/m. Let's assume a gradient strength of 10 mT/m in x direction and an isotropic voxel of 1 mm^3. The difference between the LARMOR frequencies at the center of two neighboring voxels is

$$\Delta\omega = \gamma \Delta B_0 = \gamma 10^{-5} \approx 424.8\,\text{Hz} \tag{4.1}$$

which is very small compared to the LARMOR frequency.

Thus, the total signal is a superposition of waves that differ only slightly from the LARMOR frequency ω_0. Since only the frequency differences contain information, ω_0 is removed from the signal by demodulation. This procedure is called quadrature detection. It has the same effect as if the signal was acquired in the rotating frame [1].

Demodulation is achieved by multiplying the signal by a sine or cosine of a reference signal having the frequency ω_0 (or a frequency close to ω_0). Using the trigonometric relations one obtains

$$\sin(\omega_0 + \Delta\omega)t \sin \omega_0 t = \frac{\cos \Delta\omega t + \cos(2\omega_0 + \Delta\omega)t}{2} \tag{4.2}$$

and

$$\sin(\omega_0 + \Delta\omega)t \cos \omega_0 t = \frac{\sin \Delta\omega t + \sin(2\omega_0 + \Delta\omega)t}{2} \tag{4.3}$$

respectively. The terms containing the $2\omega_0$ offset can be suppressed by a low pass filter. The final signal will thus oscillate with $\Delta\omega$. The two equations can also be interpreted as the real and imaginary part of a complex signal

$$S(t) = S_{\mathrm{re}}(t) + iS_{\mathrm{im}}(t) = \cos \Delta\omega t + i \sin \Delta\omega t \; . \tag{4.4}$$

An interpretation of the physical meaning of the complex signal can be derived in following way: from equation 2.64 it is known that the signal is proportional to the integral of the product of $\boldsymbol{B}_{\mathrm{receive}}(\boldsymbol{r})$ and the magnetic moments $\boldsymbol{\mu}(\boldsymbol{r}, t)$

$$s(t) \propto -\frac{\mathrm{d}}{\mathrm{d}\, t} \int_{\mathrm{sample}} \mathrm{d}^3\, r \boldsymbol{\mu}(\boldsymbol{r}, t) \cdot \boldsymbol{B}_{\mathrm{receive}}(\boldsymbol{r}) \; . \tag{4.5}$$

Since the derivative of the relaxation can be neglected compared to the rotation of $\boldsymbol{\mu}$ equation 4.5 can be simplified to

$$s(t) \propto -\frac{\mathrm{d}}{\mathrm{d}\, t} \int_{\mathrm{sample}} \mathrm{d}^3\, r \mu_x(\boldsymbol{r}, t) B_{\mathrm{receive}_x}(\boldsymbol{r}) + \mu_y(\boldsymbol{r}, t) B_{\mathrm{receive}_y}(\boldsymbol{r}) \; . \tag{4.6}$$

Remembering that μ_x and μ_y were given by

$$\mu_x = \mu_\perp \cos \omega t \quad \text{and} \quad \mu_y = \mu_\perp \sin \omega t \; , \tag{4.7}$$

the magnetic moment's transverse component $\boldsymbol{\mu}_\perp$ can also be written as a complex number

$$\mu_+ = \mu_\perp \, \mathrm{e}^{i\omega t} \; . \tag{4.8}$$

Also the transverse component of $\boldsymbol{B}_{\mathrm{receive}}$ can be defined to be a complex number

$$B_+ = B_{\mathrm{receive}_x} + iB_{\mathrm{receive}_y} = B_\perp \, \mathrm{e}^{i\theta} \; , \tag{4.9}$$

where $B_\perp = \sqrt{B^2_{\text{receive}_x} + B^2_{\text{receive}_y}}$. Using trigonometric relations, it can be easily shown that the signal given in equation 4.6 is nothing else but the real part of a complex signal given by

$$s'(t) \propto \int_{\text{sample}} \mathrm{d}^3\, r \mu_+(\boldsymbol{r}, t) B^*_+(\boldsymbol{r}) \ . \tag{4.10}$$

By inserting $\omega = \omega_0 + \Delta\omega$ into equation 4.1 one obtains

$$\mu_+ = \mu_\perp \, \mathrm{e}^{i(\omega_0 + \Delta\omega)t} = \mu_\perp \, \mathrm{e}^{i\omega_0 t} \, \mathrm{e}^{i\Delta\omega t} \tag{4.11}$$

thus the components of the complex signal $S(t)$ obtained by demodulation represent the orientation of the magnetic moment in the rotating frame of reference.

Introducing the k-space notation for imaging gradients and an additional spatial varying phase into equation 4.10 yields (after removing $\mathrm{e}^{i\omega_0}$ by demodulation)

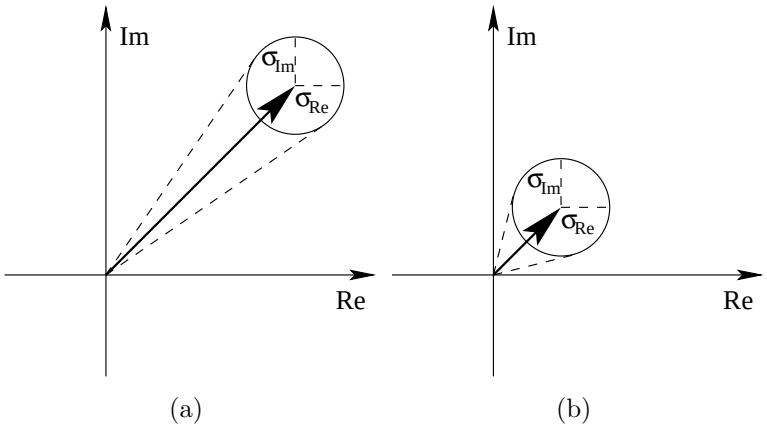
$$S(k_x, k_y) \propto \int_{\text{sample}} \mathrm{d}^3\, r \mu_\perp(\boldsymbol{r}) \, \mathrm{e}^{i\phi(\boldsymbol{r})} \, \mathrm{e}^{ik_x r_x} \, \mathrm{e}^{ik_y r_y} \ . \tag{4.12}$$

The signal $S(k_x, k_y)$ is therefore nothing but the two dimensional FOURIER transform of $\mu_\perp(\boldsymbol{r}) \, \mathrm{e}^{i\phi(\boldsymbol{r})}$. The density of magnetic moments is given by the magnitude of $\mu_\perp(\boldsymbol{r}) \, \mathrm{e}^{i\phi(\boldsymbol{r})}$, and the distribution of their phases by $\phi(\boldsymbol{r})$. This phase depends on the history of a magnetic moment at $\boldsymbol{r}$. A non-zero phase can, for example, be the result of an inhomogeneity δB of the static magnetic field B_0.

4.2 Noise in MR (Phase) Images

As any experimentally obtained data, MR data is affected by noise. In case of MR, noise can originate from a number of sources: random charge fluctuations in the sample, electrical noise in the coil and the receiver as well as RF transmitter noise [2]. However, if the coil is loaded significantly (i.e. substantial damping is caused by the sample), noise currents in the sample are the dominating noise source [3] and other sources can be neglected. This is of importance for the discussion of the noise properties of the two demodulated channels (i.e. the real and imaginary part of the signal). While transmitter and receiver may cause correlated noise in the two channels, randomly fluctuating currents in the sample have a random phase and thus produce uncorrelated noise [2]. In general the coil will be loaded sufficiently, because this also results in an efficient absorption of RF energy during the exciting pulse. Thus for the following discussion the noise of the real and imaginary part of the complex signal will be assumed to be uncorrelated. The noise in each channel can then be treated as an independent random variable.

The noise of each channel can be assumed to be white noise and to satisfy a GAUSSIAN distribution with zero mean. Because the FOURIER transform is



(a) (b)

Figure 4.1: Illustration of the relationship between the variance of the phase and the magnitude. An example of a pixel with a high intensity is given in (a). The variance of the real and imaginary part are indicated by the circle. The variance of the phase is illustrated by the cone. The opening of the cone (i.e. the variance of the phase) in (a) is much smaller compared to the one of a pixel with a smaller magnitude but the same variances of the real and imaginary part (b).

orthogonal and linear it preserves the GAUSSIAN characteristics of noise. Furthermore, because of the FOURIER transform, the variance will be uniform over the image and the real and imaginary parts will be uncorrelated [4].

Both, magnitude and phase images, are computed from the complex image using non-linear mappings. The noise in the respective image is thus no longer GAUSSIAN.[1] It can be shown that in case of the magnitude image the noise is distributed according to a RICIAN distribution [4]. In case of regions where only noise is present this distribution reduces to the RAYLEIGH distribution

$$p(I) = \frac{I}{\sigma^2}\, \mathrm{e}^{-\frac{I^2}{2\sigma^2}} \tag{4.13}$$

with I being the modulus of the voxel's complex value and σ the variance of the real and imaginary part under the assumption that both variances are the same. If the signal to noise ratio (SNR) becomes sufficiently high (SNR > 3) the RICIAN distribution can be approximated by a GAUSSIAN distribution [4].

The distribution of the phase noise $\Delta\phi$ is more complicated. Nevertheless two situations for which simple equations can be given can be distinguished [4]. The first limit occurs in regions where only noise is present. In this case the phase

[1] In case of the magnitude image this becomes obvious by considering voxels in which no signal but noise is present. If the noise was GAUSSIAN the magnitude would assume negative values, which is not possible because of the definition of the magnitude.

noise is distributed uniformly

$$p(\Delta\phi) = \begin{cases} \frac{1}{2\pi} & \text{if} - \pi < \Delta\phi < \pi \\ 0 & \text{otherwise} . \end{cases} \tag{4.14}$$

If the SNR is substantially large (SNR > 3) the phase noise $\Delta\phi$ can be approximated by a GAUSSIAN distribution

$$p(\Delta\phi) \approx \frac{\bar{I}}{\sqrt{2\pi}\sigma} \, \mathrm{e}^{-\frac{\Delta\phi^2}{2(\sigma/\bar{I})^2}} \quad . \tag{4.15}$$

The variance is given by

$$\sigma_{\Delta\phi} = \frac{\sigma}{\bar{I}} \approx \frac{\sigma}{I} \, . \tag{4.16}$$

The variance of phase noise $\Delta\phi$ is thus independent of the phase angle and inversely proportional to the magnitude. This is also valid on a pixel by pixel basis [2]. The only condition is that the SNR has to be sufficiently large. Figure 4.1 provides an schematic illustration of the relationship between the variance of phase noise and the magnitude of a complex value.

4.3 Phase Wraps and Phase Unwrapping

In section 4.1 not only the origin of complex values in MRI was derived, but a physical interpretation was given as well. According to this interpretation, physical quantities can be easily attributed to the components of the complex numbers if the polar notation is used. The magnitude I corresponds to the magnitude of the magnetic moment $\mu_\perp(\boldsymbol{r})$ and the phase $\phi(\boldsymbol{r})$ to the orientation of this vector in a coordinate system rotating at the LARMOR frequency.

In chapter 3 a few effects that can lead to phase offsets were discussed briefly. These effects were for example deviations of the magnetic field strength B_0 from the expected value or flow. If the static magnetic field B_0 differs from the expected value by ΔB the phase offset $\Delta\phi$ in a voxel is given by

$$\Delta\phi = -\gamma \Delta B \Delta t \tag{4.17}$$

with Δt being the echo time. From this equation it is obvious that, in principle, the phase offset $\Delta\phi$ is boundless. On the other hand, the phase of a complex number is limited to $[-\pi, \pi)$ and is periodic. Thus, the phase image $\phi(\boldsymbol{r})$ is the result of a mapping of a boundless function to a variable that has a limited domain. This mapping is commonly described by the so called wrapping operator W [5]

$$\phi_{\text{wrapped}}(\boldsymbol{r}) = \mathrm{W}[\phi(\boldsymbol{r})] \, . \tag{4.18}$$

Although, in general, the "true" phase is a continuous function, the wrapped phase $\phi_{\text{wrapped}}(\boldsymbol{r})$ has discontinuities if the "true" phase has values that are not

within the interval $[-\pi, \pi)$. The result is aliasing which is often referred to as phase wrapping. In phase images these discontinuities appear as borderlines between black and white. Figure 4.2(a) gives an example of a phase image in which phase wraps occur. The underlying continuous function is a two dimensional GAUSS function that ranges from 0 to 5π. It thus exceeds the domain of the phase.

The wrapped phase $\phi_{\text{wrapped}}(\boldsymbol{r})$ differs from the true phase $\phi(\boldsymbol{r})$ by multiples of 2π

$$\phi(\boldsymbol{r}) = \phi_{\text{wrapped}}(\boldsymbol{r}) + 2\pi m(\boldsymbol{r}) \tag{4.19}$$

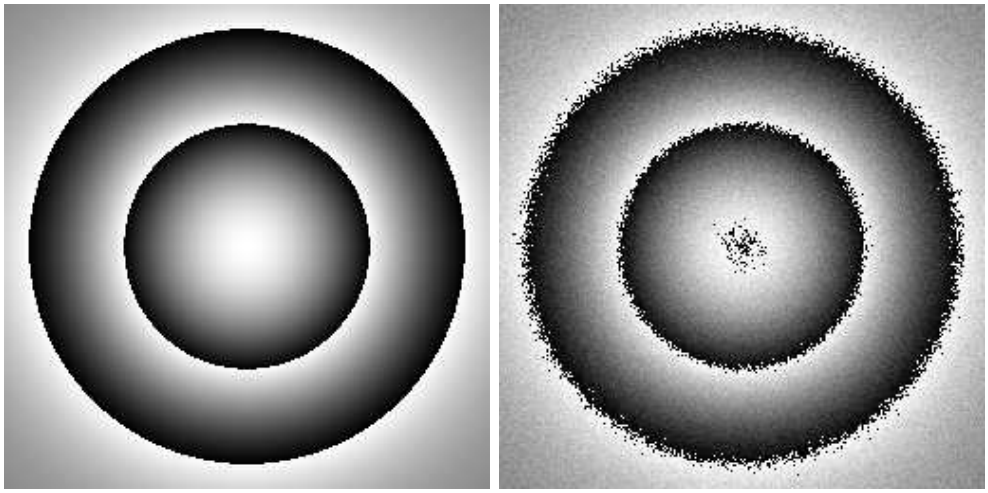
with $m(\boldsymbol{r})$ being integer numbers. The problem of determining a function $m(\boldsymbol{r})$ such that the phase $\phi(\boldsymbol{r})$ is a continuous function is known as phase unwrapping. The problem of phase unwrapping is extremely difficult, if not unsolvable, because of noise and since phase images represent samples of a continuous function at discrete points.

Because of the discrete support, phase images can be affected by a second type of aliasing. This type of aliasing occurs when the phase topography is very steep and the phase difference between two neighboring pixels exceeds 2π. Such a wrap is not necessarily visible in the image and cannot be resolved without additional knowledge [5]. The lines along which the first type of aliasing occurs are often called *fringelines*. In the second case the borderlines are referred to as *cutlines* [6]. Fringelines and cutlines may or may not overlap.

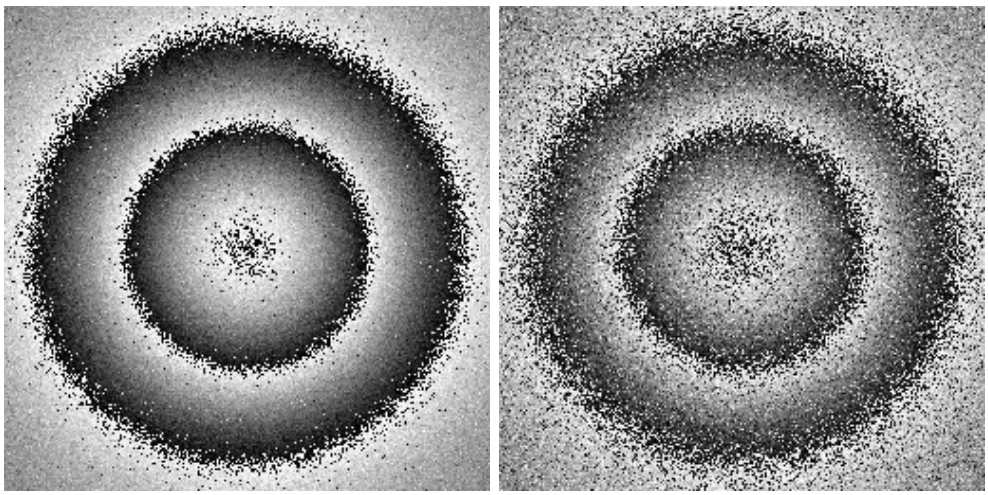
Two types of fringe lines can be distinguished [6]: closed fringelines and open ended fringelines. The fringelines in figure 4.2(a) are all closed fringelines. Unwrapping an image that has closed fringelines only is straightforward. The situation is very different if open ended fringelines are present (fig. 4.3). In this case, the image contains singularities. If poles are present it is impossible to describe the phase with a continuous function. It is therefore impossible to unwrap the image without using additional knowledge or assumptions. Two trivial examples of possible configurations of poles are shown in figure 4.4. In these cases, one could assume that a cutline connects both poles along the shortest path. In real images, more than two poles are present usually. In such a situation connecting poles is ambiguous not only because of the choice of the cutline, but also because of the pairing of poles. Nevertheless there exists a number of algorithms that attempt unwrapping by placing cutlines between poles [5, 6].

A closer look at open ended fringe lines reveals, that open ended fringe lines themselves can be divided into two groups. In areas where there is either no or just little signal present, open ended fringe lines occur often. One reason is that the phase is not defined if the magnitude is zero, the other is that phase noise is distributed uniformly in those regions. In this situation unwrapping still is possible if areas of low signal are avoided.

The second group of open ended fringe lines can be found in areas where there is a sufficiently large magnitude. As discussed before such a situation is possible only if a cutline is present - in other words the phase topography must



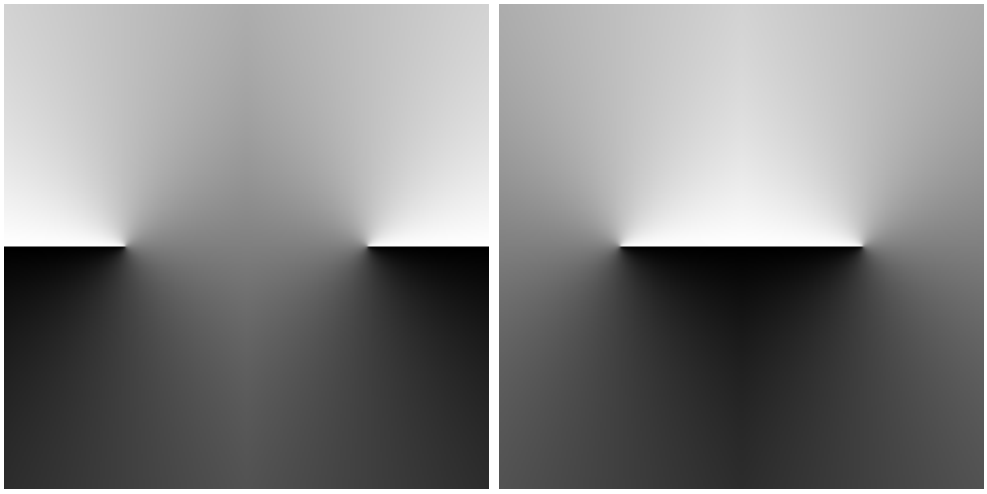
(a) (b)



(c) (d)

Figure 4.2: Due to the limited domain of the phase, phase images that are a mapping of a continuous function (in this example a two dimensional Gauss function) can have discontinuities if the underlying function has values that are not within the interval $[-\pi, \pi)$ (a). The effect of noise is shown in (b) to (d). The signal to noise ratio (SNR) is 5 in (b), 2.5 in (c), and 1.25 in (d), respectively.

Figure 4.3: Example of an open ended fringeline. The image was generated by computing arctan $\frac{y}{x}$.



(a) (b)

Figure 4.4: Phase images with two poles. Both images cannot be unwrapped without making assumptions where cutlines might be present.

be extremely steep within the pixels that constitute the cutline. However, in all phase images investigated during this thesis, it was found that if a phase topography reached a steepness that made it likely that a cutline might occur in some place, signal loss due to dephasing occurred. The only exception were images that were either not reconstructed correctly, or that were obtained from data that was subject to a filter.

Since phase unwrapping is a very hard problem, there exists a large number of different approaches to solve this problem. Algorithms were developed with having different applications in mind. Areas in which phase unwrapping is of importance include Synthetic Aperture Radar (SAR) interferometry, optics, as well as MRI.

A program that has become established for unwrapping MR images is PRELUDE [7] which is part of the FSL Analysis Package (Oxford Centre for Functional Magnetic Resonance Imaging of the Brain, United Kingdom). PRELUDE is based on the "Split and Merge" principle. Other approaches to phase unwrapping include Minimum Spanning Trees (e.g. [8, 9]), region growing (e.g. [10, 11, 12]), neural networks (e.g. [13]), various approaches to minimizing errors between the unwrapped phase image and the wrapped phase image (e.g. [14, 15]) and many others (e.g. [16, 17, 18, 19]). A detailed overview and a comparison of several algorithms can be found in [5].

4.4 Phase Images and Phased Arrays/Parallel Imaging

Fast imaging techniques such SiMultaneous Acqusition of Spatial Harmonics (SMASH) [20], SENSitivity Encoding (SENSE) [21] or GeneRalized Autocalibrating Partially Parallel Acquisitions (GRAPPA) [22] require phased arrays to reconstruct information that is missing because of k-space undersampling. Phased arrays consist of a number of decoupled coils that acquire data simultaneously [23]. A complex image can be reconstructed from the data of each coil. Depending on the arrangement of the coils and the coils' geometry, the coils' sensitivities to a signal coming from a given area of the measured object will differ. An example is given in figure 4.6 and figure 4.7. The magnitude and phase image of each channel of the 12 channel SIEMENS TIM Trio head-coil depicted

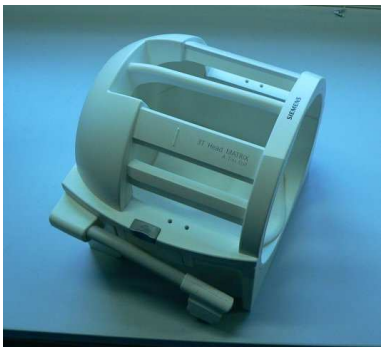


Figure 4.5: A standard 12-channel SIEMENS TIM Trio head coil.

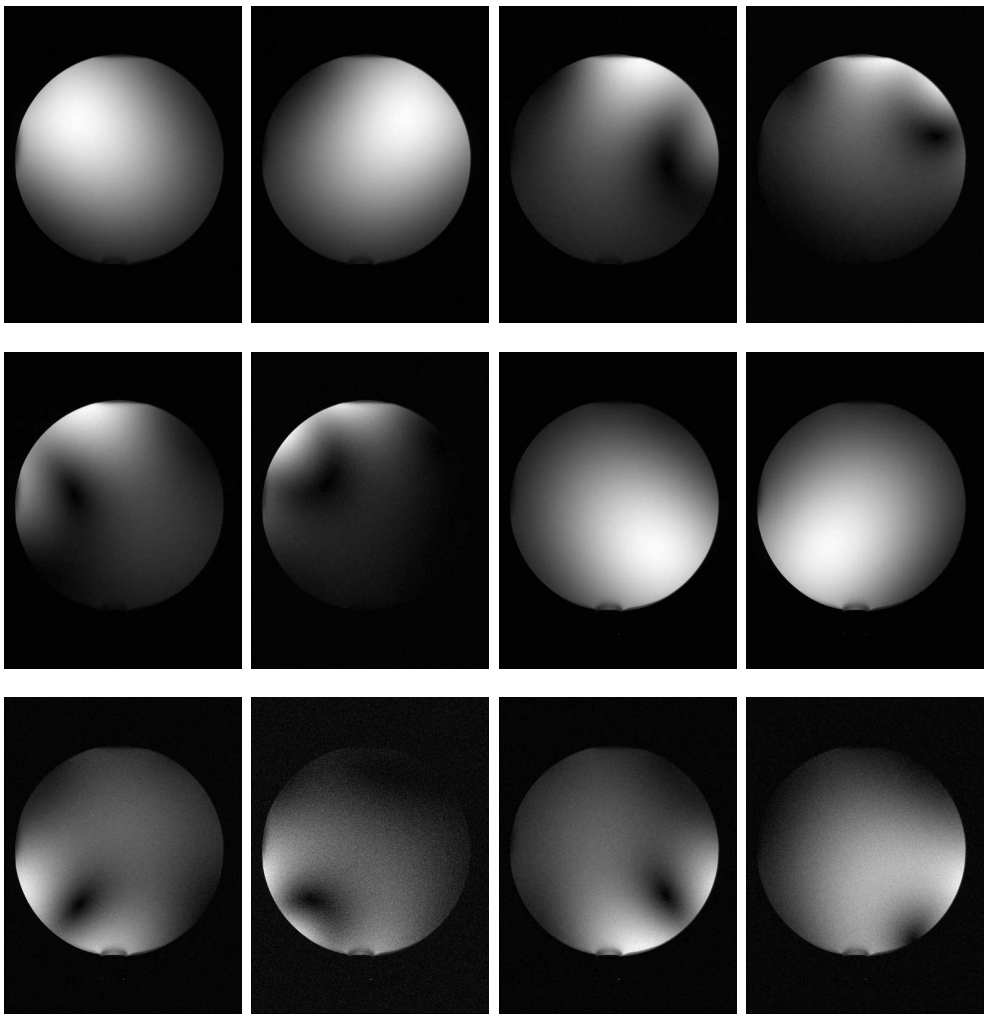


Figure 4.6: Magnitude images of a homogeneous spherical water-phantom. Each image corresponds to single channel of a 12 channel Siemens TIM Trio head coil.

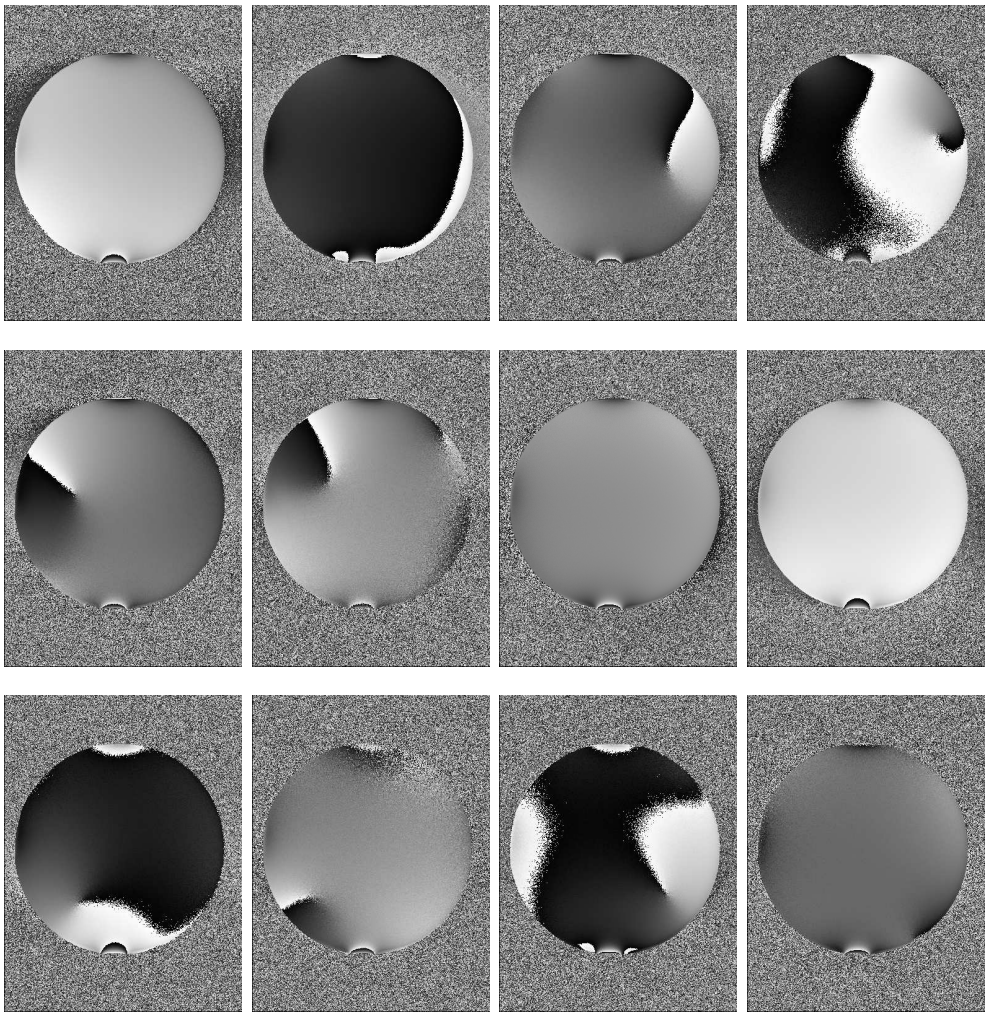


Figure 4.7: Phase images of the same phantom. Each phase image corresponds to single channel of a 12 channel SIEMENS TIM Trio head coil.

in figure 4.5 are shown in figure 4.6 and figure 4.7, respectively.[2] To obtain a single (complex) MR image the data of these coils have to be combined somehow.

In [23] Roemer *et al.* showed that using global weighting factors and phase shifts to combine the images of a phased array results in an image that is optimal only in a single point. To obtain an image that is optimal in each point, weighting factors and phase shifts have to be a function of position. These parameters reflect the coils' sensitivities. From basic considerations concerning electro-magnetic fields a simple equation for combining the data of a set of N_{coils} coils in a SNR-optimal way was derived. This formula, which has to be evaluated for each point of the image, can be written as

$$\hat{p}_{x,y,z} = \lambda_{x,y,z} \mathbf{s}_{x,y,z}^H \psi^{-1} \mathbf{p}_{x,y,z} \ , \tag{4.20}$$

where $\hat{p}_{x,y,z}$ is the complex value of a voxel at (x, y, z) in the combined image, $\lambda_{x,y,z}$ a real valued scaling factor, $\mathbf{s}_{x,y,z}$ a N_{coils}-dimensional vector consisting of the complex sensitivities of each coil at (x, y, z), ψ the noise correlation matrix of the N_{coils} coils and $\mathbf{p}_{x,y,z}$ a N_{coils}-dimensional vector containing the complex image values of each coil at (x, y, z). $\mathbf{s}_{x,y,z}^H$ denotes the transposed complex conjungate of $\mathbf{s}_{x,y,z}$. Unfortunately, in general, neither $\lambda_{x,y,z}$, $\mathbf{s}_{x,y,z}$ nor ψ are known.[3]

To overcome the problem of unknown sensitivities several assumptions and simplifications are often made for calculating combined images. The simplification made most often is that each coil's signal is by itself an estimate for the coil's sensitivity. It is obvious that the signal must be proportional to the sensitivity. Nevertheless, using the signal as estimate adds a bias, since the signal also depends on properties of the object such as, for example, the proton density. If $\mathbf{s}_{x,y,z}$ is substituted by $\mathbf{p}_{x,y,z}$ and $\lambda_{x,y,z}$ set to $\lambda_{x,y,z} = 1$ and ψ to the unit matrix, equation 4.20 can be rewritten as

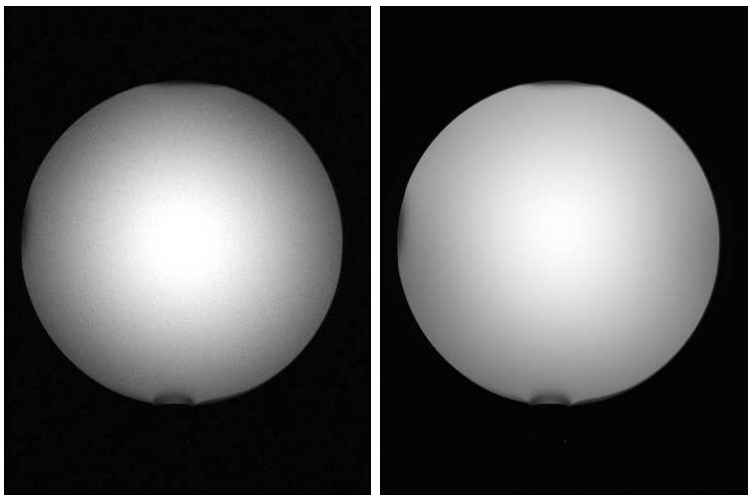
$$\hat{p}_{x,y,z} = \mathbf{p}_{x,y,z}^H \mathbf{p}_{x,y,z} = \sum_{i=1}^{N_{\text{coils}}} |p_{i;x,y,z}|^2 \quad . \tag{4.21}$$

Usually the square root of equation 4.21 is taken. $\hat{p}_{x,y,z}$ then has the dimension of a magnitude. This method is known as *Sum of Squares* (fig. 4.8b shows the *Sum of Squares* image of the images shown in fig. 4.6; fig. 4.8a displays the magnitude image measured with the body coil for comparison). It is intuitively evident that the phase information is lost.

Apparently, combining images to obtain a phase image is more difficult because of the locally varying phase shifts. Nevertheless, phase images are often calculated by summing over all N_{coils} complex images and subsequently taking

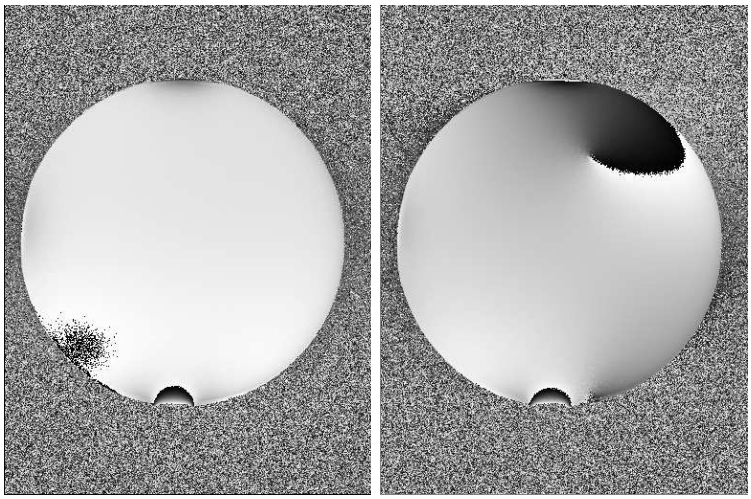
[2] In case of TIM coils the channels do not reflect individual coils of the phased array, but are linear combinations of three coils.

[3] In principle ψ can be measured easily. But in most cases the noise correlation can be neglected and ψ can be set to $\mathbb{I}$ [21].



(a) (b)

Figure 4.8: Magnitude image of the phantom measured with the body coil (a) compared to the *Sum of Squares* image of the indiviudual images of a 12 channel SIEMENS TIM Trio head coil shown in fig. 4.6 (b). Both images were normalized for comparison.



(a) (b)

Figure 4.9: Phase image measured with the body coil (a) and combined phase image computed by complex summation of the individual images of a 12 channel SIEMENS TIM Trio head coil shown in fig. 4.6 and fig. 4.7.

the arc tangent. By using equation 4.20 it can easily be shown that this is the same as assuming that all coils have the same sensitivity and that this sensitivity is constant. This, on the other hand, would mean that the images of all coils would have to be identical, which, of course, is not the case. It is not surprising that the resulting phase image does not resemble the phase as measured with a homogeneous coil, such as the body coil (fig. 4.9a and fig. 4.9b, respectively). Furthermore, the combined phase often is affected by singularities. These singularities impede phase unwrapping. However, even in situations were phase unwrapping would be possible the physical interpretation of the result is questionable.

List of References

[1] Bernstein, M. A., King, K. F., and Zhou, X. J. *Handbook of MRI Pulse Sequences*. Elsevier (2004).

[2] Conturo, T. E. and Smith, G. D. *Signal-to-noise in phase angle reconstruction: dynamic range extension using phase reference offsets*. Magnetic Resonance in Medicine, **15**(3), 420–437 (1990).

[3] Edelstein, W. A., Glover, G. H., Hardy, C. J., and Redington, R. W. *The intrinsic signal-to-noise ratio in NMR imaging*. Magnetic Resonance in Medicine, **3**(4), 604–618 (1986).

[4] Gudbjartsson, H. and Patz, S. *The rician distribution of noisy MRI data*. Magnetic Resonance in Medicine, **34**(6), 910–914 (1995).

[5] Ghiglia, D. C. and Pritt, M. D. *Two-Dimensional Phase Unwrapping*. John Wiley & Sons, Inc. (1998).

[6] Chavez, S., Xiang, Q.-S., and An, L. *Understanding phase maps in MRI: a new cutline phase unwrapping method*. IEEE Transactions on Medical Imaging, **21**(8), 966–977 (2002).

[7] Jenkinson, M. *Fast, automated, N-dimensional phase-unwrapping algorithm*. Magnetic Resonance in Medicine, **49**(1), 193–197 (2003).

[8] An, L., Xiang, Q.-S., and Chavez, S. *A fast implementation of the minimum spanning tree method for phase unwrapping*. IEEE Transactions on Medical Imaging, **19**(8), 805–808 (2000).

[9] Ching, N. H., Rosenfeld, D., and Braun, M. *Two-Dimensional Phase Unwrapping Using a Minimum Spanning Tree Algorithm*. IEEE Transactions on Image Processing, **1**(3), 355–365 (1992).

[10] Baldi, A. *Phase Unwrapping by Region Growing.* Applied Optics, **42**(14), 2498–2505 (2003).

[11] Xu, W. and Cumming, I. *A region-growing algorithm for InSAR phase unwrapping.* IEEE Transactions on Geoscience and Remote Sensing, **37**(1), 124–134 (1999).

[12] Bone, D. J. *Fourier fringe analysis: the two-dimensional phase unwrapping problem.* Applied Optics, **30**(25), 3627–3632 (1991).

[13] Schwartzkopf, W., *et al. Two-dimensional phase unwrapping using neural networks.* In *Proc. IEEE Southwest Symposium on Image Analysis, April 2-4, 2000*, 274–277 (2000).

[14] Stramaglia, S., Nico, G., Pasquariello, G., and Guerriero, L. *Phase unwrapping method based on stochastic relaxation.* In *EUROPTO - Image Processing, Signal Processing, and Synthetic Aperture Radar for Remote Sensing proc. SPIE*, volume 3217, 4–12 (1997).

[15] Frey, B. J., Koetter, R., Moran, J., and Jr., D. J. M. *Variations on phase unwrapping.* In *International Symposium on Information Theory and Its Applications.* International Symposium on Information Theory and Its Applications, Honolulu, USA (2000).

[16] Koetter, R., Frey, B. J., Petrovic, N., and Jr., D. C. M. *Unwrapping phase images by propagating probabilities across graphs.* In *International Conference on Acoustics, Speech, and Signal Processing.* ICASSP (2001).

[17] Frey, B. J., Koetter, R., and Petrovic, N. *Very loopy belief propagation for unwrapping phase images.* In *Advances in Neural Information Processing Systems 14.* MIT Press (2001).

[18] Chen, C. and Zebker, H. *Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization.* Journal of the Optical Society of America. A, Optics, image science, and vision, **18**(2), 338–51 (2001).

[19] Volkov, V. V. and Zhu, Y. *Deterministic phase unwrapping in the presence of noise.* Optics Letters, **28**(22), 2156–2158 (2003).

[20] Sodickson, D. K. and Manning, W. J. *Simultaneous Acquisition of Spatial Harmonics (SMASH): fast imaging with radiofrequency coil arrays.* Magnetic Resonance in Medicine, **38**(4), 591–603 (1997).

[21] Pruessmann, K. P. *Encoding and reconstruction in parallel MRI.* NMR in Biomedicine, **19**(3), 288–299 (2006).

[22] Griswold, M. A., *et al. Generalized autocalibrating partially parallel acquisitions (GRAPPA).* Magnetic Resonance in Medicine, **47**(6), 1202–1210 (2002).

[23] Roemer, P. B., *et al. The NMR phased array.* Magnetic Resonance in Medicine, **16**(2), 192–225 (1990).

5

Phase Unwrapping by Region Growing

Rauscher *et al.* showed that phase unwrapping by a region growing algorithm can be applied to MR data [1]. For this study a simplified variant of a region growing phase unwrapping algorithm originally developed for SAR interferometry [2] was used. This algorithm tries to predict the unwrapped phase of a given pixel by extrapolating the phase from neighboring pixels that have been unwrapped previously. The reliability of such an estimate is estimated using a set of quality criteria. One of these criteria is the local coherence of the phase. The method does not take information that might be contained in the signal intensity into account because in SAR interferometry only the phase is available. Although the algorithm had difficulties with some of the special properties of MR phase images, the results were promising nonetheless. Thus this approach was taken as a starting point for developing an optimized phase unwrapping program. The main goals were to define quality criteria that were more appropriate for MR images and to write an efficient implementation. The modified approach was presented at the Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM) [3] and the European Congress of Radiology (ECR) [4].

This chapter is structured as follows: first the principles of the algorithm are described, secondly the problems encountered in MR phase images using this approach are discussed. Subsequently a set of quality maps are defined and finally their influence on unwrapping is described.

5.1 The Principles of Unwrapping by Region Growing

As described in the previous chapter the problem of phase unwrapping can be summarized as estimating the number of multiples of 2π that have to be added to the phase of a given pixel to yield a smooth phase topography:

$$\phi^u_{i,j} = \phi_{i,j} + 2\pi m_{i,j} \ , \tag{5.1}$$

where $\phi_{i,j}$ is the wrapped phase defined in the range $[-\pi, \pi)$ and $\phi^u_{i,j}$ the unwrapped phase of a voxel (i, j). The idea of unwrapping by region growing is to try to predict the unwrapped phase for each pixel neighboring a region for which the value of the unwrapped phase is known already. Each pixel that has been unwrapped this way is then added to the region. These steps are repeated until the whole image is unwrapped.

The problem thus consists of two sub-problems:

- how to predict the unwrapped phase of a pixel from pixels in its neighborhood that have been unwrapped in previous iterations, and
- how to estimate how reliable such a prediction is.

Especially latter is of great importance, because it essentially controls which pixels are added to a region of unwrapped pixels and thus how the region grows.

Prediction of the Unwrapped Phase

The approach on which this work is based uses linear extrapolation along short, straight lines to calculate predictions for an unwrapped pixel [2]. Predictions are calculated along one or two pixel long lines, depending on how many unwrapped pixels are available on this line. Let l denote the index of a line that belongs to a set of lines through pixel (i, j) and $i', j'; l$ and $i'', j''; l$ the nearest neighbor and the next nearest neighbor to (i, j) on the line l, respectively. The linear extrapolation $\phi^p_{i,j;l}$ for the phase along this line is then given by

$$\phi^p_{i,j;l} = \begin{cases} 2\phi^u_{i',j';l} - \phi^u_{i'',j'';l} & \text{if two unwrapped pixels are along the line } l \\ \phi^u_{i',j';l} & \text{if one unwrapped pixel is along the line } l. \end{cases} \tag{5.2}$$

Depending on the direction of the line, the predictions might vary. One reason for this variation can be that only a single pixel is available on a given line, while other lines are comprised of two pixels. Another cause can be that the underlying phase topography is not linear, and finally, noise. It is thus necessary to calculate a common prediction from these individual predictions. Since a

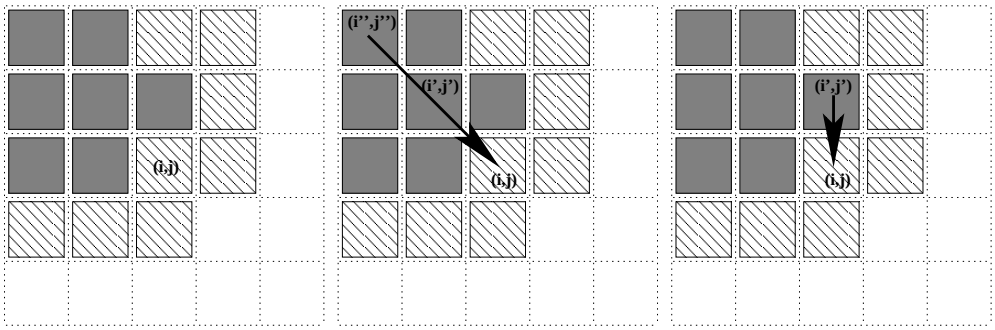


Figure 5.1: Illustration of the unwrapping algorithm's principle. The pixels of the unwrapped region are indicated by filled squares. The pixels of the fringe, which the algorithm will attempt to unwrap during the next iteration, are marked by hatched boxes. (b) and (c) display two of the three prediction lines that are used to unwrap the pixel (i, j).

higher confidence can be attributed to estimations that are obtained using longer lines, the common prediction is calculated by taking a weighted average

$$\phi^p_{i,j} = \frac{1}{\sum_{l=1}^{n} w_{i,j;l}} \sum_{l=1}^{n} w_{i,j;l}\, \phi^p_{i,j;l} \; . \tag{5.3}$$

The weights are defined as

$$w_{i,j;l} = \begin{cases} 1 & \text{if the line } l \text{ is comprised of two unwrapped pixels} \\ \frac{1}{2} & \text{if one unwrapped pixel is along the line } l. \end{cases} \tag{5.4}$$

The multiplication factor $m_{i,j}$ in equation 5.1 can be calculated from the estimated phase $\phi^p_{i,j}$ by simply taking next nearest integer to the ratio of the difference between the predicted phase $\phi^p_{i,j}$ and the wrapped phase $\phi_{i,j}$ and 2π

$$m_{i,j} = \text{round}\left(\frac{\phi^p_{i,j} - \phi_{i,j}}{2\pi}\right) \; . \tag{5.5}$$

This estimation is repeated for each pixel in the direct neighborhood of a region of pixels that have been unwrapped already. Although, in general, an arbitrary set of prediction lines can be defined, for computational efficiency it makes sense to use horizontal, vertical and diagonal lines only. Figure 5.1 provides an illustration of the principle for estimating the unwrapped phase for a pixel in the fringe around a region that has been unwrapped previously. Gray boxes symbolize the unwrapped pixels of a certain regions. The hatched boxes are the pixels that belong to the region's fringe. The algorithm will try to unwrap these pixels next. Figure 5.1(b) and figure 5.1(c) give an example for a two pixel long and a one pixel long prediction line, respectively.

Reliability Criteria

As mentioned beforehand, evaluating the reliability of a prediction of an unwrapped phase before adding it to a region is of utmost importance. The errors introduced by adding incorrectly unwrapped pixels can propagate easily, since in the worst case the algorithm bases its predictions using a single pixel only. The reliability of an unwrapped phase can be checked in various ways. First of all the reliability of a prediction can be tested by checking the agreement of the predictions along the various lines. Secondly, the quality of the data contained in the pixel can be evaluated. In total three reliability criteria were proposed [2]. Two of those three criteria judge the estimate for the unwrapped phase. The third one examines the data quality.

An obvious measure for the quality of the estimate of the predicted phase $\phi^p_{i,j}$ is the variance of the predictions $\phi^p_{i,j;l}$ along the different prediction lines. A similar measure for the variability is the average of the absolute difference of a prediction $\phi^p_{i,j;l}$ from the weighted average $\phi^p_{i,j}$. Although both measures have the same purpose the advantage of the latter one is that it is less computationally expensive. The different predictive strengths of one and two pixel long prediction lines can be taken into account by calculating a weighted average. The variability measure can thus be written as

$$v_{i,j} = \frac{1}{\sum_{l=1}^{n} w_{i,j;l}} \sum_{l=1}^{n} w_{i,j;l} \left| \phi^p_{i,j;l} - \phi^p_{i,j} \right| \ . \tag{5.6}$$

$v_{i,j}$ is positive always and can assume any value larger than 0. The corresponding reliability criteria is that $v_{i,j}$ has to be below a threshold t_v, i.e.

$$v_{i,j} < t_v \ . \tag{5.7}$$

The second reliability criteria that judges the prediction uses the difference between the unwrapped phase $\phi^u_{i,j}$ and the predicted phase $\phi^p_{i,j;l}$

$$\Delta\phi_{i,j} = \left| \phi^p_{i,j} - \phi^u_{i,j} \right| \ . \tag{5.8}$$

It is motivated by the effects of rounding in equation 5.5. If $\phi^p_{i,j} - \phi_{i,j}$ is close to π the ratio becomes ≈ 0.5. In this situation an additional small error of the estimate or a small amount of additional noise can lead to a different value for $m_{i,j}$ and thus of the unwrapped phase $\phi^u_{i,j}$. Therefore an unwrapped phase $\phi^u_{i,j}$ for which $\Delta\phi_{i,j}$ is close to π is considered less reliable than one for which $\Delta\phi_{i,j} \approx 0$. The reliability criterion using this measure is defined as

$$\Delta\phi_{i,j} < t_{\Delta\phi} \tag{5.9}$$

with the threshold $t_{\Delta\phi}$ having values between 0 and π.

Rejecting a pixel because of these two criteria does not necessarily mean that the pixel might not be unwrapped in a later iteration, even if the thresholds are

kept constant. If the region grows around the pixel predictions along lines that become possible because of newly unwrapped neighboring pixels might improve the prediction for the pixel allowing it to pass the reliability criteria.

The third criterion, evaluates the data itself. For this criterion a quality measure is defined for the data. The quality measure should allow an assessment of a phase prediction's reliability considering the properties of the data surrounding a given pixel. Since in SAR interferometry, only phase information is available, the quality measure has to be based on parameters that can be computed merely from the phase. Such a measure is the local coherence Q^{LC}. It is a measure for how coherent the phase around a given pixel is. It is defined as

$$Q_{i,j}^{\mathrm{LC}} = \frac{1}{d_x \cdot d_y} \sum_{k=i-\frac{d_x}{2}}^{i+\frac{d_x}{2}} \sum_{l=j-\frac{d_y}{2}}^{j+\frac{d_y}{2}} \mathrm{e}^{i\phi_{k,l}} \tag{5.10}$$

where d_x and d_y are the dimensions of the kernel over which Q^{LC} is computed. Q^{LC} lies within the interval $[0, 1]$. It is lowest in noisy areas as well as in regions having a steep phase topography. The reliability criterion is defined as

$$Q_{i,j}^{\mathrm{LC}} > t_Q \tag{5.11}$$

If a pixel passes all three reliability criteria it is added to the region. During each iteration, the algorithm attempts to unwrap an one pixel wide fringe around the region. This procedure is repeated until less than t_N pixels are unwrapped. If less than t_N pixels have been unwrapped the reliability criteria are relaxed. The method described herein uses an approach that slightly differs from the original algorithm. Originally t_Q or t_v and $t_{\Delta\phi}$ (with $t_v = t_{\Delta\phi}$) are relaxed while the other threshold is kept constant [2]. The algorithm described in this thesis relaxes both t_Q and t_v and $t_{\Delta\phi}$ (with $t_v = t_{\Delta\phi}$) in a double loop. t_Q is relaxed in the inner loop and t_v and $t_{\Delta\phi}$ in the outer loop. This is motivated in the following way: as long as the predictions can be made reliably, one can accept pixels for which the data quality measure yields a lower value. On the other hand, when the algorithm starts accepting predictions which are less reliable (a reason for less reliable predictions might be an uneven phase topography), it should use predictions which are based on reliable data.

The algorithm terminates as soon as the thresholds of all reliability criteria are relaxed to their lower limits and less than t_N pixels are unwrapped.

Multiple Regions

Unwrapping can be done by an arbitrary number of regions simultaneously. In this case situations will occur in which more than one region attempts to unwrap the same pixels and an overlap will be created. Even if the unwrapped phase in both regions is reliable, the unwrapped phase of pixels in an overlap might differ by multiples of 2π due to the two regions having had different start values. Since

the algorithm should yield a single unwrapped phase image, it is necessary to attempt to merge the regions. Following method has been suggested for merging two regions [2]:

1. if the overlap between two regions reaches a critical size, i.e. the two regions share more than t_O pixels an attempt to merge the two regions is made. If there are less than t_O pixels in the overlap the algorithm proceeds with unwrapping. Thus the overlap might reach a critical size in a later iteration. A typical value for t_O is $t_O = 3$

2. compute the difference

$$\Delta_{i,j} = \phi^u_{i,j;r_1} - \phi^u_{i,j;r_2} \tag{5.12}$$

   where r_1 and r_2 are the indices of the two overlapping regions and $\phi^u_{i,j;r_1}$ and $\phi^u_{i,j;r_2}$ are the unwrapped phases predicted by those two regions for each pixel in the overlap.

3. determine the most common difference Δ^{mode} between the unwrapped phases of the two regions.

4. apply the following two tests

$$N_\Delta > t_\Delta \tag{5.13}$$

   and

$$\frac{N_\Delta}{N_O} > t_f \tag{5.14}$$

   where N_Δ is the number of pixels for which $\Delta_{i,j} = \Delta^{\text{mode}}$ and N_O is the total number of pixels in the overlap. Typical values for t_Δ and t_f are $t_\Delta = 3$ and $t_f = \frac{3}{4}$, respectively [2]. If both criteria are passed the regions are merged, otherwise all pixels in the overlap are removed from both regions in the hope that an overlap that has a better agreement will be created in a later iteration.

5. merge the regions by adding the difference Δ^{mode} to all pixels of one of the regions.

6. start a new region by selecting a seed point. This keeps the number of regions growing concurrently constant.

Selecting Seed Points

Seed points have to be selected every time a new region is started, i.e. before the first iteration and every time two regions have been merged. A seed has to fulfill two criteria: it must not have been unwrapped already and it should have the highest data quality, i.e. the highest value of Q^{LC} of all pixels that have not been unwrapped. The latter criterion ensures that a region starts with a pixel from which predictions can be made most reliably.

5.2 Issues Encountered in MR Phase Images

Although, in many cases, the approach described above works quite well if applied to MR phase images [1], there are situations in which the results are unsatisfactory. The most common reason are ghosting artifacts which can be caused by subject motion. Even when these ghosting artifacts are hardly visible in the magnitude (fig. 5.2(a)), they can be quite apparent in the phase image (fig. 5.2(b)). As a result the algorithm attempts to unwrap areas with low signal intensities. Since those areas are noisy, the predictions are not very reliable. This causes, on the one hand, a significant increase in computation time, on the other hand errors can accumulate resulting in an inconsistently unwrapped phase image (fig. 5.2(c)). Furthermore the coherence in areas affected by ghosting artifacts is, in general, much higher than in areas that have a steep phase topography but a well defined signal and thus a high SNR (fig. 5.2(d)). Although unwrapping of those areas is desired it is impossible to define thresholds based on the Q^{LC} that allow unwrapping in regions of steep phase topography while avoiding ghosting artifacts. Although, this could be overcome by segmentation and masking, Q^{LC} obviously is not a measure that truly reflects the "quality" of MR images.

5.3 Quality Maps

Definition of Quality Maps

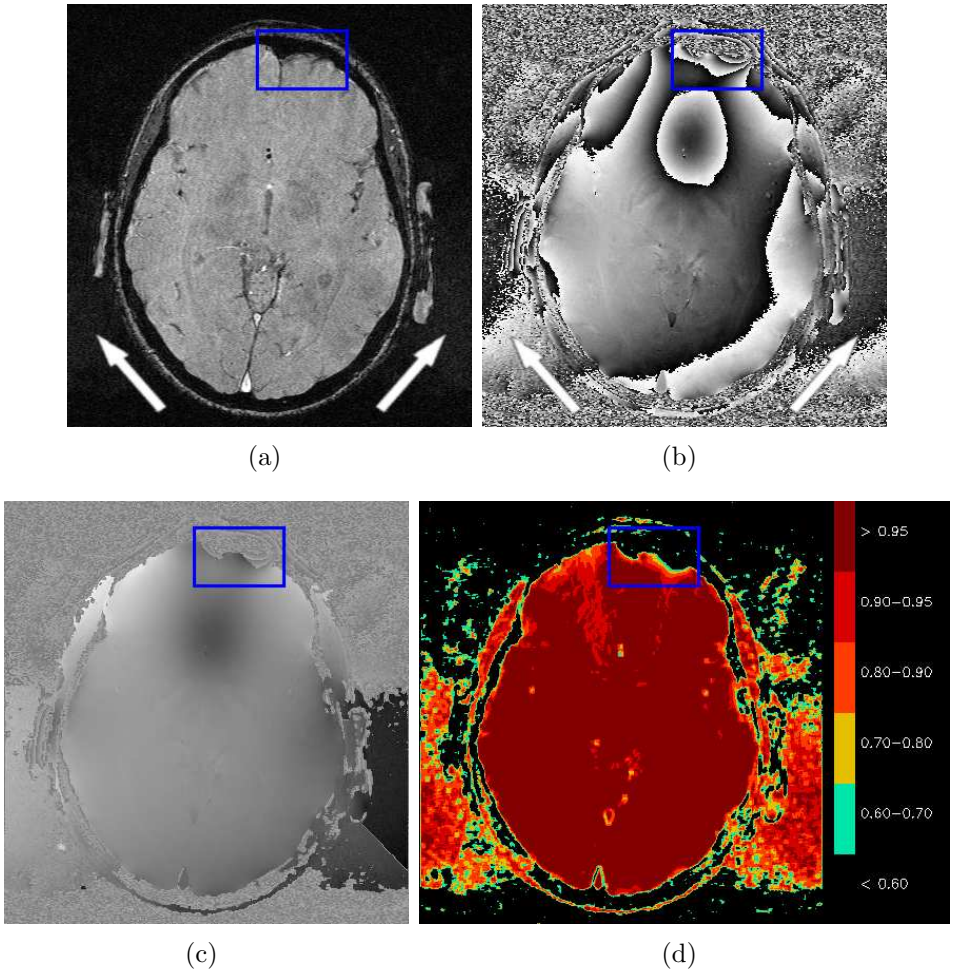
To overcome the problems described before, four additional quality maps were defined and evaluated in addition to the local coherence. The five quality maps (an example for each map is given in fig. 5.3) are motivated and defined in the following manner:

local coherence (Q^{LC}) which was introduced previously is a measure for the coherence of the phase (fig. 5.3(c)). It is calculated as

$$Q^{\mathrm{LC}}_{i,j} = \frac{1}{d_x \cdot d_y} \left| \sum_{l=i-\frac{d_x-1}{2}}^{i+\frac{d_x-1}{2}} \sum_{m=j-\frac{d_y-1}{2}}^{j+\frac{d_y-1}{2}} \frac{\hat{p}_{l,m}}{|\hat{p}_{l,m}|} \right| , \tag{5.15}$$

with (i, j) being the coordinates of the pixel for which the local coherence is calculated and d_x and d_y being the dimensions of the kernel.

average of the complex image (Q^{ACI}) takes both magnitude and phase into account. It should thus suppress ghosting artifacts in areas of low signal intensity and avoid unwrapping of those areas. It reaches its highest values when the phase is coherent and the signal is high (fig. 5.3(d)). It is defined



(a) (b)

(c) (d)

Figure 5.2: Magnitude (a), phase (b) and unwrapped phase images (c) that are affected by steep phase topography in the frontal areas due to susceptibility effects and ghosts caused by subject motion. Due to the large coherence in the areas affected by ghosts (d) the algorithm tries to unwrap areas with low signal intensities before areas with steep phase topography are unwrapped.

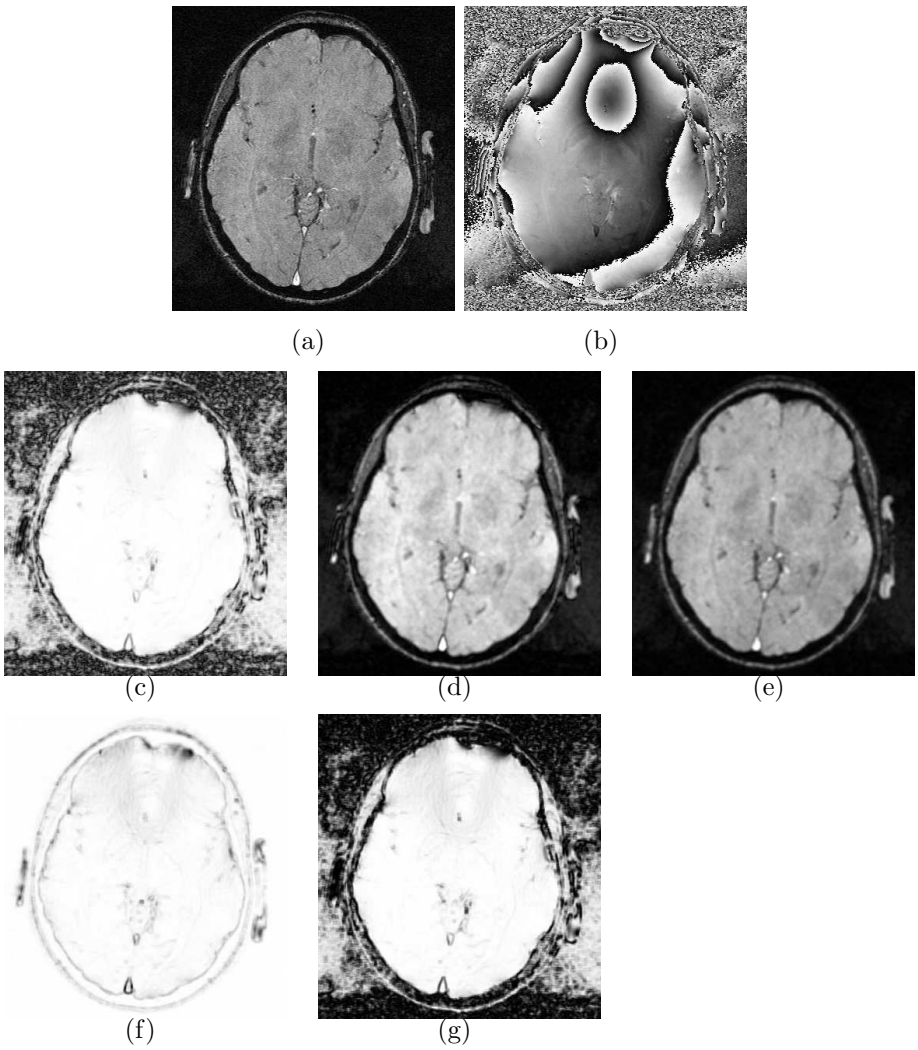


Figure 5.3: Overview of the quality maps for an image with ghosting artifacts. The magnitude and phase of this image are displayed in (a) and (b), respectively. All maps were computed with a rectangular kernel of 5×5 pixels and normalized to $[0, 1]$. (c) shows the local coherence Q^{LC}. The map exhibits the typical properties of Q^{LC}: very low values in areas of steep phase topography even if there is a significant signal, and high values in areas where almost no signal is present. (d) depicts the magnitude of the local average of the complex image Q^{ACI}. Q^{ACI} suppresses areas with low signal intensities, and has small values in regions with steep phase topography. The map of the local average of the magnitude Q^{AM} is shown in (e). The local variance of the complex image Q^{VCI} is given in (f). It separates the object from the background by a dark rim and has low values in areas of steep phase topography. The variance in noisy areas is very low (corresponding to high pixel intensities) even compared to homogeneous areas within the object which does not favor the variance as a quality criterion because it is lower in noise than in the object. (g) is an example for the local variance of the phase Q^{VPH}. It exhibits a behavior similar to Q^{LC}.

as

$$Q_{i,j}^{\mathrm{ACI}} = \frac{1}{\alpha_{\mathrm{ACI}} \cdot d_x \cdot d_y} \left| \sum_{l=i-\frac{d_x-1}{2}}^{i+\frac{d_x-1}{2}} \sum_{m=j-\frac{d_y-1}{2}}^{j+\frac{d_y-1}{2}} \hat{p}_{l,m} \right| , \tag{5.16}$$

where α_{ACI} is a normalization constant ensuring that $Q_{i,j}^{\mathrm{ACI}}$ is within the interval $[0, 1]$.

average of the magnitude (Q^{AM}) is a measure that is based on signal intensity only. It is calculating by smoothing the magnitude image (fig. 5.3(e)) with a rectangular kernel. A normalization is applied so that its values range from 0 to 1. Using the signal intensity as a measure for the phase can be justified since the variance of the phase is inversely proportional to the magnitude [5]. Mathematically Q^{AM} can be formulated as

$$Q_{i,j}^{\mathrm{AM}} = \frac{1}{\alpha_{\mathrm{AM}} \cdot d_x \cdot d_y} \sum_{l=i-\frac{d_x-1}{2}}^{i+\frac{d_x-1}{2}} \sum_{m=j-\frac{d_y-1}{2}}^{j+\frac{d_y-1}{2}} \left| \hat{p}_{l,m} \right| . \tag{5.17}$$

variance of the complex image (Q^{VCI}) measures the variance of the complex image within a rectangular area around a given pixel (fig. 5.3(f)). It was introduced to allow a better delineation of the object, since the variance can be expected to be highest at the borders of the object. By using this measure it should be possible to limit unwrapping to the object. Q^{VCI} is given by

$$Q_{i,j}^{\mathrm{VCI}} = \frac{1}{\alpha_{\mathrm{VCI}}} \left(\bar{p}_{i,j} \bar{p}_{i,j}^{*} - \frac{1}{d_x \cdot d_y} \sum_{l=i-\frac{d_x-1}{2}}^{i+\frac{d_x-1}{2}} \sum_{m=j-\frac{d_y-1}{2}}^{j+\frac{d_y-1}{2}} \hat{p}_{l,m} \hat{p}_{l,m}^{*} \right) + 1, \tag{5.18}$$

where $\bar{p}_{i,j}$ is the local average of the complex image computed with a kernel of the same size ($d_x \times d_y$) and α_{VCI} a normalization constant. The definition of Q^{VCI} ensures that pixels of highest quality (i.e. lowest variance) have a quality of approximately 1.

variance of the phase (Q^{VPH}) is motivated in the same way as Q^{VCI}, but takes only the phase into account (fig. 5.3(g)). It is calculated by applying equation 5.18 on the normalized complex image $\hat{p}'_{i,j} = \hat{p}_{i,j} / |\hat{p}_{i,j}|$. Also this map was motivated by expecting it to yield a better delineation of the object.

Evaluation of Quality Maps

The influence of the five maps on both unwrapping itself and the selection of seed points was investigated. For this study SWI data obtained on a 3 T Bruker

threshold	range
t_Q	$[0.15, 1.00]$
$t_v = t_\Delta$	$[0, \frac{3}{4}\pi]$

Table 5.1: Values of the thresholds that were chosen as default values.

MedSpec (Bruker Medical, Ettlingen, Germany) and a 1.5 T Siemens Magnetom Vision (Siemens, Erlangen, Germany) were unwrapped and analyzed retrospectively. Phantom data were not investigated, because ghosting artifacts are caused by motion and thus phantom data are hardly affected by ghosts. Each map was computed with kernel sizes of 3×3, 5×5 and 7×7 pixels, respectively.

It became immediately obvious that Q^{VCI} did not show the expected properties, namely that it would allow a better delineation of the object. While in principle the variance was higher (and thus Q^{VCI} lower) at the borders of the object than in the neighboring areas, areas within the object exhibited values almost as low as at the borders (fig. 5.3(f)). Also, in general, the variance within the object was higher than in the noise. Furthermore, the variance showed a very large dynamic range. This made it very difficult to define thresholds on (the normalized) Q^{VCI}. Several schemes to scale the variance (by for example the logarithm) before normalization were tested, but none provided a significant improvement. The use of Q^{VCI} was thus not further investigated.

Also Q^{VPH} was used only in the beginning since it exhibited the same properties as Q^{LC}. Thus the use of Q^{VPH} did not provide any improvement over Q^{LC}.

The Influence of Quality Maps on Unwrapping

The best results in terms of how well the algorithm unwrapped areas of steep phase topography while still limiting the unwrapping to the object were obtained using the local average of magnitude Q^{AM} computed with a kernel of 5×5 as quality measure. By using this measure it was possible to choose a threshold that yielded good results for all data-sets. The values for the thresholds are shown in table 5.1. Although the local average of the complex image Q^{ACI} also led to some improvements compared to Q^{LC}, these were not as significant as those obtained with Q^{AM}. Apparently the phase variations in areas of steep phase topography had a larger influence on the map than the signal intensities. An example of an image that is affected by strong field inhomogeneities in the frontal area is shown in figure 5.4. The field inhomogeneity is caused by the paranasal sinuses. The threshold t_Q was optimized for each map to allow the unwrapping to proceed as far as possible without unwrapping the background. Using Q^{AM} (fig. 5.4(e)) clearly leads to the best results.

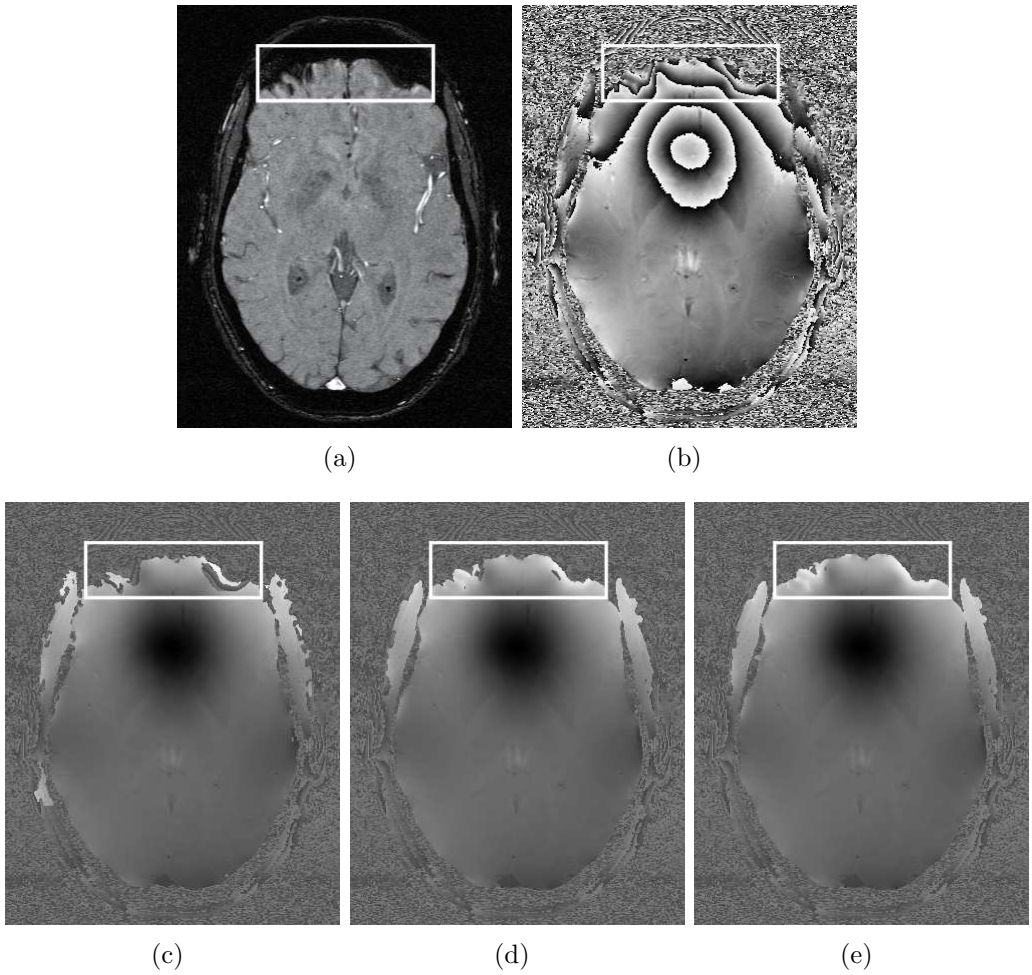
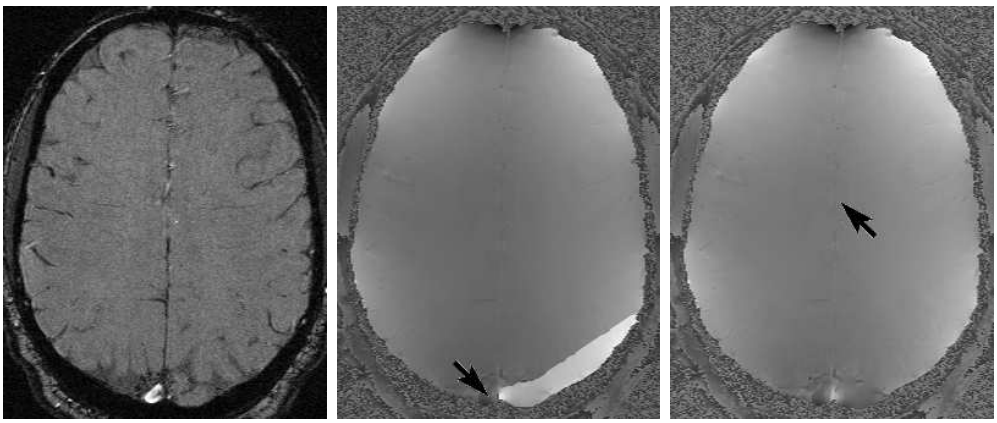


Figure 5.4: A typical magnitude (a) and phase (b) image of a data-set that suffered from strong field inhomogeneities in the frontal areas caused by the paranasal sinuses (denoted by the white rectangle). The image was unwrapped by ΦUN using Q^{LC}, Q^{ACI} and Q^{AM} to guide the unwrapping procedure. The termination criteria were optimized for each map in such a way that the unwrapping proceeded as far as possible but was limited to the object. The results are shown in the bottom row. When using Q^{LC} (c) the algorithm fails to unwrap the frontal areas. Although, Q^{ACI} (d) leads to an improved unwrapping of those regions, the best results are obtained with Q^{AM} (e).



(a) (b) (c)

Figure 5.5: Example of an artifact caused by seed points that were placed into a poorly connected region. (a) shows the magnitude image, (b) the unwrapped phase image that is affected by an artifact resulting from unwrapping starting in the sagittal sinus, (c) correctly unwrapped phase image. In case of (b) the local average of magnitude Q^{AM} was used for seed finding. (c) was obtained by applying local coherence Q^{LC} to seed finding. The arrow heads indicate the areas where seeds were placed.

The Influence of Quality Maps on Seed Finding

The map used for guiding the unwrapping is not necessarily the optimal map for seed finding. Using different maps for seed finding and unwrapping can be justified by following arguments:

- for a newly unwrapped pixel the phase topography has to be flat to allow it to pass the reliability criterion based on t_Δ as long as t_Δ is small. If a quality criterion is used for seed finding that allows seeds to be placed in areas of steep phase topography a new region starts growing only after t_Δ has been relaxed sufficiently. On the other hand it is important that, in the beginning, predictions are made using rigorous constraints so that a high reliability is ensured to avoid propagation of errors.

- since especially in the beginning the growth of regions should be as reliable as possible for as long as possible, it makes sense to base seed finding on a criterion that evaluates the data on a rather global scale. Such a global measure would have adverse effects for unwrapping, since a region should grow towards areas of poor data as far as possible.

As could be expected from these arguments the seed finding that produced the most reliable results was based on maps calculated with a kernel that was much larger than the one used for unwrapping (15×15 instead of 5×5). Q^{LC} was

found to be most suitable for seed finding, not only because it ensures that seeds are placed in areas with a flat phase topography, but also because artifacts were observed from time to time in case Q^{AM} was used. These artifacts were caused by large signal intensities in regions that were poorly connected to the bulk of the brain. Thus the brain itself was unwrapped only after all thresholds had been relaxed. Because of that errors were allowed to propagate. Figure 5.5 shows an example of such an artifact.

List of References

[1] Rauscher, A., *et al. Automated unwrapping of MR phase images applied to BOLD MR-venography at 3 Tesla*. Journal of Magnetic Resonance Imaging, **18**(2), 175–180 (2003).

[2] Xu, W. and Cumming, I. *A region-growing algorithm for InSAR phase unwrapping*. IEEE Transactions on Geoscience and Remote Sensing, **37**(1), 124–134 (1999).

[3] Witoszynskyj, S., Rauscher, A., Reichenbach, J. R., and Barth, M. *Automated phase unwrapping of MR images at different field strengths using multiple quality maps*. In *13th Annual Meeting of the ISMRM*, 2249 (2005).

[4] Witoszynskyj, S., Rauscher, A., Reichenbach, J. R., and Barth, M. *Quality maps for automated phase unwrapping of MR images*. In *European Congress of Radiology*, B–102 (2005).

[5] Conturo, T. E. and Smith, G. D. *Signal-to-noise in phase angle reconstruction: dynamic range extension using phase reference offsets*. Magnetic Resonance in Medicine, **15**(3), 420–437 (1990).

6 Implementation

This chapter describes several aspects and details of the implementation. The main design goals for the implementation were the following:

1. fast unwrapping,
2. support easy integration of further data formats,
3. reusable code in case modifications of the algorithm are added, and
4. easy integration into existing work flows.

For these reasons C was chosen as programming language. Point 2 and point 3 would have been met more easily by using an object oriented programming language such as C++. On the other hand to fulfill the last point an integration into IDL and MATLAB was considered (and also implemented). Especially in the case of IDL, according to the manuals [1], C is preferred to C++. To make use of some advantages that C++ provides, especially, late binding, the implementation makes use of pointers to functions in many cases. This allows, for example, adding support for new data types, without requiring many changes to the existing code. Also new variants of parts of the algorithm can be added easily without losing the option to use the ones implemented already.

Originally the algorithm was implemented under LINUX as a command line tool. The program was named ΦUN. It was also compiled successfully under Mac OS X and, using the Cygwin [2] environment, under Windows. After some minor modifications it was possible to compile it using the MinGW [3] environment which links against the Windows run-time libraries. Furthermore modules for IDL and MATLAB were implemented. These modules allow calling the algorithm from those two programs as if it was a built-in function.

```
typedef struct _Slice {
  Point dimension;
  BOOL phaseOnly;
  unsigned char type;
  ComplexImage slice;
} Slice;
```

Listing 6.1: Data structure for a complex valued image.

```
typedef union {
  struct {
    float re, im;
  } cart;
  struct {
    float abs, phi;
  } polar;
} Complex;
```

Listing 6.2: Data structure defining complex numbers.

6.1 Data Structures

Some important data structures that are used by various functions of the implementation are described in this section. Further data structures that are used by individual functions only, are described in the sections corresponding to these functions.

Complex Images

ΦUN uses two dimensional complex array for handling the images contained in the input data. An image is defined by the structure given in listing 6.1. The image itself is stored as a two dimensional array of complex numbers (listing 6.2). The complex numbers can be both in cartesian or polar format. The format that is used for the slice is encoded by `unsigned char type`. The advantage of allowing the image to be stored in both formats is that the number of conversions between cartesian and polar complex numbers can be reduced. ΦUN has functions not only for creating and deleting complex images, but also for creating an image consisting of polar complex numbers from one in cartesian complex numbers and vice versa. Those functions check whether a conversion is necessary and if not return a copy of the original image. This allows for a transparent use of complex images regardless of the type of the complex numbers.

```
typedef struct _UnwrappingPixel {
  unsigned char nRegions;
  float phiWrapped;
  float phiUnwrapped[MAX_OVERLAPS];
  int m[MAX_OVERLAPS];
  PRegion region[MAX_OVERLAPS];
  short unsigned int iteration;
  signed char current;
  BOOL seed;
} UnwrappingPixel;
```

Listing 6.3: Data structure for a "pixel" in the "phase image" used during unwrapping.

Phase Image for Unwrapping

The phase image used during unwrapping consists of pixels defined by listing 6.3. The rather complex structure of each pixel is caused by the fact that it is not only used for storing the phase value but also for storing information required during unwrapping. Thus it stores the number of regions by which the pixel has been unwrapped and the corresponding values for the unwrapped phase $\Phi^u_{i,j}$ and the ambiguity numbers $m_{i,j}$. The latter adds redundant information and is stored only to allow computational efficiency. For example: the unwrapped phase values are used for calculating phase predictions. Using the ambiguity number would add a significant amount of additional computation steps. On the other hand, the ambiguity numbers are used for resolving overlaps. Furthermore a pointer of the type `PRegion` is stored. This pointer points to each region by which the pixel has been unwrapped.

The variables `short unsigned iteration` and `signed char current` are used during region growing. One purpose is to make sure that only pixels are used for predictions that have been unwrapped in previous iterations. Its exact function is described in more detail in the section on unwrapping. The limited range of `iteration` is not an issue, because an absolute counting of the number of iterations is not necessary. Nevertheless the number of regions growing concurrently should be smaller than the maximum value of `iteration`. This would ensure that overflows occur every few iterations only. Otherwise the algorithm would not be able to distinguish whether a pixel was visited within the current or a previous iteration.

The purpose of `BOOL seed` is to allow for a mask that ensures that seed points are not placed too close to each other. Every time a seed point is planted a certain neighborhood around it is masked to prevent other seed points to be placed in its vicinity.

```
typedef struct _Region {
  int index;
  Fringe* unwrapFringe;
  Point leftLower, rightUpper;
  Overlap* overlaps;
  BOOL toBeDeleted;
  BOOL newOverlap;
} Region;
```

Listing 6.4: Data structure that defines a region.

```
typedef struct _Fringe {
  Point pixel;
  PFringe next;
} Fringe;
```

Listing 6.5: Data structure that defines a region.

Regions

Each region is described by the data structure given in listing 6.4. Besides the index that identifies the region, it contains pointers to the data structures containing the fringe around the region and to the overlaps with other regions, respectively. The flag `toBeDeleted` states whether the region is to be deleted because it has been merged with another region. The flag `newOverlap` is set every time an overlap with another region occurs. Its purpose is to aid an efficient resolving of overlaps. Also, the intention behind the bounding box of the region defined by `Point leftLower, rightUpper` is to allow an efficient merging of regions.

The fringe around a region is stored as single chained list. Its definition is given in listing 6.5. The reason why coordinates and not a pointer to the pixel are stored, is that the coordinates are also required for querying the neighbors.

The overlaps between regions are difficult to handle. There are many data structures one can devise to treat overlaps. Each have their own advantages and disadvantages. One possibility, for example, would be to keep the overlaps between every two regions in a separate list. This would make handling these overlaps during merging extremely efficient. On the other hand, adding new pixels to an overlap during region growing would be more expensive, since it would require looking up the correct list.

The solution that was used in this implementation is more efficient during the region growing step, but more care has to be taken during region merging. Each region has a list of all overlapping pixels. During region growing new overlaps are just added to the list. The disadvantage is that before regions can be merged, the overlaps have to be sorted. The details of how overlaps are handled are discussed later in a separate section.

```
typedef struct _Regions {
  int nRegions, maxRegions, maxIndex;
  Region** regions;
  Seeds* seeds;
  BOOL deleteRegions;
  BOOL newOverlaps;
  BOOL randM;
  int defaultM;
} Regions;
```

Listing 6.6: Data structure containing all information on the unwrapping regions.

An additional structure was implemented that stores all information that is related to the unwrapping regions. The definition of this structure is given in listing 6.6. `int nRegions` and `maxRegions` are the numbers of regions that are in use and the maximum number of regions that are allowed to grow concurrently, respectively. `int maxIndex` is the identifier used by the region added latest. `Region** region` is an array with pointers to the regions. `Seeds* seeds` contains a structure containing all seeds that are available. A description of the data structure used for storing seeds can be found in the section on seed finding. The flags `BOOL deleteRegions` and `BOOL newOverlaps` are used for resolving overlaps. The first tells whether there are any regions to be deleted after an attempt to merge regions was made, the second is set every time an overlap occurs during region growing. Finally, `int defaultM` and `BOOL randM` are used during the creation of new regions. They contain information on whether each region is to be started with the same multiple of 2π or if this number is to be chosen randomly.

6.2 Data Input and Output

The input and output functions of ΦUN were designed with having a large flexibility in mind. The flexibility is not only in the sense of supported data formats, but also in the sense of allowing an easy addition of further data formats. Both input and output is performed slice by slice.

Input

ΦUN supports three dimensional complex images and phase images for which no magnitude information is available. The complex images can be provided to ΦUN both as cartesian complex numbers and as polar complex numbers. Furthermore the complex data can be stored in a single file consisting of complex numbers or in separate files that contain real and imaginary part, or modulus

and phase, respectively. In all cases, the data has to be stored as single precision floating point values. Data files can either be in ANALYZE format [4] or be plain three dimensional data arrays that use following convention: the x coordinate represents the fastest changing index and the z coordinate the slowest.

While, in principle, data in cartesian and polar complex numbers can be handled in the same way, as long as a conversion can be performed afterwards, handling pure phase images, data stored in a single file or in two separate files requires significantly different treatment. To allow for the highest possible flexibility, following strategy was conceived: for each type of data, phase images without magnitude information, complex images stored in a single file, complex images that are stored in separate files, there is one function for opening the files as well as initializing data structures required by the other functions, one function for reading a single slice, and one function for closing the data files. While the functions for opening the various data types do not have a common interface definition, since they require, for example, a different number of file names, the functions for reading a slice and closing the files share a common interface definition. They are called through pointers to functions. These pointers are set by the functions for opening the files. Thus adding a new type of data requires only changes at a single spot in the program. Reading data and closing files is done transparently.

Since the only difference between data stored in ANALYZE format and data stored in a plain array is that the ANALYZE format has an additional header file that contains, among other information, the dimensions of the array, both type of formats are handled by the same functions. If present, the ANALYZE header is read and evaluated by the function that opens the file.

In addition, ΦUN allows data to have a different endianess than the machine on which ΦUN is executed. ΦUN determines the machine's endianess automatically. The user just has to specify the data's endianess by a command line option. In case the data's endianess differs from the machine's endianess, ΦUN performs the necessary conversion. If the data's endianess is not specified by the user, ΦUN assumes that the data has the same endianess as the machine.

Output

Compared to the data input the output is relatively simple. There is only one function for opening the file, writing a slice and closing the file, respectively. In the current implementation the unwrapped phase image is saved as float only. Since supporting other data format was of little interest, the functions are called directly without the use of pointers to functions. If the input files were in ANALYZE format, an ANALYZE header is written upon creation of the data file. In case the user requested the data to be written with a certain endianess, byte swapping is performed if necessary.

The function that actually writes a slice to disk loops over all pixels of the unwrapped phase image. For each pixel the function checks whether the pixel

has been unwrapped by one or more regions. If it has been unwrapped by more than one region, the phase of the region with the lowest identifier number is stored.

If a pixel has not been unwrapped, the original phase value, zero, or *Not A Number*, is saved depending on the choice of the user. Per default the wrapped phase value is stored. The motivation behind storing pixels that have not been unwrapped as zero or as *Not A Number* was that this allows for a mask to be created easily from the unwrapped phase image. This is especially true if *Not A Number* is used. The option of writing zeros was implemented because not all programs that handle ANALYZE files might be able to handle *Not A Number* values.

In addition to the functions for storing the unwrapped phase image, functions for storing quality maps were written. Also in this case there is a separate function for opening the file, writing a slice, and finally closing the file.

6.3 Quality Maps

All five quality maps that were described in the previous chapter were implemented. For each quality map a separate function was written. Nevertheless all five functions use the same interface and thus can be used in a transparent way by using a pointer to the respective function. The quality maps are stored in the data structure `QualityMap` that describes a two dimensional float array. The interface of the functions is defined as

```
QualityMap* calculateMap(Slice* slice, Slice* polarSlice,
                         Point kernel)
```

All functions take a pointer to both a slice that can contain cartesian complex data as well as one that can contain polar complex data. The motivation was to avoid multiple conversion between cartesian and polar complex number while keeping a common interface to all functions at the same time. Nevertheless it is not necessary to provide both data, if the data format required by the function is missing a conversion is performed. An example for this strategy would be that the original data is made up of cartesian complex numbers. Prior to unwrapping, the data has to be converted to polar complex numbers. Thus the data exists in both forms. Since the function for calculating the quality map is called through a pointer that is set to a certain function, the function calling the quality map function is not aware whether cartesian or polar complex numbers will be used for computing the quality map. In case of computing the average of the complex image Q^{ACI} the calculation would be done using data in cartesian numbers, in case of the local coherence Q^{LC} the polar form would be used. By providing both type of the data, no additional conversion has to be made. If, on the other hand, the original data consists of polar complex number, a conversion will be made only if Q^{ACI} is called, since in this case the second pointer will point to `NULL`.

The computation of all five quality criteria requires at least one integration over a rectangular neighborhood around each pixel (in case of the variance maps two such summations are required since the mean value has to be computed first). Evaluating the double sum for each pixel is rather computationally expensive. If the image dimensions are $N_x \times N_y$ and the kernel size is $k_x \times k_y$, a total number of $N_x \times N_y \times k_x \times k_y$ additions are required. For an image the size of a typical SWI slice (512×384) and a typical kernel size (5×5), this procedure would result in approximately 5 million additions. This number can be reduced easily if the sum is rewritten. This shall be illustrated first for the one dimensional case.

For a one dimensional image p_i with $i \in [1, N_x]$ and a smoothing window of k_x pixel (with k_x being an odd number) the value of a pixel $\bar{p}_i$ in the smoothed image can be written as

$$\bar{p}_i = \frac{1}{k_x} \sum_{l=i-\lfloor \frac{k_x}{2} \rfloor}^{i+\lfloor \frac{k_x}{2} \rfloor} p_l = \frac{1}{k_x} \left(\sum_{l=1}^{i+\lfloor \frac{k_x}{2} \rfloor} p_l - \sum_{l=1}^{i-\lfloor \frac{k_x}{2} \rfloor} p_l \right) = \frac{1}{k_x} \left(P_{i+\lfloor \frac{k_x}{2} \rfloor} - P_{i-\lfloor \frac{k_x}{2} \rfloor} \right) . \tag{6.1}$$

If P_i is computed and stored before $\bar{p}_i$ is calculated, the simple sequence $P_i = P_{i-1} + p_i$ can be utilized. Thus the computational effort is reduced from $N_x \times k_x$ to $2N_x$.

For a two dimensional image the reduction in computation time is more significant. Let $p_{i,j}$ be the value of a pixel at $i \in [1, N_x]$ and $j \in [1, N_y]$. The smoothing kernel shall have a dimension of $k_x \times k_y$. $\bar{p}_{i,j}$ is then given by

$$\begin{aligned}
\bar{p}_{i,j} &= \frac{1}{k_x k_y} \sum_{l=i-\lfloor \frac{k_x}{2} \rfloor}^{i+\lfloor \frac{k_x}{2} \rfloor} \sum_{m=j-\lfloor \frac{k_y}{2} \rfloor}^{j+\lfloor \frac{k_y}{2} \rfloor} p_{l,m} = \\
&= \frac{1}{k_x k_y} \Bigg(\sum_{l=1}^{i+\lfloor \frac{k_x}{2} \rfloor} \sum_{m=1}^{j+\lfloor \frac{k_y}{2} \rfloor} p_{l,m} - \sum_{l=1}^{i-\lfloor \frac{k_x}{2} \rfloor} \sum_{m=1}^{j+\lfloor \frac{k_y}{2} \rfloor} p_{l,m} - \sum_{l=1}^{i+\lfloor \frac{k_x}{2} \rfloor} \sum_{m=1}^{j-\lfloor \frac{k_y}{2} \rfloor} p_{l,m} + \\
&\qquad + \sum_{l=1}^{i-\lfloor \frac{k_x}{2} \rfloor} \sum_{m=1}^{j-\lfloor \frac{k_y}{2} \rfloor} p_{l,m} \Bigg) = \\
&= \frac{1}{k_x k_y} \left(P_{i+\lfloor \frac{k_x}{2} \rfloor, j+\lfloor \frac{k_y}{2} \rfloor} - P_{i-\lfloor \frac{k_x}{2} \rfloor, j+\lfloor \frac{k_y}{2} \rfloor} - P_{i+\lfloor \frac{k_x}{2} \rfloor, j-\lfloor \frac{k_y}{2} \rfloor} + P_{i-\lfloor \frac{k_x}{2} \rfloor, j-\lfloor \frac{k_y}{2} \rfloor} \right)
\end{aligned} \tag{6.2}$$

Figure 6.1 provides an illustration of the area integrals $P_{i+\lfloor \frac{k_x}{2} \rfloor, j+\lfloor \frac{k_y}{2} \rfloor}$, $P_{i+\lfloor \frac{k_x}{2} \rfloor, j-\lfloor \frac{k_y}{2} \rfloor}$, $P_{i-\lfloor \frac{k_x}{2} \rfloor, j+\lfloor \frac{k_y}{2} \rfloor}$ and $P_{i-\lfloor \frac{k_x}{2} \rfloor, j-\lfloor \frac{k_y}{2} \rfloor}$. Just as in the one dimensional case $P_{i,j}$ can be calculating using a sequence:

$$P_{i,j} = P_{i-1,j} + P_{i,j-1} - P_{i-1,j-1} + p_{i,j} . \tag{6.3}$$

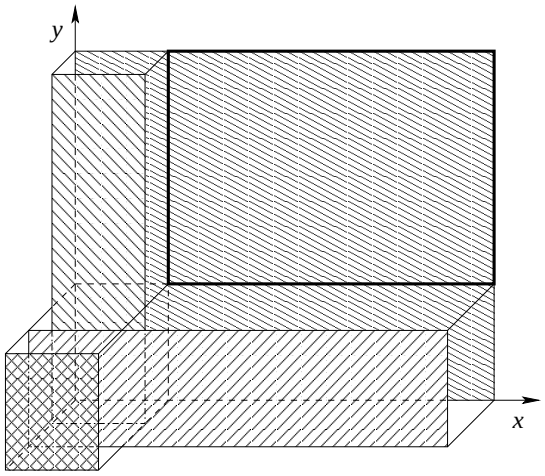


Figure 6.1: Illustration of the area integrals used for the fast integration over a sliding rectangular kernel (area enclosed by the stronger lines).

This reduces the total number of iterations from $N_x \times N_y \times k_x \times k_y$ to $2 \times N_x \times N_y$. In the case of the previous examples, this means less than 400 thousand iterations instead of just below 5 million are required.

This procedure potentially is numerically unstable since it involves subtracting large numbers that might differ by a small amount only. Double precision numbers were thus used for the computation of the sequence. The results obtained with the functions for computing the quality maps were compared to maps calculated with the built in functions of IDL on several data sets. The observed difference between the two methods were at maximum at the order of 10^{-6} and thus found to be negligible.

Special care has to be taken at the boundaries. In principle, there are two possible boundary conditions:

1. periodic boundary condition: although assuming a periodic boundary condition would be unusual in case of ordinary images, it can make sense in case of MR images, since the image is computed by a discrete FOURIER transform and is thus inherently periodic.

2. embedding the image into noise: this would correspond to extending the field of view. This boundary condition is easier to implement.

The second option was preferred for treating the boundaries, because in general the field of view is placed such that it is not too close to the boundaries to avoid aliasing due to a too small field of view. Furthermore this method is easier to implement. Nevertheless two cases have to be distinguished, namely the maps that are based on the mean value (average of magnitude Q^{AM}, local coherence Q^{LC} and average of the complex image Q^{ACI}, respectively) and those based on the variance (variance of the phase Q^{VPH} and variance of the complex image Q^{VCI}).

In the first case, one can simply add zeros around the image, since the mean value of perfectly GAUSSIAN distributed noise is zero. This ignores that the noise in the magnitude is distributed according to a RICIAN distribution. But since this just results in a decrease of the quality measure close to the border this is acceptable. In the implementation the embedding of the image into "perfect noise" is achieved by adding a rim filled with zeros is around the image. The integration starts at the origin of the original image (which is $(\lceil \frac{k_x}{2} \rceil, \lceil \frac{k_y}{2} \rceil)$ because of the additional border containing "pure noise") and proceeds to the other corner without any special treatment of the borders.

In case of the variance maps, simply adding zeros around the image would lead to a lower variance than in the noisy part of the image. Thus periodic boundary conditions were implemented for the variance maps.

All maps were normalized to the interval $[0, 1]$. In case of the variance maps, the lowest variance was defined to be 1 and the highest to be 0.

In addition to the five functions for computing the quality maps, helper functions for allocating and deallocating two dimensional double precision real valued and complex valued maps were implemented. Additionally, a function for multiplying a quality map with a bit mask was written as well. By applying a bit mask to a quality map unwrapping can be limited elegantly to arbitrary regions defined by the user without requiring additional tests during unwrapping.

6.4 Seed Finding

Every time unwrapping by a new region is to be started a seed point for this region has to be defined. In case of the single region mode this occurs only once. In the multi region mode a new seed point has to be selected every time two regions have been merged. Since choosing a seed point requires searching the quality map for the point that has the highest quality measure of all points that have not been unwrapped so far, this procedure is rather time consuming. I thus decided to identify and sort all possible seed points before unwrapping starts and store them in a special data structure. During unwrapping, every time a new region is started a seed point is picked from the top of this structure. The description of the implementation of my approach for seed finding starts with seed finding in single region mode. Afterwards seed finding in multi region mode and the data structure for storing the seed points are discussed. The functions for both modes use the same interface definition. In principle, this is not necessary, since if the multi region mode is used for unwrapping seed finding has also to be done in multi region mode and the other way round. Nevertheless using the same interface definition does not come at an additional cost and increases the reusability of the code in case new unwrapping algorithms are to be implemented in future versions of ΦUN. The template for the interface of the seed finding functions is:

```
int findSeeds(QualityMap* map, Regions* regions,
              BitMask* mask)
```

```
typedef struct _Seed* PSeed;

typedef struct _Seed {
  float quality;
  Point coordinates;
  PSeed next;
} Seed;
```

Listing 6.7: Single chained list for storing seed points.

```
typedef struct _Seeds {
  Seed** seeds;
  int* nSeeds;
  int nSeedsTotal;
  float minQuality, delta;
  int hashSize;
  int maxSorted;
} Seeds;
```

Listing 6.8: Structure for storing seed points.

The functions return the number of seed points found. The data structure containing the seed points is stored in the same structure as the regions. In addition, a pointer to a bit mask can be passed to the function. By defining a bit mask additional arbitrary constraints on the seed points (e.g. a minimum signal intensity) can be defined easily. This can be necessary if the same quality map is to be used for seed finding and unwrapping.

In addition to the two functions for seed finding, a function creating a new region by picking a seed point from the list of seed points and a function for deallocating the data structure containing the seed points were implemented.

Seed Finding in Single Region Mode

Seed finding in single region mode is straight forward and is just described for the sake of completeness. The full quality map is searched for the point having the maximum quality measure. If a bit mask has been provided, only points for which the corresponding bit in the mask is set are considered during this search. The point that has been identified as seed point is then stored in the structure for seed points (which is described in more detail in the next subsection) and returned as part of the structure describing the regions.

Seed Finding in Multi Region Mode

In principle seed finding for the multi region mode can be done by simply sorting all pixels according to their respective quality measure. Because of the large number of pixels this would be a rather time consuming and inefficient method. The main idea of the approach used by ΦUN is to minimize the actual sorting by dividing the seeds into bins of certain quality intervals. Thus each bin only contains a relatively small number of pixels and sorting those pixels can be done more efficiently. The seeds are stored in a single chained list within each bin (listing 6.7). The structure in which all seed points are stored is given in listing 6.8. `Seed** seeds` represents an array of pointers to the lists containing the seed points and thus the bins. The number of points stored in each bin is given by `int* nSeeds`. The width of each bin is given by `float delta`. The number of bins is stored in `int hashSize`. It is computed by

$$\texttt{hashSize} = \frac{1 - \texttt{minQuality}}{\texttt{delta}} \tag{6.4}$$

During seed finding the index of the bin corresponding to the pixels quality is calculated for each point that qualifies as a seed, i.e. that has a quality larger than `float minQuality` and a corresponding bit set in the bit mask (if a bit mask was passed to the function). The point is then added to this bin. In principle, all seed points within each bin should be sorted. This can be a very time consuming procedure if many points fall into a single bin. On the other hand, the seed points do not have to be sorted precisely as long as the bins are small enough. I have thus chosen an approach in which only the points having the highest quality in each bin are sorted, while the rest are just added to the list. The number of points that are sorted within each bin is given by `int maxSorted`. Thus adding a seed point to a bin requires at maximum `int maxSorted` comparisons.

Creating a New Region

Every time one or more regions have to be created a call to

```
int addRegions(Regions* regions, float t_q,
               UnwrappingImage* image)
```

is made. The functions attempts to add as many regions as are missing to the maximum number of regions that are used for simultaneously unwrapping the image. Figure 6.2 illustrates the sequence of steps for adding regions. The following steps are repeated as long as the number of regions is lower than the maximum number of regions used for unwrapping and as long as seed points are available that are above the current quality threshold:

- check whether the seed point has been unwrapped already; if not, proceed,

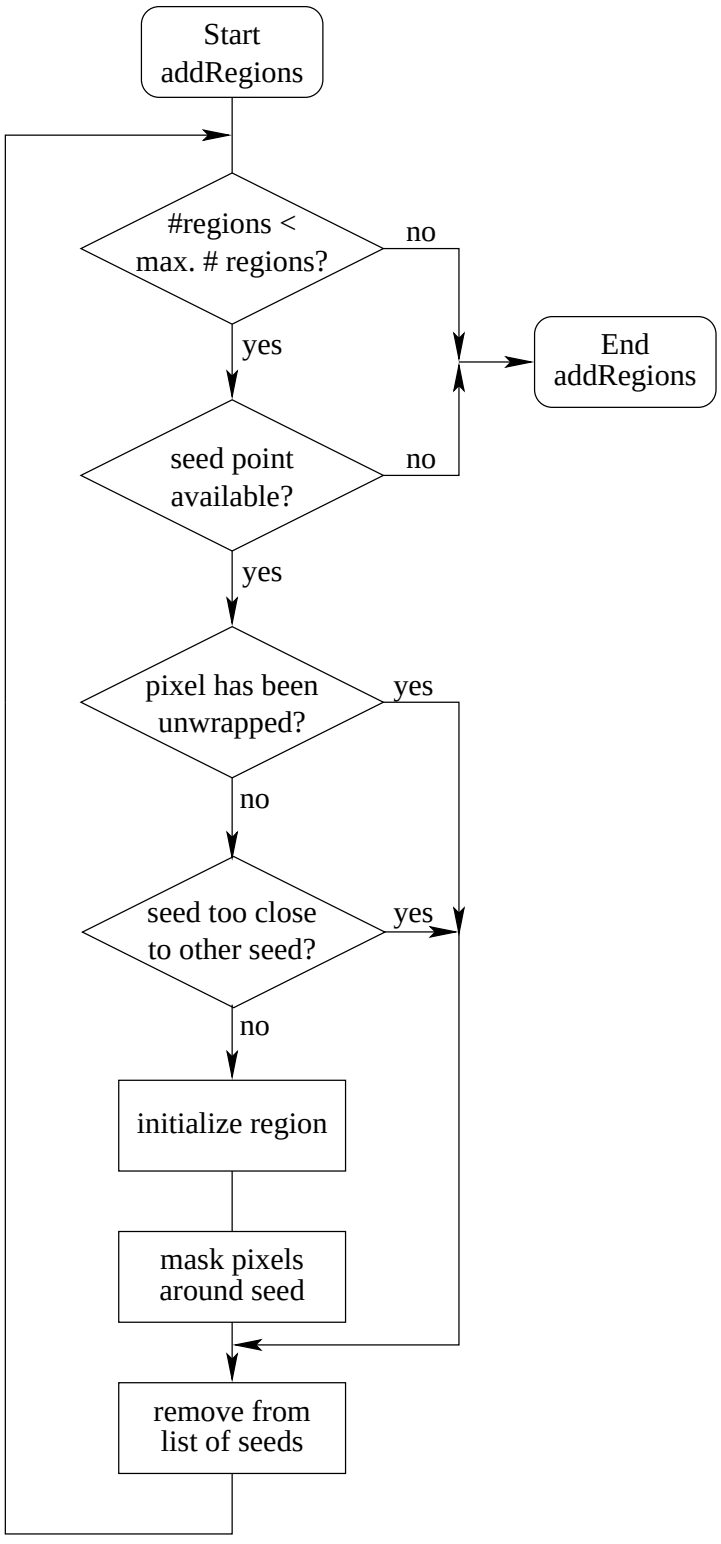


Figure 6.2: Flow chart of the function `addRegions`.

- check whether the seed point is too close to another seed point, since placing the seed points in the vicinity of another seed point would cause attempts to merge the regions to be made before the regions have actually grown. To avoid seeds being placed too close to each other the pixels around a new region are masked. During this step, it is checked whether the current seed point is masked already; if not, proceed,

- initialize the region. The structure describing the region is created and added to the list of regions. The pixel is marked as unwrapped in the unwrapped image.

- mask the pixels in the vicinity of the seed point, to avoid placing further regions.

- remove the seed point from the list of seed points.

6.5 Unwrapping

A single function controls unwrapping for both single and multi region mode. Nevertheless, the function is called through a pointer to a function to allow for a transparent replacement in case a modified approach to relaxing reliability criteria is implemented in a future release of ΦUN. The interface of the function is

```
int unwrap(UnwrappingImage* slice, QualityMap* map,
           QualityMap* seedMap, BitMask* seedMask)
```

The arguments are a phase image with the corresponding unwrapping records, two pointers to quality maps, one for seed finding and the other as quality criterion for unwrapping. Additionally a pointer to a bit mask for constraining seed finding can be given. The motivation for the bit mask is that there might be situations in which the same quality map is used for seed finding and for region growing. In this case limiting the placement of seeds by multiplying the quality map with a mask without also constraining the unwrapping would not be possible.

After initializing the data structure containing the data of the regions and calling the function for seed finding, the function starts executing the two loops for relaxing the quality criteria. The outer loop relaxes the two constraints on the extrapolation (t_v and $t_{\Delta\phi}$) while keeping both at the same value, the inner loop relaxes the quality criterion t_Q. The second loop contains a third loop, which is repeated until the criterion for relaxing a reliability criterion is reached (i.e. the number of pixels unwrapped during an iteration falls below a limit t_N). During each iteration, the algorithm checks first whether the current number of regions is equal to the desired number of regions. If this is not the case new regions are created by calling the respective function.

As a next step the function for growing a region is called for each region. This function returns the number of pixels that have been unwrapped. This value is added to the total number of pixels unwrapped during the current iteration.

After all regions have been grown, the flag indicating new overlaps is checked. If an overlap occurred the function for resolving overlaps is called. This function returns the number of pixels that had to be deleted because they were unwrapped by two regions but for which the predictions of the two regions were in disagreement. This number is subtracted from the total number of pixels unwrapped during the iteration. This is necessary to avoid oscillations and to ensure that the algorithm terminates. Finally the condition for relaxing one of the reliability criteria is checked.

6.6 Region Growing

The actual region growing is done by two functions, one for the single region mode and one for the multi region mode. Both functions are very similar and use the same interface definition:

```
int growRegion(Regions* regions, int regionsIndex,
               UnwrappingImage* image, float t_q,
               float t_d, float t_v)
```

The only difference is that handling of multiple regions is not implemented in the function for the single region mode for reasons of computational efficiency. Thus the implementation of the region growing function for single region mode is described first. Afterwards only the differences required for handling multiple regions are discussed.

Region Growing in Single Region Mode

After retrieving the pointer to the region that is to be grown from the data structure storing information on all regions using the region's index, the iteration counter is increased. The main purpose of this counter is to enable the function to discriminate between pixels that have been unwrapped during previous iterations from pixels that have been unwrapped within the same iteration.

The algorithm then starts examining each pixel stored in the fringe list. An optimization of this phase was attempted by determining as early as possible whether a pixel can be unwrapped and by trying to minimize operations by utilizing information from previous steps. As a first step the pixel is marked as having been looked at by setting the value of its `iteration` field to the current iteration and the `current` field to -2. Then the quality criterion is checked. If the pixel does not pass this test the function continues with the next pixel on the list.

Prior to investigating each neighbor, the variables for computing the prediction of the unwrapped phase are initialized. These variables are the common

prediction and the sum of the weights, the number of prediction lines, as well as an array for the individual predictions and another array for the respective weights.

The algorithm checks whether a neighbor has been unwrapped in a loop over all neighbors and immediately proceeds to the next neighbor if this is not the case. If a neighbor has been unwrapped it checks if the pixel has been visited by comparing the `iteration` field with the current iteration. If necessary the pixel is then marked as visited by updating the `iteration` field and setting `current` to 0. These steps are repeated with the next nearest neighbor on the prediction line defined by the neighbor. The individual prediction is then computed taking into account whether the next nearest neighbor belongs to the region or not. Both, the prediction and the weight, are stored in the respective arrays and added to the corresponding total values. Furthermore the counter for the number of neighbors is increased by one.

If the number of neighbors is larger than zero the algorithm tests the other reliability criteria. Actually the situation that a pixel in the fringe has no neighbors should not occur, since a pixel without any unwrapped neighbors is not really a part of the fringe. Such a pixel is thus removed from the fringe list and added to a list containing all pixels that were removed from the fringe. The memory for data structures on this list is deallocated before the function exits.

The reliability criteria are tested in the following manner: first a check whether pixel passes $t_{\Delta\phi}$ is performed. This test is done first, because it does not require any further calculations. Contrary to that computing the variability of the predictions requires a loop over the individual predictions of each prediction line. Thus the efficiency is increased if this parameter is computed and tested only if the pixel has passed the other reliability criteria.

In case the pixel has passed all reliability criteria and thus can be unwrapped, the corresponding fields in the pixel's data structure are adjusted. These fields are the counter `nRegions` which counts by how many regions the pixel has been unwrapped, the unwrapped phase `phiUnwrapped[]`, the ambiguity number `m[]` and the pointer to the region `region[]`. Furthermore the field `current[]` is decreased by `nRegions`. This way the pixel is marked as having been unwrapped within the current iteration. The pixel is then removed from the fringe list and added to a list of pixels that have been unwrapped during the current iteration. This list is used later to determine which pixels have to be added to the fringe because they are neighboring a newly unwrapped pixel.

After all pixels of the fringe have been processed, the bounding box of the region is adjusted if necessary and the fringe is updated. This is done by iterating over the list of newly unwrapped pixels. For each pixel its immediate neighbors are tested exploiting the information of the `current` and `iteration` fields: a pixel for which `current` is set to -2 is already in the fringe, either because it was not unwrapped, or because it did not pass the reliability criteria, or because it just has been added to the list. Latter happens when a given pixel is neighboring two or more pixels that have been unwrapped during the current iteration. If

the value of `current` is -1 the pixel is a neighbor that is not in the fringe list already. It thus has to be added to fringe. After this is done, its `current` field is set to -2 to avoid adding it multiple times.

Finally the list of pixels that do not belong to the fringe anymore is deleted and the associated memory is freed.

Region Growing in Multi Region Mode

The function for region growing in multi region mode is implemented almost exactly as the function for single region mode. The only difference is that it takes into account that if a pixel has been unwrapped already it has not necessarily been unwrapped by the current region and thus an overlap might occur. Such an overlap has to be taken care of in a special way. To handle multiple regions efficiently the function uses the field `current` even more than the function for single region mode.

First of all one has to distinguish five cases:

1. a pixel has been unwrapped by two regions already and is part of the current region's fringe,
2. a pixel has not been unwrapped by any region, but is part of the fringe of the current region,
3. a pixel has been unwrapped by another region and is part of the fringe of the current region,
4. a pixel has been unwrapped by the current region only, and finally
5. a pixel has been unwrapped by the current region and another region.

In the first case the function proceeds immediately to the next pixel in the fringe, because overlaps are allowed between two regions only. The algorithm will attempt to unwrap this pixel in a later iteration (after the other two regions have been merged).

The second situation is trivial and does not differ from the single region mode. In the third case the algorithm has to ensure that it handles the overlap correctly when the pixel is being unwrapped. Unwrapping of a pixel in this case results in the field `nRegions` of the pixel now being increased to two and the unwrapped phase, the ambiguity number as well as the pointer to the region being stored in the second field of the corresponding arrays. Furthermore two overlap records are created. One pointing to the region which unwrapped the pixel first which is added to the current region's list of overlapping pixels. And one pointing to the current region which is added to the other region's list.

The last two situations occur either when the pixel actually should not be on the list of fringe pixels (which usually occurs after two regions have been

merged) or when the pixel is a neighbor of a pixel that is to be unwrapped. Of course, the second case is the one of interest.

The main problem in all cases is the very often repeated check whether a pixel has been unwrapped by the current region, and if this is the case, which field in the array containing the value of the unwrapped phase corresponds to the current region. Usually it has to be performed more than once for each pixel, because pixels in the fringe have common neighbors and every time a neighbor is being looked at the check has to be done. Thus an efficient implementation that reduces the number of tests to a minimum is desirable.

To achieve this the algorithm uses the fields `iteration` and `current` in the following way: just as before `iteration` and `current` are set to the number of the current iteration and to -2, respectively, as soon as the function starts processing a pixel belonging to the fringe. Then, during the treatment of this pixel's neighbors, the algorithm checks whether the neighbor has been examined during the current iteration by using the `iteration` field. If this is not the case, it updates the `iteration` field and determines whether the pixel has been unwrapped by the current region. This is done by comparing all entries in the pixel's `region` field with the pointer to the current region. If there is a match the corresponding index is stored in the `current` field, otherwise it is set to -1. Thus the values of `current` have the following meaning:

0 and positive values indicate the index of the pixel's unwrapping record that corresponds to the current region,

-1 the pixel has not been unwrapped by the current region and is not in the fringe,

-2 the pixel belongs to the fringe.

Thus for using the neighbors for a prediction only a test whether `current` is positive or not has to be performed after testing `iteration`. In all subsequent steps `current` is used as an index to access the pixel's values that corresponds to the current region.

6.7 Resolving Overlaps

At the end of each region growing iteration, i.e. after the algorithm attempted to unwrap a one pixel wide fringe around each region, the algorithm tries to resolve any overlap that might have occurred. Resolving overlaps is a non-trivial, computationally expensive operation. It was thus tried to optimize it as much as possible.

As a first step the algorithm checks if any new overlap occurred by checking a flag in the regions' structure. If not the function returns immediately. Otherwise the algorithm loops over all regions starting with the region that has the lowest index. This ensures that at first the algorithm tries to merge overlaps that have

```
typedef struct _Overlap {
  Point coordinates;
  UnwrappingPixel* pixel;
  PRegion region;
  int diff;
  POverlap prev,next;
} Overlap;
```

Listing 6.9: Data structure for overlaps.

occurred with those regions that were started with the best seed points, i.e. the "oldest" regions.

Sorting the Overlapping Pixels

First the flag whether a pixel has been added to an overlap of the current region is tested. If it is not set the algorithm continues with the next region. Otherwise the list containing the overlapping pixels is sorted. The function updates the information of the structure in listing 6.9. This is necessary in case the region with which the current region is overlapping has already been merged with another region. In this case neither `int diff` nor `PRegion region` might be correct.

The difference between the multiplication factors $m_{i,j}$ of 2π of both regions is stored in `int diff`. The list is then sorted in terms of the indices of the regions the current region is overlapping with. If many regions are growing concurrently many overlaps may occur. Thus the list might be rather long. Thus a fast sorting algorithm, namely quick sort, was implemented.

As a next step the algorithm checks each overlap if it is large enough for an attempt to merge both regions ($N_O > t_O$). If this is the case, the most common phase difference Δ^{mode} (i.e. the mode of the phase differences) is calculated.

Calculating the Most Common Phase Difference

For calculating the mode a single chained list that stores the differences between the unwrapped phases of pixels in the overlap and their frequency is used. For each pixel in the overlap, the list is searched for the respective difference between the multiplication factors $m_{i,j}$ of both regions. If the value is found its occurrence frequency is increased by one, if not this value is appended to the list. Finally the difference that has the highest occurrence frequency (i.e. the mode Δ^{mode}) is returned together with the number of occurrences.

If the ratio between the occurrence frequency of Δ^{mode} and the total number of pixels in the overlap N_O passes the threshold t_f the two regions are merged. Otherwise the pixels in the overlap are deleted from both regions.

Merging of Regions

The algorithm starts by looping over all pixels in the overlap. For each pixel it checks whether the difference between the unwrapped phases is in agreement with the most common difference (Δ^{mode}). If there is an agreement the pixel is updated such, that the reference to the second region is removed and the region counter is set to one. If the difference is not compatible with the common difference, the references to both regions are deleted. Thus the pixel is being marked as not having been unwrapped. Furthermore the pixel is added to the fringe of the first region, so that an attempt to unwrap it is made in the next iteration.[1] The number of pixels that have been set as not unwrapped is returned.

During the next step all pixels of the second region are added to the first region. This is done by looping over all pixels that are contained within the bounding box of the second region. Each pixel is then tested if it has been unwrapped by the second region. If this is the case the reference to the second region is updated such that it points to the first region. Also the value of the unwrapped phase is adjusted.

It is also necessary to add the pixels in the fringe around the second region to the fringe of the first region. This is done by simply joining the lists. A check whether a pixel of the fringe list has been unwrapped is not done, because both fringes would have to be checked. Such a check is done during the region growing phase. Also, the list of overlaps of the second region is appended to the list of overlaps of the first region.

Finally, the bounding box of the first region is updated such that it also circumscribes the pixels added to the region. The second region is marked as to be deleted.

It is necessary to sort the list of overlaps after two regions have been merged, since the overlap with any other region might have changed in size because of a previous overlap that the second region had. After sorting the overlaps, the next overlap is processed.

Deleting an Overlap

If the agreement between the two regions is not good enough, all pixels in the overlap are deleted from both regions. This is done by setting the region counter of each pixel to zero and setting the pointers to the regions to `NULL`.

After all pixels have been removed from both regions, the fringe of each region has to be updated. This is done by looping over all pixels in the overlap once again. For each pixel a check is performed whether the pixel has neighbors that belong to one of the regions. If a pixel has neighbors it is added to the fringe of that region.

[1] The pixel is not added to the second region, because the second region is being merged with the first region. It will thus not exist during the next iteration.

Twice the number of pixels that have been deleted (once for each region) is returned. Subtracting this number from the total number of pixels which were unwrapped during the previous growth phase is necessary to avoid oscillations. Such oscillations could occur if the thresholds were not relaxed because the number of pixels unwrapped appears higher than it actually is.

Finishing Resolving the Overlaps

After all regions have been processed, the algorithm checks whether there are any regions that have to be deleted because they have been merged with other regions. If this is the case, the structure corresponding to the region is removed. The array containing all regions is modified such that after a region has been deleted, the regions with a higher index are moved in order not to leave any gaps in the array. Finally, the flags marking the inclusion of pixels to an overlap and that regions have to be deleted, are both cleared.

List of References

[1] RSI. *IDL External Development Guide*, IDL Version 6.2 edition (2005).

[2] *Cygwin*. http://www.cygwin.com/.

[3] *MinGW - Minimalist GNU for Windows*. http://www.mingw.org/.

[4] *Analyze 7.5 file format*. http://www.mayo.edu/bir/PDF/ANALYZE75.pdf.

7 Evaluation

In principle, there are two possible approaches to evaluate a phase unwrapping algorithm: to unwrap experimentally measured data and compare the results with another method or a gold-standard, or to unwrap simulated phase images. Both approaches have advantages and disadvantages.

The advantage of using experimentally obtained data is that these data give a realistic picture of the algorithms performance. The disadvantage is that for this kind of data the true value of the phase is not known. The results have thus to be compared with a second method or a gold-standard. In situations in which the results of the two methods disagree it can be hard to judge which method yielded the correct phase value.

The problem of not knowing the "true phase" can be overcome if simulated phase images are used. For such a simulation one could assume that the phase is a function to which noise is added. Unfortunately, such a simulation fails to reflect the interplay between magnitude and phase image. Furthermore it does not resemble the complexities of measured data. It is rather difficult to devise a simulation that yields realistic magnitude and phase images while preserving knowledge of the absolute phase.

For these reasons I chose the first approach, namely a comparison of ΦUN's performance on experimentally measured data with an established method. As established method, I selected PRELUDE [1] which is part of the FSL Analysis Package (Oxford Centre for Functional Magnetic Resonance Imaging of the Brain, United Kingdom). PRELUDE was mainly developed for correcting EPI data for distortions caused by magnetic field inhomogeneities.

PRELUDE belongs to the class of algorithms commonly known as "Split and Merge" algorithms. During the split phase it divides the image into partitions that have similar phase values. Each region is free of phase wraps because of

the definition of the phase intervals. In the worst case, a region might contain a single voxel. PRELUDE then attempts to merge the regions using a cost function that is minimal when the correct multiple of 2π is added to each region.

Unlike ΦUN, PRELUDE is able to unwrap not only in two dimensions but also in three dimensions. Also a third mode in which the partitioning phase is done slice by slice (i.e. in two dimensions), but with the region merging phase proceeding in three dimensions exists. By default PRELUDE chooses the hybrid mode for high resolution data. All of the data presented herein were treated as high-resolution data-sets. Nevertheless, PRELUDE can be forced to use a certain mode by setting command-line parameters. For the evaluation of ΦUN, all three modes of PRELUDE were used.

For the evaluation, both programs were applied to phantom data and subject data using their default parameters. The only exception were the parameters controlling the unwrapping modes (single region and multiple regions in case of ΦUN, and 2D, hybrid and 3D mode in case of PRELUDE).

All data-sets were unwrapped on the same multi-processor LINUX machine. The computer contained two 2 GHz dual-core AMD OPTERON processors. It was ensured that one processor core was used for unwrapping only at all times. Furthermore, the machine always had enough free memory to guarantee that no swapping occurred. Also, it was ensured that all disk operations were on local disks to allow for fast and reproducible data access. The time required to read a typical complex data-set and to save it as floating point values was measured and found to negligible (6.2×10^{-3} s for a 256×224 slice).

The results obtained with the two programs were compared both in terms of time required for unwrapping and the quality of the unwrapped phase images. Differences between the results where identified by subtracting the unwrapped phase images from each other. In case of offsets between images, a multiple of 2π was subtracted from the subtraction image. Thus any pixel of the subtraction image that differed from 0 was caused by a mismatch between the methods. Those pixels were closely investigated in the magnitude image, the wrapped phase image and the unwrapped phase image of each method to obtain an understanding for the reasons for the disagreement.

The results of the evaluation were presented at the 2007 Annual Meeting of the ISMRM [2]. A manuscript has been submitted to Medical Image Analysis [3].

7.1 Phantom Data

Using phantom data for evaluating ΦUN's performance was motivated in the following way:

- a phantom produces realistic data that takes into account the full measurement process.
- nevertheless, the complexity of the data can be controlled by the design of

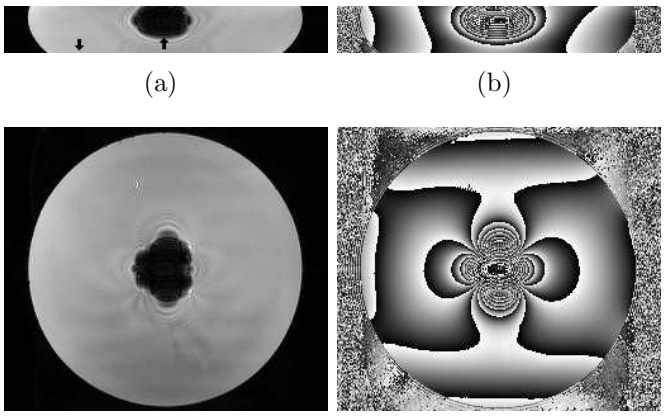


Figure 7.1: A sagittal slice of the phantom (magnitude: (a), wrapped phase: (b)) and a transversal slice of the phantom (magnitude: (c), wrapped phase: (d)). The arrow heads in (a) indicate the slices in which the ROIs for calculating the SNR were placed.

the phantom. Thus, although it is not possible to know the "true" phase of measured data, the likelihood of a certain predicted phase can be judged rather easily.

- the amount of dephasing and the signal's magnitude (and thus the SNR) can be modulated easily by the experiment's parameters.

- artifacts often found in in vivo data, such as motion or flow, can be suppressed easily.

- the total acquisition time and thus the parameter space that can be sampled is limited by the available scanner time only. Since a clinical scanner was used to acquire the data, all data had to be measured within a single night.

The goal of this study was to investigate the behaviour of ΦUN at different SNR levels, as well as how well it could handle the rather complex phase topography caused by a large susceptibility perturbation. In nature, such a perturbation can be found, for example, close to the paranasal sinuses. Furthermore, the dependence of ΦUN's performance on the density of phase wraps was of interest. The phantom was thus designed to be homogeneous with a single field perturbation. It consisted of a 5 L glass bowl with a ping-pong ball immersed in an aqueous solution containing 0.9 % NaCl and 0.2 mmol/l Gd-DTPA. Figure 7.1 shows sagittal and transversal cuts through the magnitude images (fig. 7.1(a) and fig. 7.1(c), respectively) and through the phase images (fig. 7.1(b) and fig. 7.1(d)).

Data Acquisition and Processing

A total number of 72 fully first order flow compensated 3D gradient echo [4] data-sets were acquired on a 1.5 T MR scanner (MAGNETOM VISION, SIEMENS, Erlangen, Germany) with the standard CP receive/transmit head coil. The scans had different echo-times, flip angles and resolutions. 48 of the 72 scans were low resolution scans (matrix: $256 \times 168 \times 48$ and a field of view (FoV): $256 \times 224 \times 95\,\mathrm{mm}^3$; i.e. having a voxel dimension of $1 \times 1.33 \times 2\,\mathrm{mm}^3$). TE and the flip angle α of each measurement were chosen such that all combinations of the parameter space TE $\times$ α with TE $\in \{20, 25, 30, 35, 40, 45\}$ ms and $\alpha \in \{2, 4, 6, 8, 10, 15, 20, 25\}^\circ$ were used.

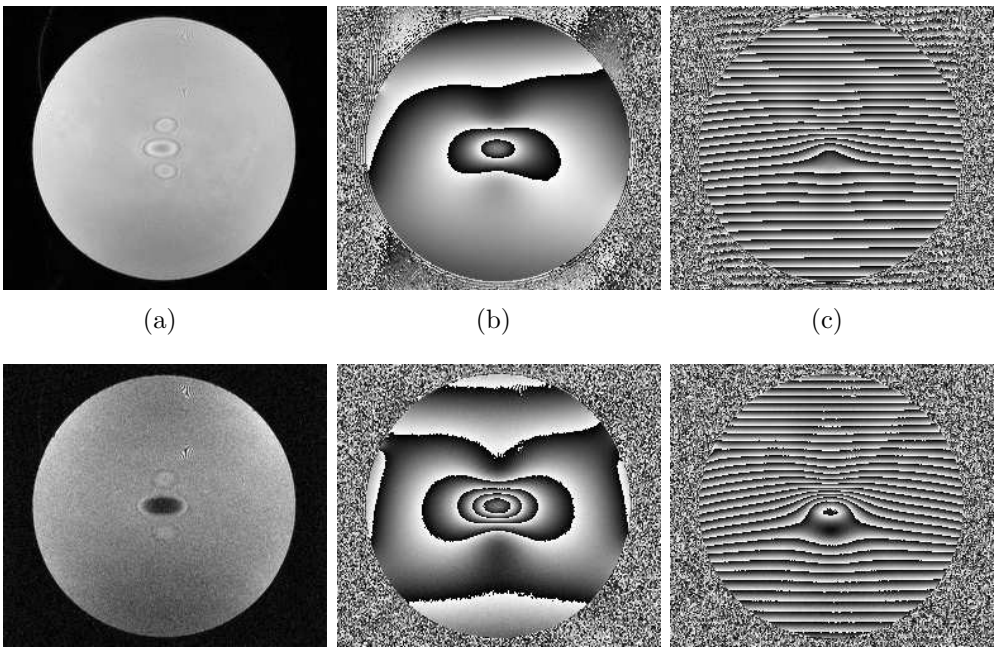
The other 24 data-sets consisted of 12 low resolution (matrix: $256 \times 168 \times 32$, FoV: $256 \times 224 \times 48\,\mathrm{mm}^3$; i.e. voxel dimensions: $1 \times 1.33 \times 1.5\,\mathrm{mm}^3$) and 12 high resolution (matrix: $512 \times 336 \times 32$, FoV: $256 \times 224 \times 48\,\mathrm{mm}^3$; i.e. voxel dimensions: $0.5 \times 0.67 \times 1.5\,\mathrm{mm}^3$) scans. The parameter space TE $\times$ α was TE $\in \{20, 30, 45\}$ ms and $\alpha \in \{2, 4, 8, 16\}^\circ$. Each low resolution and each high resolution data-set corresponded to one point in this parameters space. In z direction, the slab did not cover the full ping-pong ball. For all measurements TR was 60 ms.

Before reconstruction, the data were zero-filled in the in-plane phase encoding direction to obtain an isotropic in-plane voxel size. This step yielded an in-plane resolution of $1 \times 1\,\mathrm{mm}^2$ for the low resolution data-sets and of $0.5 \times 0.5\,\mathrm{mm}^2$ for the high resolution data-sets, respectively. While the echo was slightly off-centered in the phase encoding direction to increase the number of phase wraps and thus creating a more complex phase topography, it was centered in the z direction. The latter was done to avoid penalizing three dimensional unwrapping methods.

After reconstruction, the outermost slices were discarded. This was done for two reasons: first of all the signal in those slices was comparatively small. Secondly the images were hampered by fold overs due to violation of the NYQUIST condition in the z encoding direction caused by imperfections of the slice profile, which are typical for 3D imaging.

Additionally, to compare the performance in homogeneous and inhomogeneous regions, each data-set was split into a homogeneous part (13 slices) and one containing the inhomogeneous part (20 slices). The homogeneous data-sets did not exhibit significant distortions caused by the field inhomogeneity. In total, 216 data-sets were created and unwrapped.

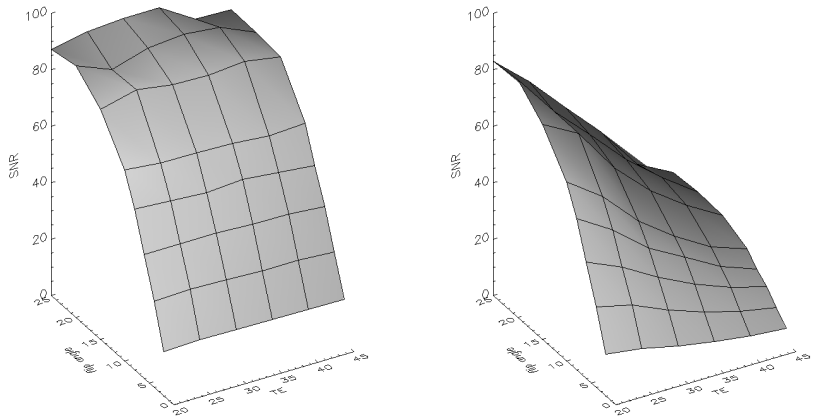
To study the performance of ΦUN and PRELUDE as a function of the number of phase wraps, two neighboring slices of three low resolution data-sets (SNR = 5.1, 14.9 and 83.0) were selected. The slices were chosen such that they were still affected by the field inhomogeneity, but that they were unwrapped by all methods completely. A two dimensional FOURIER transform was then applied to each slice. This led to a two dimensional k-space for each slice. From each k-space a set of images with different numbers of phase wraps was obtained by



(a) (b) (c)

(d) (e) (f)

Figure 7.2: Illustration of the influence of measuring parameters and echo shifts on the phantom data. The top row displays data that were acquired with TE = 20 ms and a flip angle of $\alpha = 25\,^{\circ}$. The SNR was calculated to be 80.3 in the center of the image. (a) is the magnitude image, (b) the phase image with the echo centered and (c) the phase image with the echo shifted to achieve the maximum number of phase wraps. The bottom row exhibits data of the same phantom acquired at TE = 45 ms and $\alpha = 2\,^{\circ}$. The SNR in the center of the image is 5.1. (e) and (f) are the phase images with the centered and the maximally off-centered echo, respectively.



(a) (b)

Figure 7.3: SNR as a function of echo time TE and flip-angle α of a ROI in an homogeneous area (a) and a ROI affected by the field inhomogeneity (b).

shifting the echo off center in phase encoding direction. The echo was displaced in steps of two pixels to a maximum distance of 33 pixels from the center. Depending on the displacement of the echo, the number of phase wraps ranged from 6 to 31. The median distance between wraps ranged from 22.0 to 6.5 voxels, respectively. The effect of both the off centered echo and the acquisition parameters on magnitude and phase images of the phantom is displayed in fig. 7.2. The top row corresponds to the images with the highest SNR (SNR = 80.3) and the bottom row to the images with the lowest SNR (SNR = 5.1). The middle column displays the phase images that were reconstructed with the echo centered in k-space, while the right column shows the phase images obtained with the maximally off centered echo.

Results

Low Resolution Data-Sets

The SNR was measured for each data-set in two ROIs. One ROI was placed in the vicinity of the field inhomogeneity and one in an homogeneous area. The placement of the ROIs is indicated by arrow heads in figure 7.1. In the homogeneous area the SNR ranged from 5.4 at the longest TE and the smallest flip-angle α to 90.6 at the shortest TE and the largest flip-angle α. In the inhomogeneous ROI the respective values were 5.1 and 84.3. The SNR as a function of flip angle α and TE is shown in figure 7.3 for both ROIs.

The results of all applied methods (ΦUN, ΦUN with multiple regions and PRELUDE in all three modes) were in very good agreement for all SNR levels.

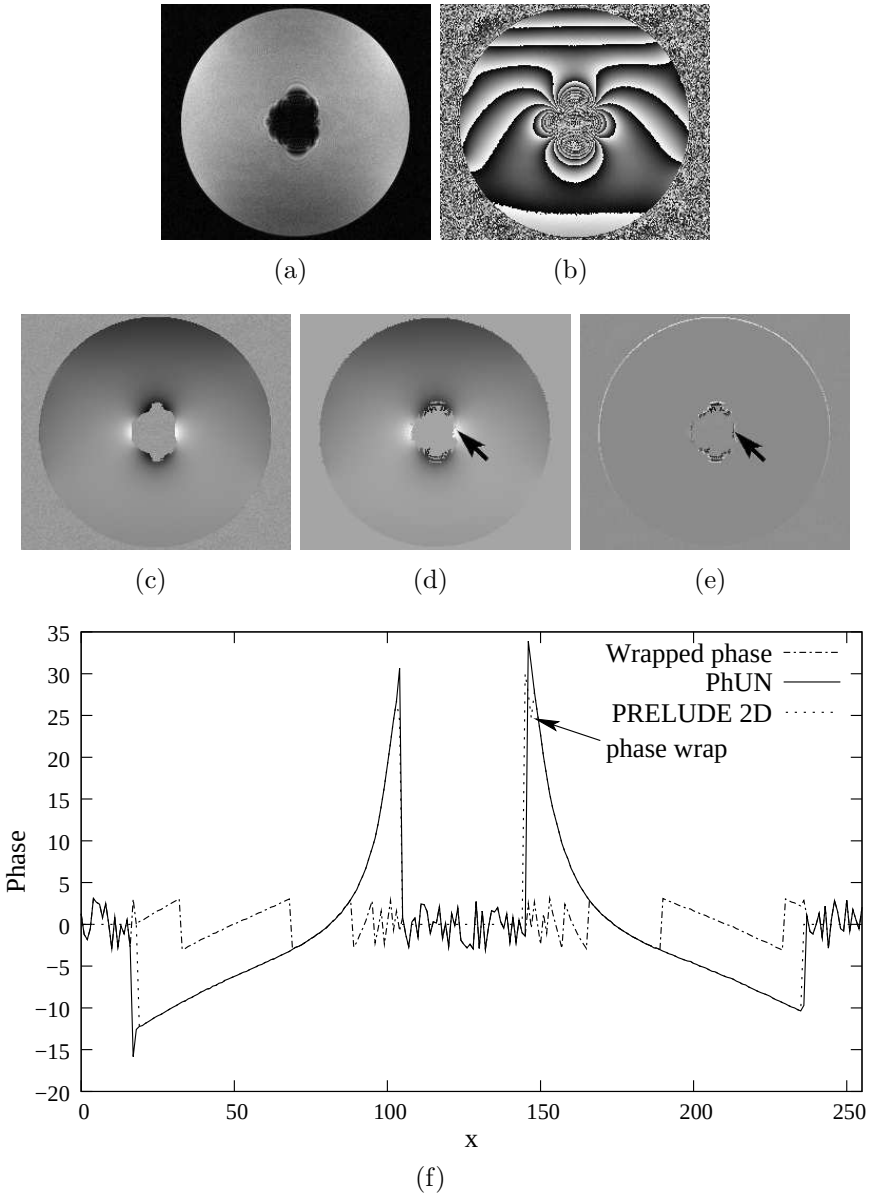
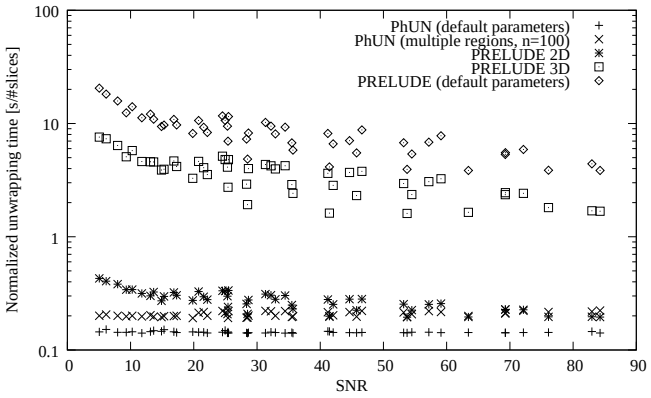
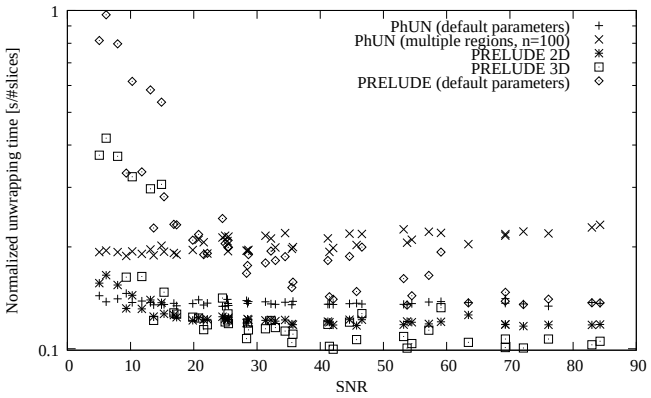


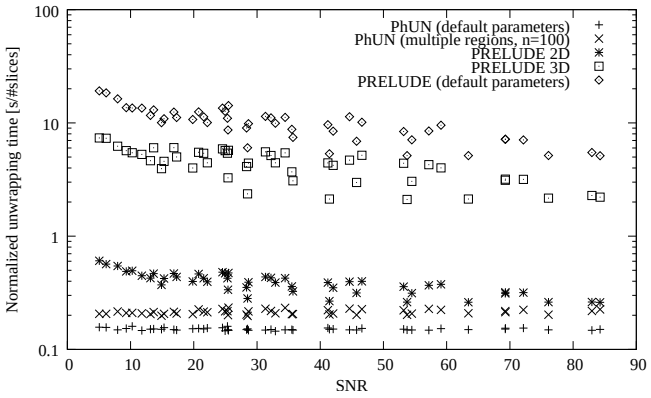
Figure 7.4: A slice of the phantom ((a) magnitude image, (b) phase image) unwrapped by ΦUN (c) and PRELUDE 2D (d). (e) shows the difference between the results obtained with ΦUN and PRELUDE. White pixels indicate pixels for which ΦUN computed an unwrapped phase but not PRELUDE, black pixels those which were unwrapped by PRELUDE but not by ΦUN. In (f) the phase along a cut parallel to the x-axis at the center of the slice is shown. At about pixel 150, PRELUDE generates an artificial phase wrap. This wrap is indicated in (d) and (e) by the arrow heads. It is hardly visible because of its small size and the rather large dynamic range of the image (the lowest value is -51.79, the highest 42.46).



(a)



(b)



(c)

Figure 7.5: Normalized time (on a logarithmic scale) required for unwrapping a slice of (a) a full low resolution data-set, (b) a slab at a distance from the source of the field inhomogeneity and (c) a slab around the field inhomogeneity versus SNR measured in a region in the proximity of the inhomogeneity's source.

Nevertheless, some small systematic differences between the two methods were observed (fig. 7.4):

- ΦUN tended to unwrap an additional one pixel wide rim compared to PRELUDE at the outer boundary of the phantom. This is the result of slightly differing termination conditions.

- in areas that were strongly affected by the signal drop out caused by the field inhomogeneity, ΦUN tended to create smoother borders, while PRELUDE unwrapped further into the field inhomogeneity, but often left small "holes" of wrapped pixels that were unwrapped by ΦUN. Also, in this case, the discrepancies can be explained by slightly differing termination criteria.

In regions with a very steep phase topography, PRELUDE occasionally created a phase wrap. These areas were small in size (well below 100 pixels). In those situations the different modes of PRELUDE often yielded differing results. ΦUN always generated smooth phase topographies in which these wraps did not appear. Nevertheless, a close inspection of all images did not provide insight into which result was the correct one. Figure 7.4(d) shows a phase image obtained with PRELUDE that exhibits such a wrap (indicated by the arrow head). This wrap is not visible in the phase image generated by ΦUN (fig. 7.4(c)). A cut through the phase image is displayed in figure 7.4(f).

Although all methods performed equally well in terms of the quality of the unwrapped phase image, the times required for unwrapping differed significantly. The computation times required by all methods versus the SNR measured in the ROI affected by the field inhomogeneity is shown in figure 7.5. The computation time is normalized to a single slice. At the highest SNR levels, ΦUN's single region mode demanded 0.14 s for unwrapping a single slice. In multi-region mode, ΦUN was about a factor 1.6 slower. PRELUDE in 2D mode needed about 0.20 s. In 3D and hybrid mode PRELUDE was more than a factor 12 slower, requiring 1.68 s and 3.85 s, respectively.

ΦUN's computation time did not show an SNR dependency. On the other hand, PRELUDE's performance in all three modes was strongly dependent on SNR. At the lowest SNR levels PRELUDE required between 2.5 times (in 2D mode) and 145 times (hybrid mode) as much time as ΦUN. Comparing the times required for unwrapping the homogeneous (fig. 7.5(b)) and the inhomogeneous (fig. 7.5(c)) part of each data-set, showed that unwrapping times required for unwrapping the full data-set were dominated by the slices most affected by the field inhomogeneity.

High Resolution Data-Sets

The SNR of the high-resolution data-sets was measured in a similar way as the one of the low-resolution data-sets. Because of the smaller voxel size the

SNR was reduced significantly. It ranged from 3.7 to 22.1 in the ROI in the homogeneous area and from 1.8 to 18.5 in the ROI in the proximity of the ping-pong ball. Unlike in the case of the low-resolution data-sets, the effects of the field inhomogeneity were not dominating the computation time anymore. PRELUDE's 3D and hybrid mode failed to unwrap low SNR (SNR < 8) data-sets within 24 h and were terminated manually. Since PRELUDE managed to unwrap the slab that contained only the field inhomogeneity, and thus had less slices, within 24 h in all cases, it is reasonable to assume that PRELUDE would have unwrapped the full low SNR data-sets if it had been given more time. Nevertheless, terminating PRELUDE of 24h was justified since the goal was not to test whether PRELUDE was able to unwrap all data-sets but an evaluation of ΦUN's performance.

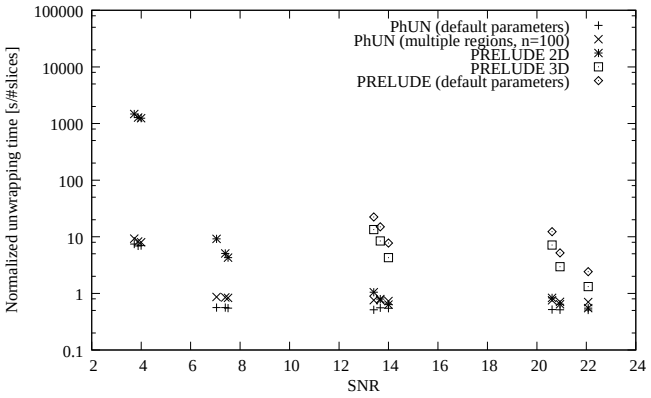
The unwrapped phase images obtained with all methods were in very good agreement above an SNR of 13. Just as in the case of the low resolution data, the only discrepancies between the methods were found in the areas most affected by the field inhomogeneity. As before ΦUN created smoother boundaries while PRELUDE grew further into the signal drop out, but left "holes" of unwrapped pixels. Except for these differences the unwrapped phases images were smooth and did not contain wrapped pixels.

At low SNR, differences between the ΦUN and PRELUDE were observed. While ΦUN still yielded smooth and fully unwrapped phase images below an SNR of 13, PRELUDE failed to unwrap an increasing number of pixels. These pixels were not connected and appeared like salt and pepper noise. The number of not unwrapped pixels increased from a single pixel in a single slice at SNR ≈ 13 to 1.5 % of the pixels of the object at the lowest SNR (SNR ≈ 3.7).

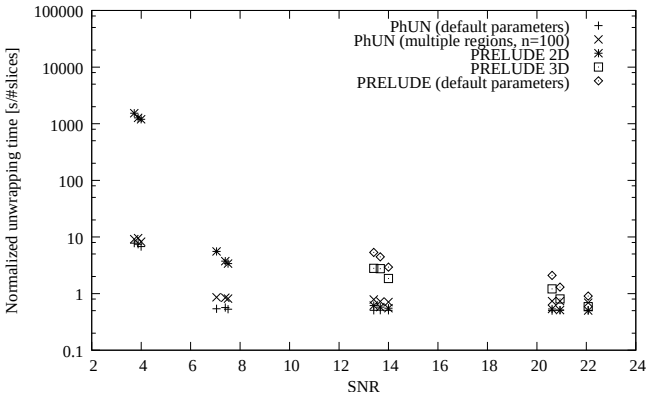
At SNR below 10, PRELUDE started attempting to unwrap the area outside of the phantom. This led to a sharp increase in computation time (a factor 10 at SNR ≈ 7 and more than a factor 1000 at SNR < 4, respectively). At SNR levels below 4, ΦUN also started to unwrap the area outside of the phantom. This resulted in an approximately 10 times longer unwrapping time (fig. 7.6). Except for the lowest SNR levels, ΦUN's unwrapping speed did not show a dependence on the SNR level. This is in agreement with the low resolution data. ΦUN required about 0.51 s for unwrapping a single slice at the highest SNR levels. This is approximately four times as long as for the low resolution data and corresponds very well to the increase in the number of pixels. The respective computation times of PRELUDE were between ≈ 0.55 s in 2D mode, ≈ 1.32 s in 3D mode and ≈ 2.42 s in hybrid mode.

Phase Wrap Density Dependence

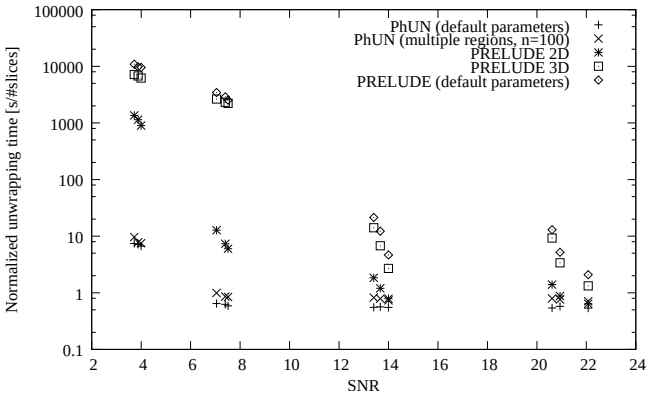
To study the dependency of ΦUN's performance on the number of phase wraps the results of ΦUN on the echo shifted data-sets were compared to those ob-



(a)



(b)



(c)

Figure 7.6: Normalized time (on a logarithmic scale) required for unwrapping a slice of: (a) the full high resolution data-set, (b) a slab at some distance from the source of the field inhomogeneity and (c) a slab around the field inhomogeneity plotted against the SNR measured in the region least affected by the inhomogeneity.

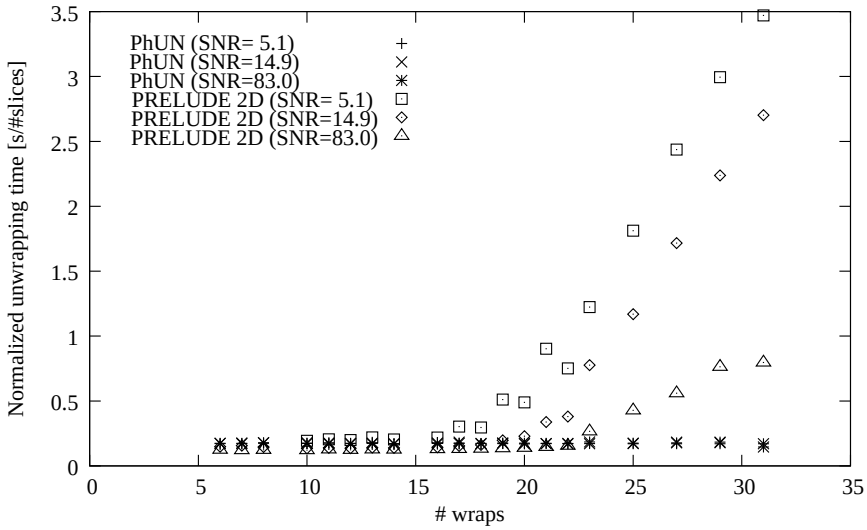


Figure 7.7: Dependency of ΦUN's and PRELUDE's (in 2D mode) performance on the number of wraps in a slice at different SNR levels. The number of wraps was modulated by shifting the echo off center before reconstruction. While ΦUN's speed is independent of the number of wraps, the time required by PRELUDE increases with increasing number of wraps.

tained with PRELUDE's 2D mode.[1] Both programs were able to unwrap all phase images. The unwrapped phase images were in exact agreement.[2] The methods differed only in the dependence of the computation time on the phase wrap density. While the unwrapping times of all methods were comparable up to 17 wraps contained within the object (corresponding to a median distance of 12 voxels between wraps) for all data-sets, PRELUDE's speed went down significantly above this number while the time required by ΦUN stayed constant (fig. 7.7). Up to the point where the behavior of ΦUN and PRELUDE started deviating, both methods needed less than 200 ms for each slice at all SNR levels. At higher wrap density PRELUDE did not only show a dependency on the number of wraps but also on SNR. For high SNR images (SNR = 83) unwrapping the images with the maximum number of wraps (31) took PRELUDE about 5 times longer than unwrapping images with up to 17 wraps. In case of the low SNR images (SNR = 5.1), PRELUDE was approximately a factor 20 slower than ΦUN.

[1] PRELUDE was used only in 2D mode for this study, since only two slices were available.

[2] The only exception being the one pixel wide rim unwrapped by ΦUN that was described previously.

7.2 Subject Data

ΦUN's main area of application is SWI. Thus, I chose to evaluate ΦUN's performance on SWI data. SWI data of five healthy subjects were investigated retrospectively.

Data Acquistion and Processing

All data were acquired using the same fully first order flow compensated 3D gradient echo sequence on the same 1.5 T MR scanner (MAGNETOM VISION, SIEMENS, Erlangen, Germany) as for the phantom data. For all measurements the standard CP receive/transmit head coil was used. The parameters of all scans were TR = 64 ms, TE = 40 ms and $\alpha = 25\,^{\circ}$. The slices were parallel to the AC-PC line. The matrix size was $512\times256\times32$ and FoV $25.6\times19.2\times6.4\,\mathrm{cm}^3$. Before reconstruction, each slice was zero filled to a resolution of 512×384 which resulted in an isotropic in-plane voxel size. One data-set was zero-filled to 64 slices, two to 96 slices and the remaining two to 128 slices. Of each data-set the outermost slices were discarded because of fold overs.

Results

The results of ΦUN, in both single region as well as in multi-region mode, and PRELUDE in 3D and in hybrid mode were in very good agreement. The results differed by single voxels within the brain at maximum. In those situations, even a close inspection of all data did not reveal which prediction was the correct one. In most cases, the disagreement was in pixels that either had a very low magnitude or were surrounded by pixels with a very poor SNR. In areas with very low signal, such as regions close to the paranasal sinuses, PRELUDE tended to unwrap further into the signal drop out, but left small holes with wrapped pixels. ΦUN, on the other hand, created smoother boundaries. This resembles the behavior observed in the phantom (fig. 7.8).

The main difference between the methods was the fatty rim around the brain. For studying brain data this region is not of interest, thus the value of the phase therein is not of relevance. Nevertheless, since the algorithms behaved differently, this behavior shall be described. The fatty rim around the brain shows rather inhomogeneous signal intensities (fig. 7.8(a)) and is rather poorly connected. In general ΦUN in single region unwrapped this fatty area only partly (fig. 7.8(c)). The reason for this is that those areas that had higher signal intensities were poorly connected to the rest. In multi-region mode, ΦUN tended to unwrap more of the fatty area, because additional seed points were placed in the rim. But because of the poor quality of the data in these areas, the phase predictions were not reliable always and artifacts could be observed. PRELUDE, on the other hand, unwrapped most of the fatty region (fig. 7.8(d)) and the phase predictions were more reliable than those of ΦUN.

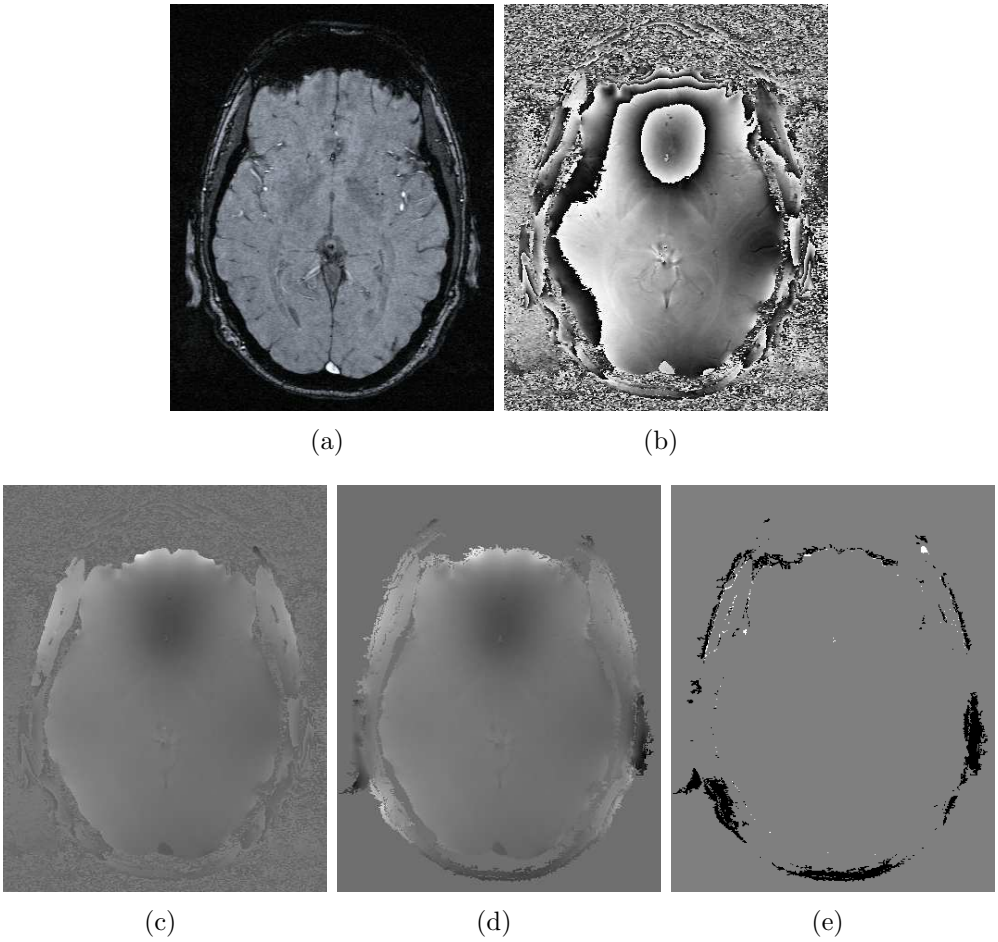
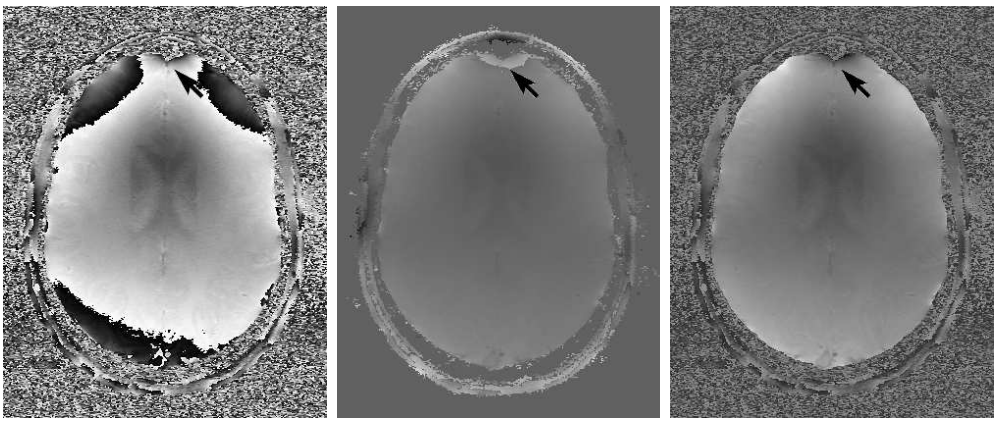


Figure 7.8: A single slice of one of the subjects ((a) magnitude image, (b) phase image) unwrapped by ΦUN (c) and PRELUDE 2D (d). (e) shows the difference between the results obtained with ΦUN and PRELUDE. White pixels indicate pixels for which ΦUN computed an unwrapped phase (that differed from the wrapped phase) but not PRELUDE, black pixels those for which PRELUDE calculated an unwrapped phase but not ΦUN. Except for the fatty area around the brain, of which ΦUN unwrapped only a small fraction, there are small differences in the frontal area because of different termination criteria. Within the brain there are four pixels that were not unwrapped by PRELUDE.



(a) (b) (c)

Figure 7.9: Wrapped phase (a) of a subject where PRELUDE in 3D mode generated phase wraps that were not present in the original phase image (b). For comparison the phase image obtained with ΦUN is shown in (c). The position of the artifact produced by PRELUDE is indicated by arrow heads in all three images.

Although in general, PRELUDE in 3D mode also yielded results that were in good agreement with the other methods, it occassionally produced wraps in areas where no wraps were present in the original phase image (fig. 7.9). An inspection of all images (magnitude, wrapped phase and phase images unwrapped by the other methods) did not give any insight into the causes for these wraps.

A comparison of the times required for unwrapping did not reveal a dependency on the number of slices interpolated from the original data-set by zero-filling. In all cases, ΦUN in single region mode required much less computation time than the other methods. Unwrapping of a single slice with ΦUN took less than 0.4 s. For unwrapping in multi-region mode, ΦUN required between 0.6 s and 0.8 s for each slice. In 2D mode, PRELUDE required between 2.49 s and 7.59 s per slice. The unwrapping times of PRELUDE in the other modes were extremely long. In 3D mode, unwrapping by PRELUDE took between 29.42 s and 397.56 s. In hybrid mode, PRELUDE was even slower and had computation times ranging from 383.16 s to 1509.13 s (fig. 7.10).

7.3 Summary

In case of both the phantom as well as the subject data, the phase predictions by ΦUN and PRELUDE were in very good agreement. ΦUN produced better results on low SNR data-sets. While ΦUN still managed to fully unwrap the images in those situations, PRELUDE left holes with pixels that were not unwrapped. The most likely reason for this behavior is that ΦUN's termination criterion is based on smoothed images, while PRELUDE seemed to consider the magnitude

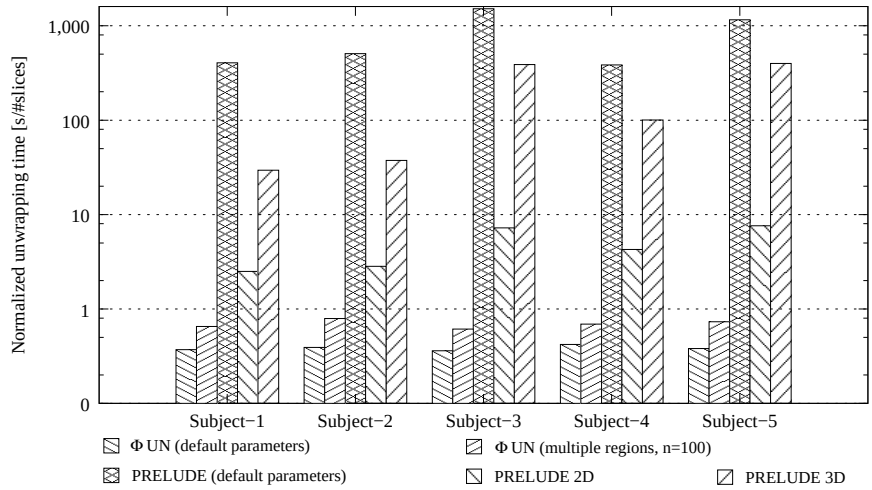


Figure 7.10: Comparison of the times required for unwrapping SWI data-sets of five different subjects. For better comparison the times were normalized to a single slice. While ΦUN achieved unwrapping times below 400 ms in single region mode and less than 800 ms in multi-region mode, PRELUDE was about 10 times slower in 2D mode. Furthermore, PRELUDE used with default parameters and PRELUDE in 3D mode were between 100 and 5000 times slower than ΦUN.

in a single point only. This also explains the differences that were observed in areas close to field inhomogeneities where a similar behavior was observed. ΦUN's tendency to produce smoother boundaries might be advantageous for subsequent post processing steps, since small holes correspond to high spatial frequencies. Thus those areas would have to be excluded if a filter is to be applied to the phase image.

In case of subject data, ΦUN and PRELUDE behaved differently in the fatty area around the brain. PRELUDE tended to unwrap more of this region than ΦUN. In multi-region mode ΦUN manages to unwrap the fatty rim, but does not always provide reliable phase predictions using the default parameters. Nevertheless, this discrepancy is not of importance, because the fatty rim is not of interest and usually is removed in post-processing steps.

The most important difference between the methods concerned the time required for unwrapping and its dependency on SNR. While ΦUN's unwrapping time were independent of SNR, PRELUDE exhibited a strong dependency on SNR. This dependency was larger for the 3D and the hybrid mode than for the 2D mode. The 3D and hybrid mode were unacceptable slow on low SNR phantom data as well as on subject data. In general the hybrid mode required significantly longer than the 3D mode, which is a result of the larger number of initial regions caused by the initial partitioning which is done slice by slice in hybrid mode.

List of References

[1] Jenkinson, M. *Fast, automated, N-dimensional phase-unwrapping algorithm.* Magnetic Resonance in Medicine, **49**(1), 193–197 (2003).

[2] Witoszynskyj, S., Rauscher, A., Reichenbach, J. R., and Barth, M. *ΦUN (Φase UNwrapping) – validation of a 2D Region-Growing Phase Unwrapping Program.* In *Joint Annual Meeting ISMRM-ESMRMB*, 3436 (2007).

[3] Witoszynskyj, S., Rauscher, A., Reichenbach, J. R., and Barth, M. *Phase unwrapping of MR images using ΦUN – a fast and robust region growing algorithm.* Medical Image Analysis, *submitted* (2007).

[4] Reichenbach, J. R. and Haacke, E. M. *High-resolution BOLD venographic imaging: a window into brain function.* NMR in Biomedicine, **14**(7-8), 453–467 (2001).

8 Discussion

8.1 Algorithm

It is intuitively obvious that any region growing approach strongly depends on the criteria used to guide the growth of regions. In my opinion, both, the underlying physics of the phase image as well as the properties of the phase extrapolation technique have to be reflected in the choice of the criteria guiding the growth of regions. The use of the magnitude information is motivated by the physics of MRI: magnitude and the phase SNR are related to each other [1]. Although the magnitude could, in principle, be utilized through binary masks that constrain unwrapping as well, there is a fundamental difference between these approaches. Using the magnitude as quality criterion can rather be seen as using a dynamic binary mask that is adjusted according to the requirements of unwrapping. For each iteration of the loop that relaxes the reliability constraint on the extrapolation, the quality criterion based on the magnitude information is relaxed. This ensures that for each constraint on the extrapolation, pixels with the highest phase SNR are added to the region first.

Using only magnitude information for seed finding is not appropriate, because it does not pose any constraint on the flatness of the phase topography. This flatness is of special importance in the beginning for two reasons: if the number of pixels in a region is small, which is the case in the beginning, predictions tend to be flat, because prediction lines will often contain a single pixel only. Secondly, since the multiples of 2π are integer, the prediction of the unwrapped phase involves rounding to the nearest integer. Because of noise the result of the rounding step is more reliable if the number that is to be rounded is close to an integer. This always is the case if the phase topography is flat. Thus for

seed-finding the local phase coherence is used, since it is highest in areas of flat phase topography.

Just as any post-processing scheme, ΦUN has certain requirements and poses constraints on the data it is applied to. The most obvious constraint is that the phase topography is free of singularities in areas with sufficient signal intensity. However, this condition is not fulfilled always. The two most important causes for singularities are incorrectly combined phase images acquired with phased array coils and filters that were applied to the image before unwrapping. Combining images of phased array coils is challenging by itself although the formula for a correct combination has been known for years [2]. The difficulty arises from the generally unknown coil sensitivities required for such a combination. Phase unwrapping of images obtained with phased array coils thus remains a challenge.

Another precondition on the data is that the phase topography has to be smooth. There are situations where this is not the case. It was mentioned in chapter 3 that the different resonance frequencies of Hydrogen nuclei bound to water and Hydrogen nuclei bound to lipids cause a modulation of the signal in dependence on the echo time. By carefully choosing the echo time the phase of a voxel containing fat can differ from a voxel containing water by exactly π [3, 4]. It is obvious that in such a situation, the requirement on the smoothness of the phase image is not fulfilled. As long as no partial volume effects occur this problem should have no influence on the algorithms performance, since regions containing fat and regions containing water would be unwrapped separately. Nevertheless the consequences of partial volume effects would have to be studied carefully. Such an investigation was not done for this thesis, because the main area of application of ΦUN are brain images. Since brain tissue barely contains fat, this problem does not occur.

8.2 2D vs. 3D

MRI data, especially SWI data, often are three dimensional. Since, in principle, extending the region growing approach to three dimensions should be possible, modifying the algorithm for three dimensional phase unwrapping was considered. Nevertheless there are several reasons why the two dimensional approach was preferred.

The main advantage of a three dimensional algorithm compared to two dimensional unwrapping is that the unwrapped phase is consistent across slices. On the other hand, in many applications, especially in SWI, the application ΦUN was mainly intended for, such a consistency is not of importance. In those applications the focus is on local phase changes and the main goal of unwrapping is to enable removing phase changes of low spatial frequency. Furthermore as long as each slice is unwrapped reliably, at least in principle, a consistent phase across slices can be achieved relatively easy by adding multiples of 2π. Thus

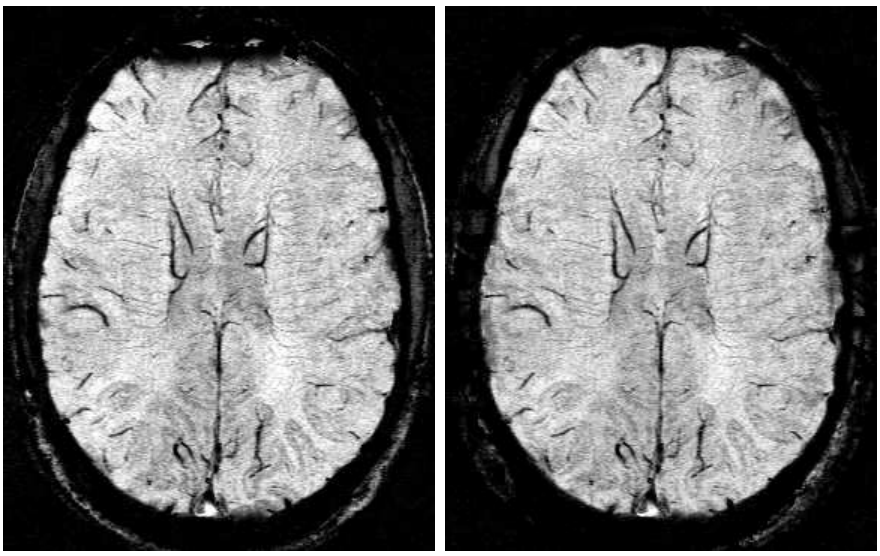
the main question when considering a three dimensional unwrapping algorithm is whether unwrapping in three dimensions can improve phase predictions or if this adds additional complexity to the problem.

For a discussion of possible advantages and disadvantages of a three dimensional unwrapping algorithm, the different types of data have to be considered first. As described in chapter 3, MRI data can be acquired either truly three dimensional, or slice by slice. In the latter case, there often is a gap between slices. In this situation two dimensional unwrapping is more appropriate, since little is known about how the phase evolves between slices. For the other cases, data obtained with three dimensional sequences and data acquired with two dimensional sequences with no gap between slices, three dimensional phase unwrapping might have advantages.

Voxel dimensions often are anisotropic – the in plane dimensions generally are much smaller than the slice thickness. The influence of the voxel anisotropy on the phase predictions would have to be carefully investigated before an attempt to extend the region growing approach to three dimensions can be made. It is quite likely that the anisotropy has to be taken into account somehow. This would add an additional complexity and further parameters to the problem.

Furthermore, in three dimensions, each voxel has 26 neighbors instead of eight. This does not only increase the number of predictions that have to be made, but also the chance that a prediction fails to meet one of the criteria and thus more attempts to unwrap a voxel might be necessary. Both, the increased number of predictions and the higher likelihood of repeated attempts to unwrap a voxel, would increase the computation time significantly, which is contrary to one of the main goals of the implementation presented in this thesis. Furthermore the results of PRELUDE in 3D mode show, that three dimensional unwrapping does not necessarily lead to improved phase images but can be an additional source of artifacts.

A compromise between two dimensional and three dimensional unwrapping could be a hybrid approach. This could be achieved by using three dimensional quality maps for two dimensional unwrapping. This could be either done by using three dimensional maps for seed finding, or unwrapping, or even both. In every case an important question is how to take voxel anisotropy into account. While this is probably not of great importance for seed finding, voxel anisotropy would have to be considered for unwrapping especially since it might cause a premature termination of the algorithm. The main advantage of a three dimensional map for seed finding would be that seeds would be placed such that the unwrapped phase of the first regions to be created in each slice would be consistent across slices. This phase consistency would apply to the first seeds only, it is thus not guaranteed that the final phase images would be consistent across slices. For this reason and because consistency across slices is not a concern for the application ΦUN was mainly intended for, this approach was not tried.



(a) (b)

Figure 8.1: Venogram computed with homodyne filtered phase images (a) and with unwrapped phase images (b). The venogram computed with homodyne filtered phase images is distorted in the frontal areas because the homodyne filter creates artifacts in areas of steep phase topography.

8.3 Applications

Susceptibility Weighted Imaging

ΦUN was developed with having Susceptibility Weighted Imaging (SWI) in mind as its main application. SWI is an MRI technique that utilizes long echo times and high resolution data acquisition [5]. Because of the long echo times SWI is sensitive to extremely small field inhomogeneities, which are caused by small differences of the susceptibilities of different types of tissue. At the same time the high resolution limits signal loss due to intra-voxel dephasing. By using this technique small venous vessels can be uncovered, since the susceptibility of deoxygenated blood differs from the surrounding tissue. Usually, phase information is incorporated into the magnitude image by weighting the magnitude image with a phase mask. This mask should be sensitive to local phase changes only (e.g. as caused by venous vessels). The conventional approach is to apply a homodyne filter [6]. This filter removes phase variations of low spatial frequencies as well as phase wraps. Nevertheless, a trade off between preserving structures with high spatial frequencies and the (desired) insensitivity towards steep phase topographies has to be made. This can lead to artifacts in areas with steep phase topographies. Figure 8.1(a) shows a venogram that was obtained using this technique. The artifacts caused by steep phase topography are clearly visible in the frontal area.

Artifacts in areas with steep phase topography can be avoided if the phase images are unwrapped. However, before the phase mask can be computed the background field inhomogeneities have to be suppressed. This can be achieved by high-pass filtering [7, 8]. Figure 8.1(b) displays a venogram computed from unwrapped phase images of the same data set as before. Since SWI data-sets have a high-resolution and a comparatively low SNR, it is necessary that a phase unwrapping algorithm can handle such images efficiently. That ΦUN fulfills those requirements was shown in this thesis.

Magnetic Field Mapping

An important application for phase images is the computation of maps of the static magnetic field. These maps can be used to correct for distortions in EPI [9]. The standard approach for calculating field maps is to acquire two images at two different echo times and to compute the complex ratio of these images [10, 11]

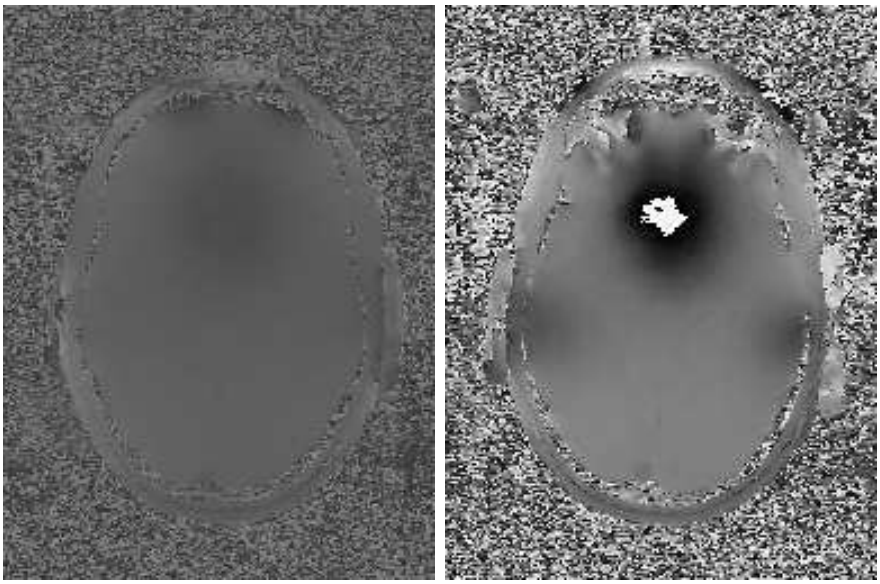
$$\frac{a_2 + ib_2}{a_1 + ib_1} = \frac{A_2\,\mathrm{e}^{i\phi_2}}{A_1\,\mathrm{e}^{i\phi_1}} = \frac{A_2}{A_1}\,\mathrm{e}^{i(\phi_2-\phi_1)}\,, \tag{8.1}$$

where $A_1 = |a_1 + ib_1|$ and $A_2 = |a_2 + ib_2|$ are the magnitude images and ϕ_1 and ϕ_2 the respective phase images. The phase difference $\Delta\phi = \phi_2 - \phi_1$ is proportional to the static magnetic field. Just as a phase image, the map of $\Delta\phi$ may be affected by phase wraps, which have to be resolved. In principle, there are two approaches to obtain field maps that are not affected by phase wraps. The first is to unwrap each image first and then subtract the unwrapped phase images from each other. The second is to apply phase unwrapping after the complex ratio of the two images has been taken. Although this second approach should work in principle, the first method might be more stable in case one image contains areas of signal drop-outs. So far all field maps were calculated using the subtraction of two unwrapped phase images.

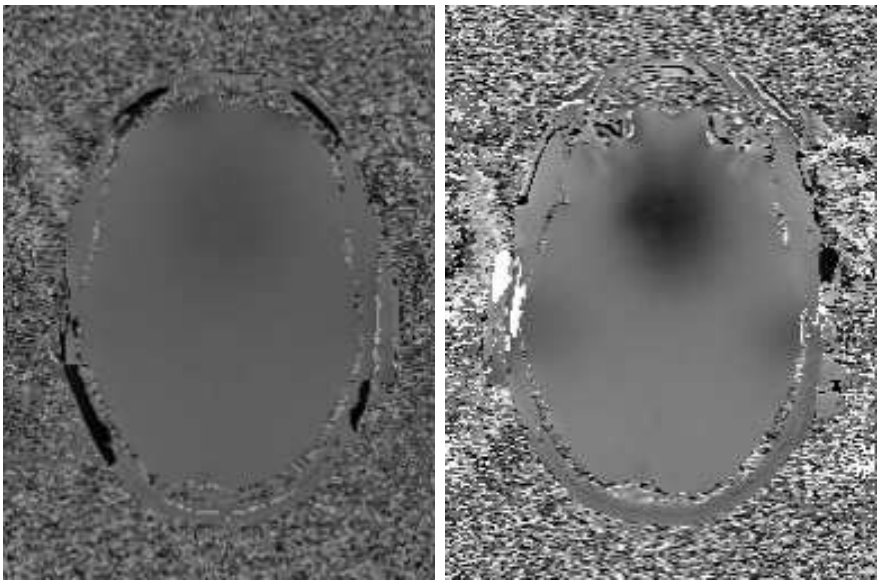
Figure 8.2 shows an example of two field maps computed by taking the complex ratio (top row) and of the same field maps computed by subtracting unwrapped phase images (bottom row). In case of strong field inhomogeneities the field map computed by the complex ratio might be affected by phase wraps (fig. 8.2(b)). Using unwrapped phase images instead avoids this problem (fig. 8.2(d)).

Other Applications

Although ΦUN was developed with the application to brain data in mind, there are other organs and applications for which ΦUN could be of interest. One of these organs is the female breast. Phase unwrapping could facilitate the segmentation of tissues by exploiting the opposed phase condition [3, 4]. Although the opposed phase condition violates one of the preconditions (i.e. the smoothness

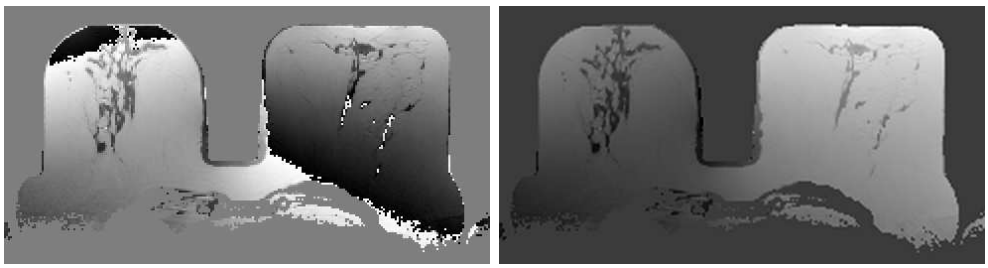


(a) (b)



(c) (d)

Figure 8.2: Field maps computed by taking the complex ratio of two complex images (top row) and by subtracting the unwrapped phase images (bottom row). Two slices of a data-set are shown in the left and the right column, respectively. If the field inhomogeneity is too large for a given echo time difference, the field map obtained by the complex ratio can be affected by phase wraps (b). This is not the case if the same map is computed from unwrapped phase images (d).

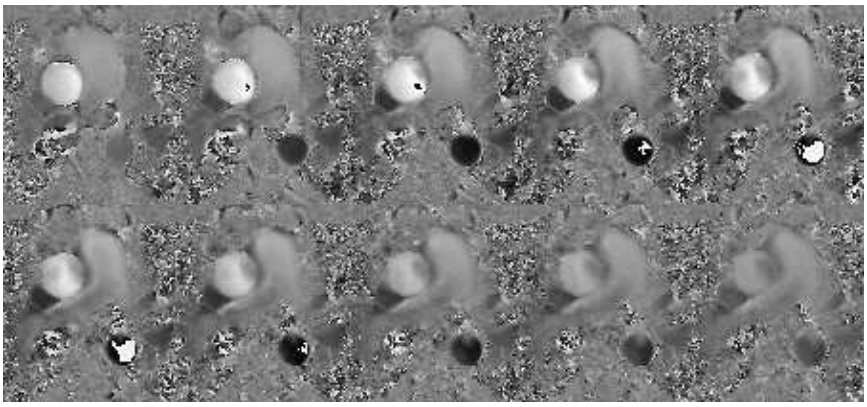


(a) (b)

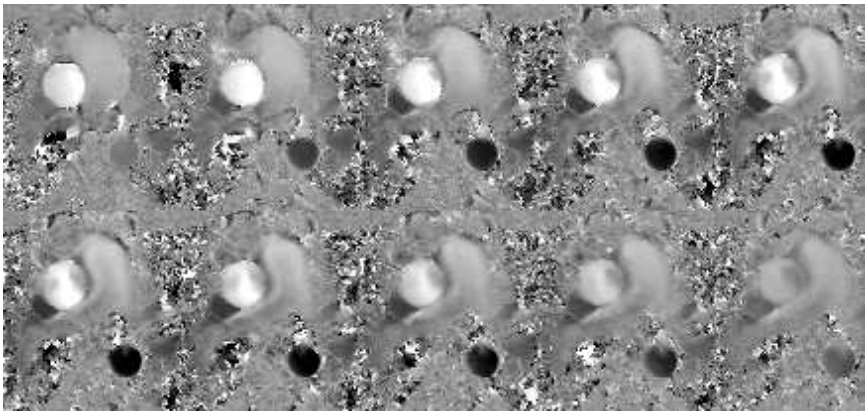
Figure 8.3: Wrapped (a) and unwrapped (b) opposed phase image of a mamma.

of the phase) for applying ΦUN, a first attempt to apply ΦUN to images of the breast was successful (fig. 8.3). Also SWI of the breast might be of interest to distinguish between calcification and bleeding [12]. Nevertheless, the applicability of ΦUN to breast data sets would have to be investigated on a larger number of data-sets.

Another application where phase unwrapping can be of importance is phase contrast angiography. In this application additional gradients are used to encode flow [13]. Just as in DOPPLER sonography, aliasing occurs if the flow exceeds the maximum encoding velocity. In the phase images this effect appears as phase wraps. Phase unwrapping can thus be used to extract the correct flow velocity. It was shown exemplarily that ΦUN can, at least in principle, be applied to phase contrast data (fig. 8.4). However, also in this case a study involving more data-set would be necessary to conclude on how applicable ΦUN is for this type of data.



(a)



(b)

Figure 8.4: A time series of flow encoded images of the aorta. In both the ascending and the descending branch of the aorta the phase is wrapped in the original data (a). Phase unwrapping with two or more regions produces correct velocity profiles for both branches.

List of References

[1] Conturo, T. E. and Smith, G. D. *Signal-to-noise in phase angle reconstruction: dynamic range extension using phase reference offsets.* Magnetic Resonance in Medicine, **15**(3), 420–437 (1990).

[2] Roemer, P. B., *et al. The NMR phased array.* Magnetic Resonance in Medicine, **16**(2), 192–225 (1990).

[3] Dixon, W. T. *Simple proton spectroscopic imaging.* Radiology, **153**(1), 189–194 (1984).

[4] Reichenbach, J. R., *et al. Subtraction of in-phase and opposed-phase images in dynamic MR mammography.* Journal of Magnetic Resonance Imaging, **21**(5), 565–575 (2005).

[5] Reichenbach, J. R., Venkatesan, R., Schillinger, D., and Haacke, E. M. *Small vessels in the human brain: MR-venography with deoxyhemoglobin as an intrinsic contrast agent.* Radiology, **204**, 272–277 (1997).

[6] Noll, D. C., Nishimura, D. G., and Makovski, A. *Homodyne detection in magnetic resonance imaging.* IEEE Transactions on Medical Imaging, **10**, 154–163 (1991).

[7] Rauscher, A., *et al. Magnetic susceptibility-weighted MR phase imaging of the human brain.* American Journal of Neuroradiology, **26**(4), 736–742 (2005).

[8] Rauscher, A., *et al. Automated unwrapping of MR phase images applied to BOLD MR-venography at 3 Tesla.* Journal of Magnetic Resonance Imaging, **18**(2), 175–180 (2003).

[9] Jezzard, P. and Balaban, R. S. *Correction for geometric distortion in echo planar images from B_0 field variations.* Magnetic Resonance in Medicine, **34**, 65–73 (1995).

[10] Cusack, R. and Papadakis, N. *New robust 3–D phase unwrapping algorithms: Application to magnetic field mapping and undistorting echoplanar images.* NeuroImage, **16**, 754–764 (2002).

[11] Hutton, C., *et al. Image distortion correction in fMRI: A quantitative evaluation.* NeuroImage, **16**(1), 217–40 (2002).

[12] Deistung, A., *et al. Demonstration of paramagnetic and diamagnetic cerebral lesions by using susceptibility weighted phase imaging (SWI).* Zeitschrift für medizinische Physik, **16**(4), 261–267 (2006).

[13] O'Donnell, M. *NMR blood flow imaging using multiecho, phase contrast sequences.* Medical Physics, **12**(1), 59–64 (1985).

9 Conclusions

For this thesis a 2D region-growing phase unwrapping algorithm optimized for MRI data was developed, implemented and evaluated. This was achieved by using magnitude information for guiding the unwrapping. The use of magnitude information can be motivated by the physics of MRI. The algorithm's performance was tested extensively on both phantom and in vivo data and compared to an established method (PRELUDE). Although both algorithms performed reliable in high SNR areas, ΦUN provided significant improvements in terms of speed and robustness in low SNR areas. This makes ΦUN a suitable application for SWI and EPI data which often are hampered by low SNR areas. Applying ΦUN to other organs and applications, such as data of the female breast and phase contrast angiography, led to promising results. Nevertheless, testing ΦUN on these data was out of the scope of this thesis and was only done as a prove of principle.

List of Figures

List of Tables

Bibliography

[1] *Analyze 7.5 file format*. http://www.mayo.edu/bir/PDF/ANALYZE75.pdf.

[2] *Cygwin*. http://www.cygwin.com/.

[3] *MinGW - Minimalist GNU for Windows*. http://www.mingw.org/.

[4] An, L., Xiang, Q.-S., and Chavez, S. *A fast implementation of the minimum spanning tree method for phase unwrapping*. IEEE Transactions on Medical Imaging, **19**(8), 805–808 (2000).

[5] Badurek, G. *Biological and Medical Applications of Nuclear Physics 2*. Lecture Notes.

[6] Baldi, A. *Phase Unwrapping by Region Growing*. Applied Optics, **42**(14), 2498–2505 (2003).

[7] Beiser, A. *Concepts of Modern Physics*. McGraw-Hill, 6 edition (2003).

[8] Bernstein, M. A., King, K. F., and Zhou, X. J. *Handbook of MRI Pulse Sequences*. Elsevier (2004).

[9] Bone, D. J. *Fourier fringe analysis: the two-dimensional phase unwrapping problem*. Applied Optics, **30**(25), 3627–3632 (1991).

[10] Bourgeat, P., *et al. MR image segmentation of the knee bone using phase information*. Medical Image Analysis, **11**(4), 325–335 (2007).

[11] Bracewell, R. N. *The Fourier Transfrom and Its Applications*. McGraw-Hill, 3 edition (1999).

[12] Bryant, D. J., Payne, J. A., Firmin, D. N., and Longmore, D. B. *Measurement of flow with NMR imaging using a gradient pulse and phase difference technique*. Journal of Computer Assisted Tomography, **8**(4), 588–593 (1984).

[13] Chavez, S., Xiang, Q.-S., and An, L. *Understanding phase maps in MRI: a new cutline phase unwrapping method*. IEEE Transactions on Medical Imaging, **21**(8), 966–977 (2002).

[14] Chen, C. and Zebker, H. *Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization.* Journal of the Optical Society of America. A, Optics, image science, and vision, **18**(2), 338–51 (2001).

[15] Ching, N. H., Rosenfeld, D., and Braun, M. *Two-Dimensional Phase Unwrapping Using a Minimum Spanning Tree Algorithm.* IEEE Transactions on Image Processing, **1**(3), 355–365 (1992).

[16] Committee on Data for Science and Technology. *2006 CODATA recommended values.* http://physics.nist.gov/cuu/Constants/.

[17] Conturo, T. E. and Smith, G. D. *Signal-to-noise in phase angle reconstruction: dynamic range extension using phase reference offsets.* Magnetic Resonance in Medicine, **15**(3), 420–437 (1990).

[18] Cusack, R. and Papadakis, N. *New robust 3–D phase unwrapping algorithms: Application to magnetic field mapping and undistorting echoplanar images.* NeuroImage, **16**, 754–764 (2002).

[19] Deistung, A., *et al.* *Demonstration of paramagnetic and diamagnetic cerebral lesions by using susceptibility weighted phase imaging (SWI).* Zeitschrift für medizinische Physik, **16**(4), 261–267 (2006).

[20] Demtröder, W. *Experimentalphysik 4.* Springer, 2 edition (2005).

[21] Demtröder, W. *Experimentalphysik 1.* Springer, 4 edition (2006).

[22] Demtröder, W. *Experimentalphysik 2.* Springer, 4 edition (2006).

[23] Dixon, W. T. *Simple proton spectroscopic imaging.* Radiology, **153**(1), 189–194 (1984).

[24] Duyn, J. H., *et al.* *High-field MRI of brain cortical substructure based on signal phase.* Proceedings of the National Academy of Sciences, **104**(28), 11796–11801 (2007).

[25] Edelstein, W. A., Glover, G. H., Hardy, C. J., and Redington, R. W. *The intrinsic signal-to-noise ratio in NMR imaging.* Magnetic Resonance in Medicine, **3**(4), 604–618 (1986).

[26] Frey, B. J., Koetter, R., Moran, J., and Jr., D. J. M. *Variations on phase unwrapping.* In *International Symposium on Information Theory and Its Applications.* International Symposium on Information Theory and Its Applications, Honolulu, USA (2000).

[27] Frey, B. J., Koetter, R., and Petrovic, N. *Very loopy belief propagation for unwrapping phase images.* In *Advances in Neural Information Processing Systems 14.* MIT Press (2001).

[28] Ghiglia, D. C. and Pritt, M. D. *Two-Dimensional Phase Unwrapping*. John Wiley & Sons, Inc. (1998).

[29] Griffiths, D. J. *Introduction to Quantum Mechanics*. Prentice Hall (1994).

[30] Griswold, M. A., *et al. Generalized autocalibrating partially parallel acquisitions (GRAPPA)*. Magnetic Resonance in Medicine, **47**(6), 1202–1210 (2002).

[31] Gudbjartsson, H. and Patz, S. *The rician distribution of noisy MRI data*. Magnetic Resonance in Medicine, **34**(6), 910–914 (1995).

[32] Haacke, E. M., Brown, R. W., Thompson, M. R., and Venkatesan, R. *Magnetic Resonance Imaging – Physical Principles and Sequence Design*. Wiley-Liss (1999).

[33] Hedley, M. and Rosenfeld, D. *A new two-dimensional phase unwrapping algorithm for MRI images*. Magnetic Resonance in Medicine, **24**(1), 177–181 (1992).

[34] Hutton, C., *et al. Image distortion correction in fMRI: A quantitative evaluation*. NeuroImage, **16**(1), 217–40 (2002).

[35] Jenkinson, M. *Fast, automated, N-dimensional phase-unwrapping algorithm*. Magnetic Resonance in Medicine, **49**(1), 193–197 (2003).

[36] Jezzard, P. and Balaban, R. S. *Correction for geometric distortion in echo planar images from B_0 field variations*. Magnetic Resonance in Medicine, **34**, 65–73 (1995).

[37] Johnson, G. and Hutchison, J. M. S. *The limitations of NMR recalled-echo imaging techniques*. Journal of Magnetic Resonance, **63**(1), 14–30 (1985).

[38] Koetter, R., Frey, B. J., Petrovic, N., and Jr., D. C. M. *Unwrapping phase images by propagating probabilities across graphs*. In *International Conference on Acoustics, Speech, and Signal Processing*. ICASSP (2001).

[39] Kruse, S. A., *et al. Tissue characterization using magnetic resonance elastography: preliminary results*. Physics in Medicine and Biology, **45**(6), 1579–1590 (2000).

[40] Lauterbur, P. *Image formation by induced local interaction: Examples employing nuclear magnetic resonance*. Nature, **242**, 190–191 (1973).

[41] Liang, Z. P. and Lauterbur, P. *Principles of Magnetic Resonance Imaging*. IEEE Press (2004).

[42] Noll, D. C., Nishimura, D. G., and Makovski, A. *Homodyne detection in magnetic resonance imaging.* IEEE Transactions on Medical Imaging, **10**, 154–163 (1991).

[43] O'Donnell, M. *NMR blood flow imaging using multiecho, phase contrast sequences.* Medical Physics, **12**(1), 59–64 (1985).

[44] Pruessmann, K. P. *Encoding and reconstruction in parallel MRI.* NMR in Biomedicine, **19**(3), 288–299 (2006).

[45] Rauscher, A. *Phase information in magnetic resonance imaging.* Ph.D. thesis, Technische Universität Wien (2005).

[46] Rauscher, A., *et al. Automated unwrapping of MR phase images applied to BOLD MR-venography at 3 Tesla.* Journal of Magnetic Resonance Imaging, **18**(2), 175–180 (2003).

[47] Rauscher, A., *et al. Magnetic susceptibility-weighted MR phase imaging of the human brain.* American Journal of Neuroradiology, **26**(4), 736–742 (2005).

[48] Reichenbach, J. R. and Haacke, E. M. *High-resolution BOLD venographic imaging: a window into brain function.* NMR in Biomedicine, **14**(7-8), 453–467 (2001).

[49] Reichenbach, J. R., Venkatesan, R., Schillinger, D., and Haacke, E. M. *Small vessels in the human brain: MR-venography with deoxyhemoglobin as an intrinsic contrast agent.* Radiology, **204**, 272–277 (1997).

[50] Reichenbach, J. R., *et al. Subtraction of in-phase and opposed-phase images in dynamic MR mammography.* Journal of Magnetic Resonance Imaging, **21**(5), 565–575 (2005).

[51] Roemer, P. B., *et al. The NMR phased array.* Magnetic Resonance in Medicine, **16**(2), 192–225 (1990).

[52] RSI. *IDL External Development Guide*, IDL Version 6.2 edition (2005).

[53] Schwartzkopf, W., *et al. Two-dimensional phase unwrapping using neural networks.* In *Proc. IEEE Southwest Symposium on Image Analysis, April 2-4, 2000*, 274–277 (2000).

[54] Sodickson, D. K. and Manning, W. J. *Simultaneous Acquisition of Spatial Harmonics (SMASH): fast imaging with radiofrequency coil arrays.* Magnetic Resonance in Medicine, **38**(4), 591–603 (1997).

[55] Stramaglia, S., Nico, G., Pasquariello, G., and Guerriero, L. *Phase unwrapping method based on stochastic relaxation.* In *EUROPTO - Image Processing, Signal Processing, and Synthetic Aperture Radar for Remote Sensing proc. SPIE*, volume 3217, 4–12 (1997).

[56] Twieg, D. B. *The k-trajectory formulation of the NMR imaging process with applications in analysis and synthesis of imaging methods.* Medical Physics, **10**(5), 610–621 (1983).

[57] Volkov, V. V. and Zhu, Y. *Deterministic phase unwrapping in the presence of noise.* Optics Letters, **28**(22), 2156–2158 (2003).

[58] Witoszynskyj, S., Rauscher, A., Reichenbach, J. R., and Barth, M. *Automated phase unwrapping of MR images at different field strengths using multiple quality maps.* In *13th Annual Meeting of the ISMRM*, 2249 (2005).

[59] Witoszynskyj, S., Rauscher, A., Reichenbach, J. R., and Barth, M. *Quality maps for automated phase unwrapping of MR images.* In *European Congress of Radiology*, B–102 (2005).

[60] Witoszynskyj, S., Rauscher, A., Reichenbach, J. R., and Barth, M. *Phase unwrapping of MR images using ΦUN – a fast and robust region growing algorithm.* Medical Image Analysis, *submitted* (2007).

[61] Witoszynskyj, S., Rauscher, A., Reichenbach, J. R., and Barth, M. *ΦUN (Φase UNwrapping) – validation of a 2D Region-Growing Phase Unwrapping Program.* In *Joint Annual Meeting ISMRM-ESMRMB*, 3436 (2007).

[62] Xu, W. and Cumming, I. *A region-growing algorithm for InSAR phase unwrapping.* IEEE Transactions on Geoscience and Remote Sensing, **37**(1), 124–134 (1999).

[63] Yao, W. M., *et al. Review of Particle Physics.* Journal of Physics G, **33** (2006).