MAGNETIC RESONANCE IMAGE RECONSTRUCTION
FROM HIGHLY UNDERSAMPLED K-SPACE DATA
USING DICTIONARY LEARNING

BY
SAIPRASAD RAVISHANKAR

THESIS
Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Adviser:
Professor Yoram Bresler
Compressed sensing (CS) utilizes the sparsity of MR images to enable accurate reconstruction from undersampled k-space data. Recent CS methods have employed analytical sparsifying transforms such as wavelets, curvelets, and finite differences. In this thesis, we propose a novel framework for adaptively learning the sparsifying transform (dictionary), and reconstructing the image simultaneously from highly undersampled k-space data. The sparsity in this framework is enforced on overlapping image patches emphasizing local structure. Moreover, the dictionary is adapted to the particular image instance, thereby favoring better sparsities and consequently much higher undersampling rates. The proposed alternating reconstruction algorithm learns the sparsifying dictionary, and uses it to remove aliasing and noise in one step, and subsequently restores and fills in the k-space data in the other step. Numerical experiments are conducted on MR images and on real MR data of several anatomies with a variety of sampling schemes. The results demonstrate dramatic improvements on the order of 4-18 dB in reconstruction error and doubling of the acceptable undersampling factor using the proposed adaptive dictionary as compared to previous CS methods. These improvements persist over a wide range of practical data SNRs, without any parameter tuning.

As a further enhancement to the proposed dictionary learning scheme for MRI reconstruction, we explore the use of an additive multiscale dictionary formulation. This formulation enforces sparsity of the reconstructed image simultaneously at multiple scales (patch sizes) and combines the results at those scales to obtain superior reconstructions. The multiscale dictionary in the proposed formulation is a collection of several single scale dictionaries that operate separately. The alternating reconstruction algorithm learns the various single scale sparsifying dictionaries and uses them to remove image artifacts in one step, and then restores and fills in k-space in the other step.
Experiments conducted on several MR images using simulated k-space undersampling with a variety of sampling schemes show promising improvements of up to 1.4 dB in reconstruction error with the proposed multiscale dictionary as compared to a dictionary learned at only one scale. This improvement is also achieved at a substantially lower computational complexity for the multiscale formulation, thereby demonstrating that (additive) multiscale sparse representations are both better and faster. The final improvement explored in this thesis is a sequential multiscale reconstruction algorithm that starts with the lowest scale and adds in the higher scales sequentially over iterations. This approach is shown to be faster than the one where all scales are used for all the iterations, while achieving the same PSNR in the reconstructed image.
To my parents and brother, for their support
ACKNOWLEDGMENTS

Thanks to my wonderful adviser, Prof. Yoram Bresler, for his highly insightful guidance and many helpful discussions, comments, and suggestions throughout the work; to my colleague Kiryung Lee for several useful discussions and for being very helpful and kind throughout; and to secretaries Barbara Horner and Terri Hovde for being of great, immediate assistance on several occasions.
# TABLE OF CONTENTS

## CHAPTER 1 INTRODUCTION
1.1 Figures ........................................... 5

## CHAPTER 2 BACKGROUND AND RELATED WORK
2.1 CSMRI ............................................ 6
2.2 Dictionary Learning ................................. 8
2.3 Figure ........................................... 11

## CHAPTER 3 PROBLEM FORMULATION AND ALGORITHM
3.1 Problem Formulation ................................. 12
3.2 Algorithm .......................................... 14
3.3 Convergence ...................................... 17
3.4 Parameters ...................................... 17
3.5 Algorithm Complexity ............................... 18
3.6 Figure ........................................... 19

## CHAPTER 4 EXPERIMENTS
4.1 Framework ....................................... 20
4.2 Noiseless Case ................................... 22
4.3 Performance with Noise ............................ 24
4.4 Evaluation of Undersampling Limit ............... 25
4.5 Parameter Evaluation .............................. 26
4.6 Figures ........................................... 28

## CHAPTER 5 MULTISCALE DICTIONARIES FOR MRI
5.1 Introduction ..................................... 44
5.2 Problem Formulation ............................... 47
5.3 Algorithm ........................................ 49
5.4 Algorithm Properties .............................. 51
5.5 Multiscale Experiments ............................ 53
5.6 A Sequential Multiscale Algorithm ............... 57
5.7 Figures ........................................... 59

## CHAPTER 6 CONCLUSIONS ......................... 72

## REFERENCES ..................................... 74
CHAPTER 1
INTRODUCTION

Magnetic resonance imaging (MRI) is a non-invasive and non-ionizing imaging technique. Offering a variety of contrast mechanisms, it enables excellent visualization of both anatomical structure and physiological function. Owing to these advantages, MRI is one of the major diagnostic imaging modalities. However, the main limitation of MRI, affecting both clinical throughput and image quality, especially in dynamic imaging applications, is that it is a relatively slow imaging modality. This is because the data in MRI, samples in k-space of the spatial Fourier transform of the object, are acquired sequentially in time. In spite of advances in scanner hardware and pulse sequences, the rate at which MR data are acquired is limited by MR physics and physiological constraints on RF energy deposition in the body. A variety of MR techniques therefore aim to reduce the number of data required for accurate reconstruction.

Hardware-based, parallel data acquisition (P-MRI) methods (cf. [1]) reduce the number of k-space samples acquired, and use the diversity provided by multiple RF receiver coils to eliminate the resulting aliasing. P-MRI is widely available in commercial systems and plays an important role in clinical practice. However, although tens of receiver coils may be available, P-MRI is nonetheless limited by increased noise and imperfect alias correction to accelerations typically smaller than 3 or 4 fold. Providing a complement to hardware-based acceleration are algorithmic reduced acquisition MRI methods. These rely on implicit or explicit modeling or constraints on the underlying image or object [2], with some methods even adapting the acquisition to the imaged object [3, 4]. Compressed sensing is a recent addition to this arsenal, and is the subject of this work.

The recent theory of compressed sensing (CS) [5, 6, 7] (see also [8, 9, 10, 11, 12, 13, 14, 15] for the earliest versions of CS for Fourier-sparse signals and for Fourier imaging) enables accurate recovery of signals/images
using significantly fewer measurements than the number of unknowns, or
than mandated by traditional Nyquist sampling. This is possible provided
the underlying signal or image is sparse in some transform domain, and the
acquisition is incoherent, in an appropriate sense, with the transform. The
cost of this improvement is that the reconstruction procedure is non-linear.
More recently, CS theory has been applied to imaging modalities such as
MRI [16, 17, 18, 19, 20, 21], and CT [22], demonstrating high quality recons-
tructions from a reduced set of measurements.

The sparsity of MR images in some transform domain (wavelets, finite
differences, contourlets etc.), or equivalently, MR images admitting a sparse
representation in some set of signals known as dictionary, are key to accu-
rate CS reconstruction. However, compressed sensing MRI (CSMRI) with
non-adaptive, global sparsifying transforms, is usually limited in typical MR
images to 2.5-3 fold undersampling (CS at higher undersampling rates usually
leads to large aliasing errors [23]). The limitations of CSMRI are illustrated
in Figure 1.1 on a reference fully sampled MR image of the brain (courtesy
of [24]). The CS sampling scheme shown employs variable density random
undersampling (since most of the energy of MR images is concentrated close
to the k-space origin, variable density random undersampling is more appro-
priate for MRI [16]) in k-space by a factor of 20.\footnote{This large undersampling factor was chosen for easy visualization of the reconstruction
error. Significant errors in this CS reconstruction are observed at undersampling factors
as low as 3.}

Adaptive transforms (dictionaries) can sparsify images better since they
are learned for the particular image instance or class of images. Recent
studies on adaptive dictionaries [25, 26] have shown the promise of patch-
based sparsifying dictionaries in a variety of applications such as image/video
denoising, image/video inpainting, deblurring, and demosaicing [27, 28, 29,
30, 31, 32]. The shift from global image sparsity to patch-based sparsity
is appealing since patch based dictionaries can capture local image features
effectively, and can potentially remove noise and aliasing artifacts in CSMRI without sacrificing resolution. Patch based schemes have become popular especially in denoising [33, 27] where using overlapping patches can create an additional averaging effect that removes noise. Furthermore, a single image can be decomposed into sufficiently many overlapping patches to train a sparsifying dictionary.

In this work, we exploit adaptive patch-based dictionaries to obtain substantially improved reconstruction performance for CSMRI. We propose a novel framework for simultaneously learning the dictionary and reconstructing the image from highly undersampled k-space data. Dictionaries adapted to the specific image that is to be recovered have been shown to provide state-of-the-art results in image denoising [31]. Learning a dictionary from a fraction of the image pixels has been studied in the context of image inpainting – that is, filling in missing or badly corrupted samples in an image [28, 31]. Unlike inpainting, in MRI the available partial data is in k-space rather than in the image domain – a fundamental difference. In this thesis, we learn an image-patch dictionary from a small number of k-space samples. Such a measurement-adapted dictionary can give rise to superior sparsities for every image instance thereby leading to substantially higher undersampling rates/speedups for CSMRI. Our approach thus combines the advantages of patch-based dictionaries in a completely adaptive framework, making it possible for the reconstruction performance to approach its fundamental limits.

Figure 1.1 shows a reconstruction instance using the proposed adaptive dictionary. The result is clearly devoid of the many artifacts seen in the CSMRI reconstruction described earlier and shown in Figure 1.1 (c) and (e), despite the high undersampling factor. The magnitude of the image reconstruction error (using the same scale as that in the CSMRI reconstruction in Figure 1.1(e)) also shows pixel errors of much smaller magnitude.

Our framework can automatically update (adapt) a dictionary learned \textit{a priori} from a fully sampled reference image(s) to include new features in the current scan data. Additionally, our method can give promising improvements in reconstruction even in the absence of a reference image by directly adapting to the current image content. The dictionary in Figure 1.1 was learned directly from the sampled data without utilizing any reference images. Dispensing with frequent fully sampled reference images in MRI not only improves throughput, but also enables applications in which it is difficult
to acquire relevant reference images.

The implicit goal of CSMRI is to perform accurate k-space interpolation using only a subset of the samples. However, as we determined empirically, explicit k-space interpolation (by trying to learn a dictionary for the k-space data and performing k-space “inpainting”) leads to poor reconstructions due to the lack of local structure in k-space. Instead, in this work, we perform implicit interpolation by employing learned dictionaries in the image domain.

The rest of this thesis is organized as follows. Chapter 2 details prior work in CSMRI and dictionary learning. In Chapter 3, the problem formulation for MRI reconstruction based on adaptive dictionary learning is described along with the algorithm for solving such a problem and its relevant properties. Chapter 4 demonstrates the performance of the algorithm on various examples, using a variety of sampling schemes, undersampling factors, and noise levels. The undersampling limit achievable with our adaptive algorithm as compared to previous CSMRI methods is also studied. Moreover, the robustness of the algorithm to parameter selection is also presented. In Chapter 5, we describe the usefulness of a dictionary learned at multiple scales. The problem formulation for MRI reconstruction based on multiscale (or multiple scale) adaptive dictionary learning is then presented along with the algorithm for solving such a multiscale problem and its relevant convergence and computational properties. The performance of the multiscale algorithm is shown on various examples. The ability of the multiscale dictionary to enable superior reconstructions compared to the single scale dictionary of Chapter 3 is demonstrated. In Chapter 6, we conclude with suggestions for future work.
1.1 Figures

Figure 1.1: A comparison of non-adaptive CS versus adaptive patch based CS: (a) Axial T2-weighted image of the brain (reference), (b) Sampling mask in k-space with 20-fold undersampling, (c) CSMRI reconstruction [16] with wavelets and TV, (d) Reconstruction using the proposed adaptive framework, (e) Magnitude of reconstruction error for (c), (f) Magnitude of reconstruction error for (d).
CHAPTER 2
BACKGROUND AND RELATED WORK

In this work, we use \( x \in \mathbb{C}^P \) to represent, as a vector, the \( P \)-pixel 2D complex image to be reconstructed, and \( y \in \mathbb{C}^m \) represents the k-space measurements. The two are related (in the absence of noise) as \( F_u x = y \), where \( F_u \in \mathbb{C}^{m \times P} \) is the undersampled Fourier encoding matrix. Undersampling occurs whenever the number of k-space samples is less than the number of unknowns \( (m < P) \).

2.1 CSMRI

Compressed sensing reconstructs the unknown \( x \) from the measurements \( y \), or equivalently solves an underdetermined system of linear equations \( F_u x = y \) by minimizing the \( l_0 \) quasi norm (i.e., the number of non-zeros) of the sparsified image \( \Psi x \), where \( \Psi \in \mathbb{C}^{T \times P} \) represents a global, typically orthonormal, sparsifying transform for the image. For example, \( \Psi \) may be the wavelet transform, so that \( \Psi x \) corresponds to the wavelet coefficients of \( x \), assumed to be sparse (mostly zero). The corresponding optimization problem is

\[
\min_x \|\Psi x\|_0 \quad \text{s.t.} \quad F_u x = y
\]

This \( l_0 \) problem, often known as the sparse coding problem because it corresponds to finding a sparse code \( x \) for the given vector \( y \) using the codebook \( F_u \), is NP-hard (non-deterministic polynomial-time hard). However, there are greedy algorithms to solve this problem such as orthogonal matching pursuit (OMP)[11, 34]. Alternatively, the \( l_0 \) quasi norm is replaced by its convex relaxation, the \( l_1 \) norm [35], at which point the problem can be solved via linear programming [7]. When the measurements are noisy, the CS problem is solved using basis pursuit denoising [36]. Under certain conditions, these algorithms can be guaranteed to provide the correct solution, or provide it with high probability. In practice, these algorithms usually perform better.
than predicted by the available theory. Other algorithms, for which similar theoretical performance guarantees may not be available, often provide even better empirical performance [37, 38, 39, 40, 41].

Compressed sensing has been applied to a variety of MR modalities such as static MRI [16, 19, 20], dynamic MRI [17, 21, 42, 43], parallel imaging [44, 45], and perfusion imaging and diffusion tensor imaging (DTI) [46]. For dynamic heart imaging, Lustig [17] uses the wavelet transform in the spatial dimension and the Fourier transform in the temporal as sparsifying transforms. Jung et al. [43, 47] use a framework with prediction and residual encoding for dynamic MRI. RIGR (reduced-encoding imaging by generalized-series reconstruction) and motion estimation/compensation are shown to sparsify residual signals well. Sparsifying transforms such as TV and wavelets have been used [45, 44] for parallel imaging as well. In this work, we restrict our attention to CS for static MRI and study it in detail.

The typical formulation of the CSMRI reconstruction problem uses $l_1$ relaxation of the $l_0$ quasi norm, and accounts for the noise in the k-space measurements in the following Lagrangian setup [16]:

$$
\min_x \| F_u x - y \|_2^2 + \lambda \| \Psi x \|_1
$$

(2.2)

This problem formulation involves a global sparsity measure and an analytical, fast sparsifying transform $\Psi$. However, common transforms such as wavelets result in artifacts such as Gibbs ringing in the result. Hence, a TV penalty, corresponding to a finite difference approximation to a gradient sparsifying transform, is typically added to the formulation to enforce spatial homogeneity.

Other sparsifying transforms have been proposed for CSMRI including a dual-tree complex wavelet transform [48] and overcomplete contourlets [49]. Wavelets can recover point-like features while contourlets recover curve-like image features. A combination of wavelets, contourlets, and TV was shown to increase peak signal to noise ratio (PSNR) by 1.3 dB over wavelets + TV [50]. A refinement of the wavelet sparsifying transform uses a Gaussian scale mixture (GSM) model [20] to exploit the dependencies between wavelet coefficients in CSMRI. The effectiveness of this approach is discussed in the next paragraph.

Numerous algorithms to solve the CSMRI reconstruction problem (2.2)
have been presented. Lustig et al. [16] solve (2.2) using a non-linear con-
jugate gradient descent algorithm with backtracking line search. The focus
of that paper is primarily on Cartesian sampling schemes and results are
shown for brain imaging and angiography. Ma et al. [23] present an algo-
rithm based on an iterative operator-splitting framework that exploits fast
wavelet and Fourier transforms to enable fast reconstruction. However, the
results shown at higher undersampling factors (> 3 times) suffer from many
artifacts. Qu et al. [51] use the contourlet transform within an efficient soft
iterative thresholding algorithm for CSMRI and show improvements of about
1.5 dB over wavelet CSMRI. Kim et al. [20] combine the GSM model with
Iterative Hard Thresholding (IHT) to perform the reconstruction. However,
the improvements shown in this work on in vivo data are not quantified and
appear small, and only a low undersampling factor (about 2) is used.

In an attempt to improve the performance of CSMRI beyond that ob-
tained using $l_1$ relaxation or greedy algorithms to encode sparsity, several
authors turned to nonconvex relaxations or approximations. Chartrand [18]
used the $l_p$ quasi norm (where $0 < p < 1$), which approximates the $l_0$ quasi
norm better than the $l_1$ norm. For an image of the uterus, $p < 1$ is demon-
strated to improve the SNR of the reconstruction by a small amount (of 0.9
dB). Trzasko et al. [52, 53] propose an alternative reconstruction based on a
homotopic approximation of the $l_0$ quasi norm. Their algorithm involves it-
 erative alternation between bilateral filtering and projection of the measured
k-space samples. The same authors claim that homotopic $l_0$ minimization
can provide higher undersampling factors than $l_1$ norm based methods [19].
However, their reconstructions for spine and wrist images still contain some
prominent aliasing and other artifacts at the undersampling factors of 4 - 7.5
simulated.

2.2 Dictionary Learning

The non-adaptive CSMRI techniques are limited by the degree of undersam-
pling at which they can still give clinically useful reconstructions. Adaptive
dictionaries lead to higher sparsities and hence to potentially higher under-
sampling factors in CSMRI. The key to adapting the dictionary to the data
is dictionary learning. Given an image $x \in \mathbb{C}^P$, $x_{ij} \in \mathbb{C}^n$ is the vector repre-
sentation of the square 2D image patch of size $\sqrt{n} \times \sqrt{n}$ pixels, indexed by the location of its top-left corner, $(i, j)$, in the image. We use $D \in \mathbb{C}^{n \times K}$ to represent the image patch-based dictionary. The dictionary $D$ has $K$ atoms (columns), each an $n$-vector corresponding to a $\sqrt{n} \times \sqrt{n}$ "elemental patch." It is assumed that each patch $x_{ij}$ can be approximated by a linear combination $D\alpha_{ij}$ of dictionary atoms, where $\alpha_{ij} \in \mathbb{C}^K$ is sparse. Accordingly, we say that $\alpha_{ij}$ is the sparse representation of $x_{ij}$ with respect to $D$. When $K = n$, the number of atoms equals the dimension of a patch, and $D$ is said to be a basis. Otherwise, when $K > n$, the dictionary is said to be overcomplete.

Figure 2.1 shows an example overcomplete dictionary of size $49 \times 98$ where the atoms (magnitudes) are displayed as $7 \times 7$ elemental patches and are learned from the 20-fold undersampled k-space data of Figure 1.1 using the formulation described later in Chapter 3.

Dictionary learning (DL) aims to solve the following optimization problem:

$$\min_{D, A} \sum_{ij} \| R_{ij}x - D\alpha_{ij} \|_2^2 \quad s.t. \quad \| \alpha_{ij} \|_0 \leq T_0 \ \forall \ i, j \quad (2.3)$$

Matrix $R_{ij} \in \mathbb{C}^{n \times P}$ represents the operator that extracts the patch $x_{ij}$ from $x$ as $x_{ij} = R_{ij}x$. The $l_0$ quasi norm is used to encode the sparsity of the patch representation, and $T_0$ is the required sparsity level. $A$ is used to denote the set $\{\alpha_{ij}\}_{ij}$ of sparse representations of all patches. This learning formulation minimizes the total fitting error of all image patches with respect to the dictionary, subject to sparsity constraints.

The DL problem (2.3) is NP-hard, because for fixed $D$ and $x$ it reduces to the sparse coding problem. However, unlike the latter problem (or equivalently, the compressive sensing problem (2.1)), even with a convex relaxation of the $l_0$ quasi norm, the optimization problem for learning is non-convex in the unknown variables, making it harder. Numerous algorithms have been proposed to solve such a dictionary learning problem [25, 26, 54, 55, 56]. These algorithms typically alternate between finding the dictionary $D$ (dictionary update step) and the sparse representations $A$ (sparse coding step). In particular, the K-SVD algorithm [25] has been widely used in many applications [27, 28, 29]. It performs the dictionary update step in a sequential manner where each column/atom of $D$ is updated jointly with the corresponding representation coefficients for the patches that currently use it.
This sequential update is similar to the one performed in the K-means algorithm [57] but involves computing K singular value decompositions (SVDs); one for each atom. Hence, the name K-SVD. Older algorithms [26] do not perform a sequential update of $D$ and are shown to provide inferior learning as compared to K-SVD [25].

Dictionary learning for medical imaging has received only recent attention. The learning is typically done using reference images. In an application to ultrasound breast imaging [58], the dictionary is trained from reference MRI scans using maximum likelihood dictionary learning [55], which is an older algorithm than K-SVD.

Bilgin et al. [59] learn a K-SVD patch dictionary from a fully sampled reference MRI image slice and use it in IHT to recover undersampled test image slices. However, the k-space undersampling factor used is low (about 2) and the improvement in reconstruction SNR compared to wavelet-IHT is small (1.5 dB). Chen and Ye [60] learn a patch-based K-SVD dictionary from fully sampled reference images, and employ the $l_1$ norm for sparsity in the reconstruction. The results show improved performance with the learned dictionary over CSMRI with wavelets as sparsifying transform. However, the improvement is small (SNR increases by 1.6 dB compared to CS with wavelets), and the error map shown in the work has considerable structure indicating loss of features in the reconstructed image. These results suggest that a dictionary learned from a reference image would not be able to effectively sparsify new features in the current scan.

Otazo and Sodickson [61] learn a one-dimensional dictionary for MRI, using K-SVD, directly from the columns of an initial reconstruction obtained by CSMRI with 1D-wavelets. In comparison to the CSMRI result, the learned dictionary is shown to produce better reconstruction. However, a 1D dictionary is a very restricted dictionary that cannot exploit the 2D local structure of the image. The number of training patches for such a dictionary would be equal to the number of image columns, which is insufficient for training. As might be therefore expected, the reconstructed image in this work is seen to have visible artifacts even at 2.5-fold undersampling.
2.3 Figure

Figure 2.1: The atoms of a $49 \times 98$ dictionary shown as $7 \times 7$ elemental patches. The magnitudes are displayed.
CHAPTER 3

PROBLEM FORMULATION AND ALGORITHM

3.1 Problem Formulation

The problem formulation for CSMRI based on dictionary learning needs to have two characteristics. It should be able to enforce sparsity of the patches of the reconstructed image in an adaptive dictionary, and produce a reconstruction that is consistent with the available k-space data. It should also be able to avoid artifacts typically seen in the zero-filled Fourier reconstruction. Artifacts arise chiefly due to two reasons: undersampling of k-space, and noise in the samples. Undersampling of k-space causes aliasing in the image domain, and independent and identically distributed (i.i.d.) zero-mean complex Gaussian noise in the measured k-space samples (undersampled) translates to colored Gaussian noise in the image domain.

A possible formulation is

\[
(P0) \min_{x, D, A} \sum_{ij} \| R_{ij} x - D \alpha_{ij} \|^2_2 + \nu \| F_u x - y \|^2_2 \tag{3.1}
\]

subject to \( \| \alpha_{ij} \|_0 \leq T_0 \forall i, j \)

The first term in the cost function captures the quality of the sparse approximations of the image patches with respect to the dictionary \( D \). The second term in the cost enforces data fidelity in k-space. The patch sparsity constraint is the same as in generic dictionary learning (2.3). The weight \( \nu \) in our formulation depends on the measurement noise level (\( \sigma \)) as \( \nu = \frac{\lambda}{\sigma} \) where \( \lambda \) is a positive constant. This makes it more robust to noise. Such a form of the weight uses prior knowledge of the measurement process and has been shown to work well in image denoising [31]. Previous CSMRI methods such as [16] without explicit noise-adapted data-fidelity weights struggle to enforce a good trade-off between data consistency and denoising. The
proposed formulation considers the measurement scenario with known noise level. An estimate of the noise level or the observed noise level can be used when the exact value is unknown.

The problem formulation for the simpler case of noiseless measurements can be obtained by dropping the data fidelity term from the cost and adding it instead as a constraint, \( F_u x = y \). This case represents the limiting behavior of Problem (P0) when \( \sigma \to 0 \), or \( \nu \to \infty \), and is useful for high SNR of the measurement process.

We use periodically positioned, overlapping 2D image patches in our formulation. The \textit{overlap stride} \( r \) is defined to be the distance in pixels between corresponding pixel locations in adjacent image patches. The patches are said to have maximum overlap for \( r = 1 \). In this case, every image pixel \((i, j)\) (except the ones near the right and bottom image boundaries) would be the top-left corner of a square 2D patch. If patches are assumed to “wrap around” at image boundaries, then pixels at the right and bottom image boundaries can also constitute the top-left corners of patches. The patches in this scenario begin at the image boundary and wrap around on the opposite side of the image. When there is wraparound and \( r = 1 \), each pixel in the image belongs to \( n \) different patches, where \( n \) is the number of pixels in a patch.

Our formulation (P0) is capable of both designing an adaptive dictionary, and also using it to reconstruct the underlying image. This is done using only the undersampled k-space measurements, \( y \). However, just as the generic dictionary learning problem (2.3), Problem (P0) of simultaneous reconstruction and dictionary learning is NP-hard, and non-convex even when the \( l_0 \) quasi norm is relaxed to an \( l_1 \) norm. Alternative (but not necessarily easier) problem formulations can also be generated by modifying Problem (P0). For instance, the sparsity constraints can be combined into the cost function as penalties using Lagrange multipliers. However, in this work, we focus on Problem (P0).

The adaptive patch-sparsity based formulation in (P0) can potentially remove aliasing and noise while also learning local image features effectively. Overlapping patches are used to create an additional averaging effect for artifact removal. These aspects of the formulation make it very useful for the CSMRI setting.
3.2 Algorithm

Problem (P0) is solved using an alternating minimization procedure. In one step of this alternating scheme, \( x \) is assumed fixed, and the dictionary and sparse representations of the patches are jointly learned. In the other step, the dictionary and sparse representations are fixed, and \( x \) is updated to satisfy data consistency. These two steps are further detailed in the following subsections.

3.2.1 Dictionary Learning Step

In this step, Problem (P0) is solved with fixed \( x \). This corresponds to the sub-problem

\[
(P1) \quad \min_{D,A} \sum_{ij} \| R_{ij} x - D_{ij} \|_2^2 \\
\text{s.t.} \quad \| d_k \|_2 = 1 \forall k; \quad \| \alpha_{ij} \|_0 \leq T_0 \forall i,j
\]

The cost function for learning takes into account the fitting errors of all the overlapping patches using the sparsifying dictionary. Alternatively, similarly to the application of DL in denoising [31], we use only a fraction \( \delta \) of all patches to train the dictionary. The columns of the designed dictionary (represented by \( d_k, 1 \leq k \leq K \)) are additionally constrained to be of unit norm. An optimal dictionary \( D_0 \) for the learning problem that does not enforce such a norm constraint can be scaled by some non-zero \( \tau \in \mathbb{C} \), and the corresponding optimal set of sparse representations \( A_0 \) can be scaled by \( \frac{1}{\tau} \), without affecting the cost function in (P1). Then, all such scaled versions of \( D_0 \) and \( A_0 \) would also be optimal. However, as \( |\tau| \rightarrow 0 \), the sparse approximations can become unbounded. Hence, a unit norm constraint is enforced on the columns of the dictionary \( D \) in order to avoid such a scaling ambiguity [62]. The K-SVD algorithm [25] is used to learn the dictionary, \( D \). It obtains each atom of \( D \) as the first left singular vector of an SVD, thus directly enforcing the unit norm constraint. Once the dictionary is learned, sparse coding is performed on all patches to determine the \( \alpha_{ij} \).
3.2.2 Updating the Reconstruction

In this step, Problem (P0) is solved with fixed dictionary and sparse representations. The corresponding update problem is

\[
(P2) \quad \min_x \sum_{ij} \| R_{ij} x - D_{ij} \|^2_2 + \nu \| F_u x - y \|^2_2
\]

Problem (P2) is a simple least squares problem admitting an analytical solution. The least squares solution satisfies the normal equation.

\[
\left( \sum_{ij} R_{ij}^T R_{ij} + \nu F_u^H F_u \right) x = \sum_{ij} R_{ij}^T D_{ij} + \nu F_u^H y
\]

The superscript $H$ denotes the Hermitian transpose operation, and the superscript $T$ is used instead of $H$ when the operand is real. Solving (3.4) directly can be tedious because it requires to invert the $P \times P$ matrix premultiplying $x$. In general, it requires $O(P^3)$ operations which is impractical even for a modest image size of $P = 256 \times 256$. Fortunately, the solution simplifies by using the structure of the different quantities.

The term $\sum_{ij} R_{ij}^T R_{ij} \in \mathbb{C}^{P \times P}$ is a diagonal matrix, where the diagonal entries correspond to image pixel locations and their values are equal to the number of overlapping patches contributing at those pixel locations. The diagonal entries become all equal, and $\sum_{ij} R_{ij}^T R_{ij} = \beta I_P$ (where $I_P \in \mathbb{C}^{P \times P}$ is an identity matrix), if we assume that patches wrap around at image boundaries. In particular, $\beta = n$ when the overlap stride $r = 1$ for the patches. When patches are restricted to the field of view (FOV) (no wraparound), the number of patches contributing to pixels near image boundaries would be less than the number contributing to pixels in the rest of the image and $\sum_{ij} R_{ij}^T R_{ij} \approx \beta I_P$. The wraparound assumption has been used before for dictionary design [30], and we use it here as well in order to arrive at a simple solution.

The term $\sum_{ij} R_{ij}^T D_{ij}$ with an additional scaling of $\frac{1}{\beta}$ represents the patch averaged result. The patches having been approximated by the learned dictionary are averaged at their respective locations in the image. The intensity value at each pixel is obtained by averaging the contributions of the various patches that cover it.

The next simplification is obtained by transforming from image space to
Fourier space. Let $F \in \mathbb{C}^{P \times P}$ denote the full Fourier encoding matrix normalized such that $F^H F = I_P$. Then, $Fx$ represents the full k-space data. Substituting into (3.4) yields

$$
(F \sum_{ij} R_{ij}^T R_{ij} F^H + \nu F F_u^H F_u F^H) F x = F \sum_{ij} R_{ij}^T D \alpha_{ij} + \nu F F_u^H y
$$

(3.5)

The matrix $FF_u^H F_u F^H$ is a diagonal matrix consisting of ones and zeros. The ones are at those diagonal entries that correspond to sampled locations in k-space. Vector $FF_u^H y$ represents the zero-filled Fourier measurements.

Under the wraparound assumption, $F \sum_{ij} R_{ij}^T R_{ij} F^H = \beta I_P$ and the matrix pre-multiplying $Fx$ in (3.5) becomes diagonal and trivially invertible. Both sides of the equation can be divided by the constant $\beta$, and the constant can be absorbed into the weight $\nu$ (using $\lambda' = \frac{\lambda}{\beta}$).

The “patch averaged result” is transformed to the Fourier domain producing

$$
S = \frac{F \sum_{ij} R_{ij}^T D \alpha_{ij}}{\beta}
$$

The solution to (3.5) is then

$$
Fx(k_x, k_y) = \begin{cases} 
S(k_x, k_y) & , (k_x, k_y) \notin \Omega \\
S(k_x, k_y) + \nu S_0(k_x, k_y) & , (k_x, k_y) \in \Omega
\end{cases}
$$

(3.6)

where $Fx(k_x, k_y)$ represents the updated value at location $(k_x, k_y)$, $S_0 = FF_u^H y$ represents the zero-filled k-space measurements, and $\Omega$ represents the subset of k-space that has been sampled.

Equation (3.6) uses the dictionary interpolated values for the non-sampled Fourier frequencies, and fills back the sampled frequencies, albeit with averaging in the presence of noise. For the noiseless case ($\nu \to \infty$), the sampled frequencies are merely restored to their measured values by this operation. The reconstruction, $x$, is then obtained by IFFT of $Fx$.

The proposed algorithm is summarized in Figure 3.1. The algorithm is initialized with a zero-filled Fourier reconstruction, $F_u^H y$. If the sampling scheme is non-Cartesian, this is computed rapidly with FFTs using gridding [63]. Simple FFT without gridding sufficed for the pseudo-radial sampling pattern employed in the numerical experiments in Chapter 4, because the samples are chosen to fall on a Cartesian grid. Recall that it was assumed that
patches wrap around at image boundaries to arrive at this elegant solution. The proposed algorithm outline of Figure 3.1 was implemented with patches being restricted to the FOV. We noticed little difference in reconstructions obtained this way as compared to solving (3.4) using a conjugate gradient iterative solver. The latter method is considerably slower.

3.3 Convergence

The proposed algorithm alternates between learning the dictionary and sparse representations, and estimating the reconstruction. Solving Problems (P1) and (P2) iteratively leads to monotonic decrease in the cost function of (P0). Since the cost function is non-negative, it converges. Thus, better sparse reconstructions (in the sense of the cost function of (P0)) are learned at each iteration beginning with the zero-filled result. Empirically (see Chapter 4), the iterates $x^k$ (indexed by iteration number) converge as well, but this is yet to be proved rigorously. Another open question involves conditions for perfect reconstruction (in the zero noise case). Numerical experiments suggest that the algorithm can enable perfect reconstruction at reasonably higher undersampling factors than current CSMRI methods.

The stopping criterion for the algorithm can be the value of the objective function. It can also be the norm of the reconstruction difference between successive iterations. Simulations presented in Chapter 4 show that the algorithm converges rapidly and a fixed number of iterations can suffice in practice.

3.4 Parameters

The algorithm has a few design parameters, notably the size of patches ($\sqrt{n} \times \sqrt{n}$), sparsity threshold for each patch ($T_0$), the number of atoms or degree of overcompleteness of the dictionary ($K$), fraction of patches used for training ($\delta$), patch overlap stride ($r$), and the factor $\lambda$ in the data consistency weight. We study the sensitivity of the algorithm to these parameters empirically in the next chapter.
3.5 Algorithm Complexity

The algorithm alternates back and forth between the image domain and k-space. The solution to Problem (P1) involves learning the dictionary $D$ from a fraction $\delta$ of all $N$ patches and using it to obtain sparse approximations of the $N$ patches. The dictionary learning (DL) step uses the K-SVD algorithm and employs OMP for sparse coding. The computation is dominated by sparse coding which scales as $O(\delta NK nT_0 J)$, where $J$ is the number of iterations in learning. The cost of sparse coding all $N$ patches using the learned dictionary is $O(K nT_0 N)$. The costs of DL and the final sparse coding step are balanced by choosing $\delta J = 1$. The computational complexity of the reconstruction update step (P2) is dominated by 2 FFTs which have a cost of $O(P \log P)$. The other computations for (P2), namely patch averaging and sampled frequency averaging, are fast, low complexity operations.

Under the assumptions of patch wraparound and overlap stride $r = 1$, the number of patches $N = P$, and $O(K nT_0 P) \gg O(P \log P)$ typically. This indicates that Problem (P1) dominates the computational cost and the main speed bottleneck is due to the various sparse coding steps within (P1). Using faster sparse coding algorithms along with parallel processing of patches can improve speed significantly. Reduction in the number of overlapping patches by increasing the overlap stride $r$ ($N = P/r^2$, with patch wraparound) can reduce the complexity further. Other parameters such as patch size, sparsity, and number of dictionary atoms may compromise the solution if made too low.
Algorithm 1

Input: \( y \) - k-space measurements
Output: \( x \) - Reconstructed MR image
Initialization: \( x = x_0 = F_u^H y \)
Iteration:

1. Learn dictionary and sparse representations for patches of \( x \)
2. Update \( x \): Each pixel value obtained by averaging contributions of patches that cover it
3. \( S \leftarrow FFT(x) \)
4. Restore Sampled frequencies to update \( S \) per (3.6)
5. \( x \leftarrow IFFT(S) \)

Figure 3.1: Algorithm to reconstruct MR images from undersampled k-space measurements using adaptive dictionaries.
CHAPTER 4
EXPERIMENTS

In this chapter, we first detail the experimental framework and then show the performance of our algorithm on numerous examples, using a variety of sampling schemes and noise levels.

4.1 Framework

In this section, the performance of the proposed algorithm is demonstrated at a variety of undersampling factors, with and without noise. The images used in the experiments are *in vivo* MR scans of size $512 \times 512$ (many of which are courtesy of [24]). Sampling schemes used in the experiments include 2D random sampling [19], Cartesian sampling with random phase encodes (1D random), and pseudo radial sampling [60]. In the latter scheme, samples are taken on a $512 \times 512$ Cartesian grid, at the points nearest to radial lines uniformly spaced in angle. Similarly to prior work on CSMRI [19, 61, 23, 18, 20], the CS data acquisition was simulated by subsampling the 2D discrete Fourier transform of the MR images (except in Figures 4.7, 4.10 where pulse sequences were used). Our reconstruction method is compared with a leading CSMRI method by Lustig et al. [16] (denoted as LDP), and the baseline zero-filling reconstruction. Other CSMRI methods reviewed in Chapter 2 offer only small improvements over LDP [16] and are hence not included in our comparisons. All implementations were coded in Matlab v7.8 (R2009a). Computations were performed with an Intel Core i5 CPU at 2.27 GHz and 4 GB memory, employing a 64-bit Windows 7 operating system. The Matlab implementation of LDP [16] available from the author’s website [64] was used in our comparisons. We used the built-in parameter settings in that implementation which performed optimally in our experiments.

In the experiments with or without noise, the nominal values of the various
parameters were set as \( n = 36, K = n = 36, T_0 = 0.15 \times n \approx 5, \lambda = 140 \). We worked with maximum patch overlap, \( r = 1 \). The learning stage (K-SVD) employed 10 iterations, \( 200 \cdot K \) patches, and a fixed sparsity of \( T_0 \). The K-SVD learning scheme requires an initialization for the dictionary [25]. We used the left singular vectors of the training data to form the initial \( 36 \times 36 \) dictionary. Alternative initializations involving analytical dictionaries such as wavelets were also observed to work well. After learning, the overlapping patches were each sparse coded using the sparsity threshold of \( T_0 \).

Real-valued dictionaries (as in [60]) were used for the simulated experiments with real-valued images and complex-valued dictionaries were used for the actual MR data experiments (Figures 4.7 and 4.10), in which the reconstructed image is usually complex. Complex dictionaries usually yield similar results for the real-valued simulation cases but typically with some more iterations. The algorithm was run for about 10-15 iterations for the simulated experiments, and for a few tens of iterations in Figures 4.7 and 4.10. Currently, the algorithm has an average run time of about 1.4 mins per iteration with complex dictionaries. We expect this time to decrease substantially with conversion of the code to C/C++, code optimization, and graphics processing unit (GPU) implementation.

The quality of the reconstruction is quantified using two metrics - PSNR, and a high frequency error norm. PSNR (in dB) is computed as the ratio of the peak intensity value of the reference image to the root mean square (RMS) reconstruction error relative to the reference image. This is a standard image quality measure in work on image compression and has been used in CSMRI before [51], along with the related metric of SNR (in dB) [18], [59]. The high frequency error norm (HFEN) is used to quantify the quality of reconstruction of edges and fine features. We employ a rotationally symmetric LoG (Laplacian of Gaussian) filter to capture edges. The filter kernel is of size \( 15 \times 15 \) pixels, and has a standard deviation of 1.5 pixels. HFEN is computed as the \( l_2 \) norm of the result obtained by LoG filtering the difference between the reconstructed and reference images. Needless to say, these metrics do not necessarily represent perceptual visual quality, which can only be accurately assessed by human visual observer studies. Nonetheless, large differences in these metrics typically correspond to visually perceptible differences.
4.2 Noiseless Case

The noiseless scenario (Problem (P0) with $\nu \to \infty$) can be solved by performing a direct frequency fill-in/restore at the reconstruction update step of our algorithm. One can also solve Problem (P0) with a large value for $\nu$. We study the noiseless scenario first in order to see the best, or ideal, performance that can be obtained with our formulation and algorithm.

Figure 4.1 shows the performance of the algorithm on a brain image employing 2D variable density random sampling (sampling mask shown later in Figure 4.2) of k-space at 5-fold undersampling. The zero-filled Fourier reconstruction has significant undesirable artifacts due to aliasing. The LDP algorithm [16, 64] is unable to remove the aliasing well. Some of the artifacts in the zero-filled reconstruction persist in the CSMRI-LDP result. This phenomenon can also be seen in later Figures (4.4, 4.9, 4.11, 4.10). In contrast, the result with dictionary learning (DLMRI) is seen to be free from artifacts and close to perfect reconstruction. Our algorithm was executed with a fixed number of 10 iterations for this case, to study its performance. The reconstruction error magnitude, i.e. $|x_{\text{Recon}} - x_{\text{Ref}}|$, was thresholded at 5% of the maximum reference image intensity for both the LDP method [16, 64] and DLMRI. This helps identify regions of high error which are overlaid on the reference image in green. The CSMRI result [16, 64] can be seen to have more than 5% error magnitude in many regions (more than 50% of image) while the DLMRI result is almost free of such high magnitude errors (only 0.07% of all pixels).

The norm of the reconstruction difference between successive iterations ($||x^k - x^{k-1}||_2$) shown in Figure 4.2 can be seen to converge quickly. PSNR and HFEN are also plotted in Figure 4.2. The PSNR for DLMRI after 10 iterations of the algorithm is nearly 18 dB higher than the corresponding value for the LDP method [16, 64]. As expected, the zero-filling reconstruction has the worst PSNR. HFEN is also lower for DLMRI than for LDP [16, 64], indicating the superior performance of DLMRI in capturing edges and fine features. DLMRI was also observed to reconstruct the non-sampled frequencies in k-space much better than LDP [16, 64]. The reconstruction $x$ showed very little visual change after 5 or 6 iterations of the algorithm. The various plots also show fast convergence and, thus, quantitatively demonstrate this fact.
In Figure 4.3, variable density cartesian sampling with 4-fold undersampling is employed on a non-contrast MRA of the Circle of Willis. The reconstruction with DLMRI is clearer, sharper than with LDP [16, 64], and relatively devoid of aliasing artifacts. In particular, the vessels in the bottom half of the DLMRI reconstruction appear less obscured than those in the LDP [16, 64] result. The magnitudes of the reconstruction error for DLMRI and LDP [16, 64] are shown on the same scale. The error image of LDP [16, 64] is seen to have significantly more structured errors, indicating loss of features.

In Figure 4.4, Cartesian sampling is employed with 7.11-fold undersampling. LDP [16, 64] is unable to remove the large aliasing artifacts seen in the zero-filling result. DLMRI, on the other hand, produces an alias-free reconstruction that looks close to the reference. The PSNR and HFEN error metrics also demonstrate the promising performance of DLMRI for this case. A small degree of smoothing in the reconstruction as compared to the reference appears, however, inevitable at high undersampling factors as perfect k-space interpolation may not be achieved.

A low resolution reconstruction obtained by sampling the central k-space (cartesian) phase encoding lines at 7.11-fold undersampling is shown in Figure 4.5. This reconstruction with a PSNR of 29.5 dB has some visible ringing artifacts marked with green arrows. The DLMRI algorithm when executed with only the central k-space data produced a result with a PSNR of 30.7 dB. The small improvement is due to the complete absence of high frequency information in the phase encoding direction. However, some ringing is reduced compared to the low resolution reconstruction due to the local sparsity constraint employed in our formulation. On the other hand, the LDP [16, 64] result (not shown here) had a poorer PSNR of 29.3 dB for this case.

Figure 4.6 shows results for the same image and undersampling factor as in Figure 4.4, but employing variable density 2D random sampling and pseudo radial sampling (also employed and shown later in Figure 4.11 (c)) respectively. The DLMRI method for both these sampling schemes is seen to outperform LDP [16, 64]. The adaptive dictionary produces a more significant reduction in aliasing artifacts (i.e. incoherent aliasing artifacts for 2D random sampling, and streaking artifacts for pseudo radial sampling) compared to non-adapted ones. The magnitudes of the reconstruction errors displayed for the pseudo-radial sampling case are also smaller for DLMRI than for LDP
Figures 4.4, 4.5, and 4.6 show that DLMRI can perform well with a variety of sampling schemes at a given undersampling factor.

In Figure 4.7, T2-weighted k-space data of a brain was acquired using a Cartesian FSE sequence. Randomly undersampled phase encodes of the 2D FSE were obtained in order to test the performance of the proposed reconstruction algorithm. This data was collected by the authors of [16] and made available at the author’s website [64]. The reconstructions with the LDP [16, 64] and DLMRI methods at 2.5-fold undersampling do not show much visual difference. However, when the phase encodes were undersampled further, the reconstructions with the two methods displayed notable differences in visual quality. While the LDP [16, 64] reconstructions at 4 and 5 fold undersampling display visible aliasing artifacts along the phase encoding direction (horizontal in the image plane), the DLMRI reconstructions are clear and artifact-free at these higher undersampling factors.

### 4.3 Performance with Noise

The noisy case involves weighted averaging in k-space during the reconstruction update step of the algorithm (3.6). Figure 4.8 demonstrates the performance of our algorithm on the reference image of Figure 4.1 using Cartesian sampling at 5.23-fold undersampling. Zero-mean complex white Gaussian noise of standard deviation, $\sigma = 18.8$ was added in k-space. The fully sampled noisy image is shown along with the magnitude of the noise-only image, and can be observed to be considerably noisy compared to the reference. The PSNR of the noisy image with respect to the reference is about 30.68 dB. The reconstruction with LDP [16, 64] is unable to sufficiently remove the aliasing and noise seen in the zero-filled result (shown in Figure 4.9). Our algorithm, on the other hand, provides a good reconstruction. The magnitude image of the reconstruction error for DLMRI shows pixel errors of much smaller magnitude and less structure than that of LDP [16, 64].

PSNR and HFEN are plotted over iterations in Figure 4.9. The PSNR of the DLMRI result (with respect to the noise-free reference) is about 4.1 dB higher than that of LDP [16, 64]. It is also higher than the PSNR of both the fully sampled noisy image and the zero-filled result, indicating good denoising and aliasing removal. The HFEN metric is also better for DLMRI. The
results indicate the promising performance of our formulation/algorithm in
the presence of a reasonable amount of noise. The algorithm was executed
with 10 iterations and the convergence (i.e. norm of the reconstruction dif-
ference between successive iterations) is as rapid as for some of the previous
noiseless cases.

In Figure 4.10, a phantom \( (256 \times 256) \) was scanned using a 2D Cartesian
GRE sequence. The scanned data was noisy and is available at [64] (also
used by [16]). The k-space was undersampled by a factor of 4 by randomly
choosing phase-encode lines. The parameter \( \nu \) was set at 0.7 based on the
observed noise. The reconstruction with LDP [16, 64] retains some of the ar-
tifacts seen in the zero-filled reconstruction. On the other hand, the DLMRI
reconstruction displays some of the image features better and is clearer, alias
free. It is also less noisy than the LDP [16, 64] reconstruction, indicating the
superior denoising ability of our framework.

Figure 4.11 involves a T2-weighted sagittal view of the lumbar spine with
pseudo radial sampling and 6.09-fold undersampling of the k-space data.
Complex Gaussian noise of \( \sigma = 14.2 \) was added to k-space. The reconstruc-
tions and error image magnitudes with DLMRI and LDP [16, 64] methods
are shown in Figure 4.12. The DLMRI reconstruction is seen to be better
than that of LDP [16, 64]. Some regions of error in the LDP [16, 64] result
have been indicated with arrows. The corresponding regions in the DLMRI
reconstruction are clearer and sharper. The reconstruction error magnitude
images show pixel errors of lower magnitude and structure for DLMRI than
for LDP [16, 64]. The trends in PSNR and HFEN were similar to Figure 4.9
with the exception that the PSNR for the LDP [16, 64] method (29.25 dB)
was worse than that of the fully sampled noisy image (34.26 dB). The PSNR
for DLMRI (35.6 dB) was 6.35 dB higher than that of LDP [16, 64], and 1.34
dB better than that of the fully sampled noisy image, in spite of using less
than one sixth of the data.

4.4 Evaluation of Undersampling Limit

In this experiment, the effective undersampling limit achievable with DLMRI
as compared to LDP [16, 64] is evaluated. The reference image in Figure 4.4
was used for the evaluation with 2D random sampling. I.i.d. zero-mean
complex Gaussian noise of $\sigma = 10.2$ was added to the k-space samples. The PSNR of the fully sampled noisy image was 35.3 dB. Reconstructions with DLMRI and LDP [16, 64] methods were obtained at numerous k-space undersampling factors. The variation of PSNR and HFEN as a function of the undersampling factor is shown in Figure 4.13.

The PSNR of DLMRI is high even at very high undersampling factors such as 20, indicating good removal of aliasing and noise. On the other hand, the PSNR of the LDP [16, 64] method is comparable to that of DLMRI only at a low undersampling factor of 2.5. This indicates that the LDP [16, 64] method is unable to remove aliasing and noise sufficiently at higher undersampling factors. The HFEN values for the LDP [16, 64] method are also higher than the corresponding values for DLMRI at all undersampling factors. HFEN for the LDP [16, 64] method at 2.5-fold undersampling is comparable to that of DLMRI at 4 or 6 fold undersampling.

Figure 4.14 shows reconstructions at some undersampling factors. The reconstruction for the LDP [16, 64] method at 2.5-fold undersampling is shown along with the magnitude of the reconstruction error. Some structure is seen in the error, indicating loss of features. The reconstruction for DLMRI is observed to be free of aliasing and noise even at 10-fold undersampling. However, there is some error observed at the edges which can be inferred from the higher value of HFEN at this factor. The reconstructions with DLMRI at 4, 6 fold undersampling shown in Figure 4.15 can be seen to be almost error-free. The results indicate that DLMRI can achieve almost 2.5-4 times higher undersampling factors than existing CSMRI methods at comparable reconstruction errors.

4.5 Parameter Evaluation

In this experiment, the sensitivity of the algorithm to parameter settings was evaluated by varying one parameter at a time while keeping the rest fixed at their nominal values. The reference image in Figure 4.4 was used for the evaluation with 2D random sampling and 10-fold undersampling. The parameters evaluated were the patch size ($n$), sparsity threshold ($T_0$), weight ($\lambda$), and the overcompleteness of the dictionary ($K$). PSNR and HFEN are plotted in Figure 4.16 over these parameters. The overlap stride ($r$), and
training fraction \((\delta)\) were observed to work well at their nominal values, and are not studied separately.

In Figure 4.16, when the sparsity threshold is increased from 2 to 5, both PSNR and HFEN improve. However, at higher sparsity levels such as 7, the performance degrades. The poorer performance at the low sparsity level of 2 is due to the loss of resolution and higher sparse coding residuals encountered at low sparsity. At high sparsity levels such as 7, the algorithm begins to allow aliasing artifacts in the reconstruction, thereby degrading performance. The performance with respect to patch size conforms to expectations. As the patch size is increased from \(4 \times 4\) to \(7 \times 7\), both PSNR and HFEN improve. However, the amount of improvement from patch size of \(6 \times 6\) to \(7 \times 7\) or higher is small as compared to going from \(4 \times 4\) to \(6 \times 6\). The increase in patch size also increases computation, thereby preventing us from working at patch sizes that are too high. It is to be noted that the nominal values for \(T_0\) and \(K\) depend on \(n\), and so the actual numerical values change with \(n\). The change in PSNR and HFEN with increase in overcompleteness is rather small, indicating that square dictionaries can suffice in practice. The K-SVD learning scheme for the overcomplete cases was initialized using a combination of left singular vectors of the training patches and some training patches. We evaluated \(\lambda\) at two different noise levels. Complex Gaussian noise of \(\sigma = 18.8\), and 0.01 was added to the k-space samples. At each of these noise levels, the variation of PSNR is plotted vs. \(\lambda \in [60, 600]\). The changes in PSNR are seen to be small in that range and the values are quite good around \(\lambda = 140\) used by us. The same value of \(\lambda = 140\) works well at the two very different noise levels spanning a range of 65 dB. This is due to the noise-adapted weighting that we use \(\left(\frac{1}{\sigma^2}\right)\). The slow variation in PSNR with \(\lambda\) also indicates that a rough estimate for the noise level \(\sigma\) would work when the actual value is unknown. The drop in reconstruction PSNR when \(\sigma\) is increased from 0.01 to 18.8 is only about 2.7 dB, which demonstrates that DLMRI is reasonably robust to noise.

The plots of Figure 4.16 indicate that the “nominal” parameter values work reasonably well. They also show that the algorithm is not overtly sensitive to the parameters and can be used with little or no tuning. The various experimental results demonstrated using different sampling schemes, noise levels, undersampling factors, and images also indicate the good performance of the set parameters.
Figure 4.1: Performance of the algorithm with 5-fold undersampled 2D variable density sampling pattern: (a) Reference MR image of the brain, (b) Reconstruction with zero-filling, (c) Reconstruction using LDP [16, 64] with wavelets and TV, (d) Reconstruction using DLMRI, (e) Regions of high error (> 5%) overlaid on the reference image in green for LDP [16, 64], (f) Regions of high error overlaid on the reference image in green for DLMRI (zoom-in is required to view them).
Figure 4.2: Performance of the algorithm in the experiment of Figure 4.1: (a) Sampling mask in k-space, (b) Norm of the reconstruction difference between successive iterations, (c) PSNR vs. iterations for DLMRI with comparison to LDP [16, 64] and zero-filling, (d) HFEN vs. iterations for DLMRI with comparison to LDP [16, 64].
Figure 4.3: Cartesian sampling: (a) Reference non-contrast MRA of the Circle of Willis, (b) Sampling mask in k-space with 4-fold undersampling, (c) Reconstruction using LDP [16, 64] with wavelets and TV, (d) Reconstruction using DLMRI, (e) Reconstruction error magnitude for LDP [16, 64], (f) Reconstruction error magnitude for DLMRI.
Figure 4.4: Variable density Cartesian sampling at 7-fold undersampling: (a) Reference image, (b) Reconstruction with zero-filling, (c) Reconstruction using LDP [16, 64] with wavelets and TV, (d) Reconstruction using DLMRI, (e) PSNR vs. iterations with comparison to LDP [16, 64] and zero-filling, (f) HFEN vs. iterations with comparison to LDP [16, 64].
Figure 4.5: Dense Cartesian sampling of center of k-space at 7-fold undersampling: (a) Zero-filled reconstruction, (b) Reconstruction using DLMRI.
Figure 4.6: 2D random and pseudo radial sampling at about 7-fold undersampling: (a) Reconstruction using LDP [16, 64] with wavelets and TV for 2D random sampling, (b) Reconstruction using DLMRI for 2D random sampling, (c) Reconstruction using LDP [16, 64] with wavelets and TV for pseudo radial sampling, (d) Reconstruction using DLMRI for pseudo radial sampling, (e) Magnitude of reconstruction error for (c), (f) Magnitude of reconstruction error for (d).
Figure 4.7: Cartesian Sampling: (a) Reconstruction using LDP [16, 64] with wavelets and TV at 2.5-fold undersampling, (b) Reconstruction using DLMRI at 2.5-fold undersampling, (c) Reconstruction using LDP [16, 64] with wavelets and TV at 4-fold undersampling, (d) Reconstruction using DLMRI at 4-fold undersampling, (e) Reconstruction using LDP [16, 64] with wavelets and TV at 5-fold undersampling, (f) Reconstruction using DLMRI at 5-fold undersampling.
Figure 4.8: Performance of the algorithm with noisy data for 2D variable density random sampling at 5-fold undersampling (compare to Figure 4.1): (a) Fully sampled noisy reconstruction, (b) Noise magnitude in (a), (c) Reconstruction using LDP [16, 64] with wavelets and TV, (d) Reconstruction using DLMRI, (e) Magnitude of reconstruction error in (c), (f) Magnitude of the reconstruction error for (d).
Figure 4.9: Performance of algorithm for the experiment of Figure 4.8 (compare to Figure 4.2): (a) Reconstruction with zero-filling, (b) Norm of the reconstruction difference between successive iterations, (c) PSNR vs. iterations for DLMRI with comparison to LDP [16, 64], Figure 4.8 (a), and zero-filling, (d) HFEN vs. iterations for DLMRI with comparison to LDP [16, 64].
Figure 4.10: Cartesian sampling with noisy data: (a) Sampling scheme with 4-fold undersampling, (b) Zero-filled reconstruction, (c) Reconstruction using LDP [16, 64] with TV, (d) Reconstruction using DLMRI.
Figure 4.11: Pseudo radial sampling: (a) Reference T2-weighted sagittal view of the lumbar spine, (b) Noisy fully sampled reconstruction, (c) Sampling mask in k-space with 6-fold undersampling, (d) Reconstruction with zero-filling.
Figure 4.12: Reconstructions for the experiment of Figure 4.11: (a) Reconstruction using LDP [16, 64] with wavelets and TV (some reconstruction errors marked with arrows), (b) Reconstruction using DLMRI, (c) Magnitude of the reconstruction error for (a), (d) Magnitude of the reconstruction error for (b).
Figure 4.13: Undersampling limit with 2D random sampling and noisy data: (a) PSNR vs. undersampling factor for DLMRI and LDP [16, 64], (b) HFEN vs. undersampling factor for DLMRI and LDP [16, 64].
Figure 4.14: Sample reconstructions for the experiment of Figure 4.13: (a) Reconstruction using LDP [16, 64] with wavelets and TV at 2.5-fold undersampling, (b) Magnitude of reconstruction error for (a), (c) Reconstruction using DLMRI at 10-fold undersampling, (d) Magnitude of reconstruction error for (c).
Figure 4.15: Sample reconstructions for the experiment of Figure 4.13: (a) Reconstruction using DLMRI at 6-fold undersampling, (b) Magnitude of reconstruction error for (a), (c) Reconstruction using DLMRI at 4-fold undersampling, (d) Magnitude of reconstruction error for (c).
Figure 4.16: Parameter evaluation with 2D random sampling at 10-fold undersampling: (a) PSNR and HFEN vs. sparsity, (b) PSNR and HFEN vs. patch size, (c) PSNR and HFEN vs. overcompleteness of dictionary, (d) Noisy data, PSNR vs. $\lambda$ at $\sigma = 18.8$ (PSNR1), and $\sigma = 0.01$ (PSNR2).
5.1 Introduction

In this chapter, we study the use of multiscale dictionaries for DLMRI. The multiscale (or multiple scale) sparsifying dictionary enforces sparsity of the reconstructed image simultaneously at several scales or patch sizes. The problem formulation for this case involves simultaneously learning a multiscale dictionary and reconstructing the MR image from highly undersampled k-space data. We show that such an adaptive multiscale dictionary can provide better reconstruction quality and lower computational complexity than a dictionary learned at only one scale. This makes the use of multiscale dictionaries a very attractive prospect for MR image reconstruction. Such dictionaries can be learned directly using the undersampled k-space data without utilizing any reference images (just as in previous chapters for the single scale/single patch size based dictionary).

The advantage of the proposed multiscale MR image reconstruction formulation can be seen from the example presented in Figure 5.1. The sampling scheme shown employs variable density random undersampling in k-space by a factor of 7.11. The CS data acquisition was simulated by subsampling the 2D discrete Fourier transform of the reference image of the brain. The reconstructions with DLMRI (with single patch size) at two different scales/patch sizes of $5 \times 5$ and $6 \times 6$ have PSNRs (peak signal to noise ratios) of 35.35 dB and 35.47 dB respectively. The reconstruction with the proposed multiscale adaptive dictionary utilizing the scales/patch sizes of $3 \times 3$, $4 \times 4$, and $5 \times 5$ is shown in Figure 5.2 and has a superior PSNR of 36.67 dB. The magnitude of the image reconstruction error was thresholded at 4.8% of the maximum reference image intensity for the three cases yielding the regions of high reconstruction errors. The result with the multiscale dictionary in
Figure 5.2 can be seen to have fewer pixels of high error (about half the number) compared to the two single scale results in Figure 5.1.

Figure 5.2 shows the pixel-level amount of improvement in reconstruction error for the multiscale dictionary compared to the single scale ones (computed as $|x_s - x_{ref}| - |x_m - x_{ref}|$, where $x_{ref}$ is the reference image, $x_m$ is the multiscale reconstruction, and $x_s$ is the single scale reconstruction). The yellowish regions observed in these results indicate that the multiscale representation obtains much lower errors for those regions.

The reconstruction obtained with the CSMRI method of Lustig et al. [16] that employs non-adaptive dictionaries is shown in Figure 5.3. Wavelets and total variation were used as sparsifying transforms and the reconstruction can be clearly seen to have many aliasing artifacts. For this result, when the magnitude of the reconstruction error was thresholded at 4.8% of the maximum reference image intensity, nearly 51% of the image pixels fell above the threshold. This can also be inferred from the fact that the PSNR of this reconstruction was only 25.64 dB. Thus, owing to the relatively poor performance of non-adaptive CSMRI methods, we focus on only adaptive strategies in our comparisons for multiscale dictionaries.

It was shown in Chapter 4 that changing the parameters of the single scale DLMRI algorithm (such as increasing patch size) typically leads to only small fractions of a dB improvements in reconstruction PSNR. On the other hand, a multiscale adaptive reconstruction formulation that learns and combines the reconstructions at various scales can potentially lead to much higher improvements in reconstruction quality. This can be seen in the example in Figure 5.2.

The alternating algorithm that we proposed for single patch-size based dictionaries in Chapter 3 resulted in substantial improvements compared to CSMRI [16] even at high undersampling factors and utilizing no reference images. However, the dictionary in (3.1) involves only one scale and does not conform to any specific structure. Our multiscale framework exploits the fact that image patches have sparse representations at several scales. It obtains sparse approximations of patches at several scales and combines them to obtain a reconstruction. Our iterative algorithm for MR image reconstruction alternates between learning the multiscale dictionary and sparse representations at various scales and performing the image reconstruction. As a result, better multiscale dictionaries are learned at each iteration, leading to better
reconstruction in the next step.

Previous work [31] presented a semi-multiscale extension of the K-SVD algorithm. The authors use a quadtree decomposition of the learned dictionary and overlapping image patches. The semi-multiscale dictionary structure is obtained by arranging several fixed-sized learned dictionaries of different scales over a dyadic grid. This approach allows the sparse representation of a lower scale patch to contribute to the sparse representation of the higher scale patch to which it belongs. The resulting dictionary structure is found to have substantially superior performance over the single-scale K-SVD dictionary in applications such as denoising and inpainting. Our formulation is fundamentally different from this previous work. Firstly, the sparse representations of the patches at different scales do not contribute to one another. Instead, sparse representations of patches of different sizes (scales) are obtained independently, and these representations are combined additively. We call this the additive multiscale approach. Secondly, we do not restrict the scales to conform with a quadtree (dyadic) structure, thereby allowing a much larger/richer set of scales to contribute to the reconstruction. Finally, our work appears to be the first to apply dictionary learning at multiple scales to the solution of inverse problems in imaging.

Our multiscale formulation is similar to previous work in CSMRI [50] where a combination of sparsifying transforms such as wavelets, contourlets, and TV was used to obtain improved (by 1.3 dB PSNR) reconstructions compared to wavelets + TV. However, as opposed to this previous work, we work with only adaptive dictionaries and combine the sparsifying dictionaries from several scales to obtain improved reconstructions compared to a single scale dictionary.

Other adaptive dictionaries might also be considered for MRI reconstruction. A candidate is the class of recently introduced parametric dictionaries where the learned dictionary is allowed to have a special structure, such as the quadtree structure described earlier [31]. One alternative idea is an Image-Signature Dictionary (ISD) [30], which is essentially an image in which each 2D patch serves as a representing atom. A periodic extension of the ISD image is assumed. Hence, every pixel in the ISD serves as the corner of a patch and there are as many atoms as pixels in the ISD. The ISD is shown to perform reasonably well in image denoising compared to (older dictionary learning) methods such as MOD (method of optimal directions).
other parametric dictionary is a double sparse dictionary \cite{65} that combines the advantages of trained and analytic dictionaries. It is computationally cheaper than the standard unconstrained K-SVD dictionary \cite{25}. However, it is shown to perform only slightly better (up to a very small fraction of a dB in PSNR) than the latter while denoising CT data. Most recently, Ramirez and Sapiro \cite{66} have drawn on a codelength minimization interpretation of sparse coding, and used tools from universal coding theory for designing sparsity regularization terms in dictionary learning. They also try to minimize the correlation (or coherence) between the dictionary atoms in the framework. However, the improvement obtained over K-SVD \cite{25} in applications such as denoising is typically only 0.1-0.2 dB in PSNR.

A number of other parametric dictionaries have been proposed as well \cite{56}. Thus, one could enforce various interesting properties on the dictionary. However, the scope of the exploration of alternative dictionaries for MRI reconstruction can be narrowed by noting the superior performance of a multiscale quadtree representation \cite{31} compared to other dictionaries like ISD, double sparse dictionaries, and incoherent dictionaries.

As described earlier, in this work, we focus on the proposed additive multiscale approach. Work on a truly multiscale version of our framework (that generalizes \cite{31}) is ongoing and will be presented in the future. Finally, while we focus on MRI here, the iterative multiscale framework can be easily extended to general inverse problems in signal processing and imaging.

5.2 Problem Formulation

The problem formulation for CSMRI based on multiscale dictionary learning needs to have two features. It should be able to enforce sparsity of the patches of the reconstructed image at multiple scales, and also produce a reconstruction that is consistent with the available k-space data. It should also be able to combine the multiple scales to produce higher artifact reduction compared to a single scale dictionary.
An appropriate problem formulation is

\[
(P_{0m}) \quad \min_{x, D, A} \sum_{s=1}^{N_s} \sum_{ij} \gamma_s \left\| R_s^{ij} x - D_s^{ij} \alpha_s^{ij} \right\|_2^2 + \nu \left\| F_u x - y \right\|_2^2
\]

s.t. \( \| \alpha_s^{ij} \|_0 \leq T_0^s \ \forall \ i, j, s \)  \hspace{1cm} (5.1)

The superscript/subscript \( s \) is used to denote the scale. A total number of \( N_s \) scales/patch sizes are assumed. Patches at scale \( s \), denoted as \( x_{ij}^s \), are of size \( \sqrt{n_s} \times \sqrt{n_s} \). Matrix \( R_s^{ij} \in \mathbb{C}^{n_s \times P} \) represents the operator that extracts the patch \( x_{ij}^s \) from \( x \) as \( x_{ij}^s = R_s^{ij} x \). \( A \) here is used to denote the set \( \{ \alpha_{ij}^s \}_{ij,s} \) of sparse representations of all patches at all scales. Matrix \( D_s \in \mathbb{C}^{n_s \times K_s} \) denotes the sparsifying dictionary at scale \( s \) that contains \( K_s \) atoms. Let \( D \) denote the multiscale dictionary, i.e. \( D = \{ D_s \}_{s=1}^{N_s} \). The first term in the cost function captures the quality of the sparse approximations of the image patches at different scales with respect to the scale dictionaries \( D_s \). \( \gamma_s \) denotes the weighting for the quality of sparse approximations at scale \( s \). The second term in the cost enforces data fidelity in k-space. The patch sparsity constraint is the same as in single scale dictionary learning (3.1) except that a different sparsity level \( (T_0^s) \) is now enforced at each scale. The weight \( \nu \) is the same as in (3.1) and is used to make the formulation robust to noise.

We use periodically positioned, overlapping patches in the formulation. The overlap stride \( r_s \) at scale \( s \) is defined to be the distance in pixels between corresponding pixel locations in adjacent image patches at that scale. In this work, we assume the overlap stride to be the same at all scales \( (r_s = r, \ \forall \ s) \). The patches at a particular scale are said to have maximum overlap for \( r = 1 \). When there is patch wraparound and \( r = 1 \), each pixel in the image belongs to \( \sum_{s=1}^{N_s} n_s \) different patches where \( n_s \) patches contribute at scale \( s \).

Problem (P0m) of simultaneous reconstruction and multiscale dictionary learning is NP-hard, and non-convex even when the \( l_0 \) quasi norm is relaxed to an \( l_1 \) norm.
5.3 Algorithm

Problem (P0m) is solved using an iterative alternating minimization procedure similar to the one in Chapter 3. In one step of the alternating scheme, \( x \) is assumed fixed, and the multiscale dictionary is jointly learned with the sparse representations of the patches at various scales. In the other step, the dictionary and sparse representations are fixed, and \( x \) is updated to satisfy data consistency. These two steps are briefly detailed as follows.

5.3.1 Dictionary Learning Step

In this step, Problem (P0m) is solved with fixed reconstruction, \( x \). This corresponds to the Problem

\[
\begin{align*}
(P1_m) \quad & \min_{D, \mathcal{A}} \sum_{s=1}^{N_s} \sum_{ij} \gamma_s \| R_{ij}^s x - D^s \alpha_{ij}^s \|_2^2 \\
& \text{s.t.} \quad \| d_k^s \|_2 = 1 \quad \forall \, k, s, \quad \| \alpha_{ij}^s \|_0 \leq T_0^s \quad \forall \, i, j, s
\end{align*}
\]  

(5.2)

The cost function for dictionary learning accounts for the fitting errors of the overlapping patches at various scales using the multiscale sparsifying dictionary \( D \). The columns of the designed dictionary at each scale \( s \) (represented by \( d_k^s, 1 \leq k \leq K_s \)) are constrained to be of unit norm to avoid the scaling ambiguity [62]. The multiscale dictionary learning problem can be separated into sub-problems of the form

\[
\begin{align*}
(P1_s) \quad & \min_{D^s, \mathcal{A}^s} \sum_{ij} \| R_{ij}^s x - D^s \alpha_{ij}^s \|_2^2 \\
& \text{s.t.} \quad \| d_k^s \|_2 = 1 \quad \forall \, k, s, \quad \| \alpha_{ij}^s \|_0 \leq T_0^s \quad \forall \, i, j
\end{align*}
\]  

(5.3)

\( \mathcal{A}^s \) is used to denote the set \( \{ \alpha_{ij}^s \} \) of sparse representations of all patches at scale \( s \). The sub-problems (P1\(^s\)) can be solved independently at each scale, in combination providing a solution for Problem (P1\(^m\)). We use only a fraction, \( \delta \), of all patches to train the dictionary at each scale. The K-SVD algorithm [25] is used to learn the complex dictionary \( D^s \) at each scale. Once the dictionary \( D^s \) is learned, sparse coding is performed on all the patches at that scale to determine \( \mathcal{A}^s \).
5.3.2 Updating the Reconstruction

In this step, Problem (P0m) is solved with a fixed multiscale dictionary and sparse representations. The corresponding update problem is

\[
(P2m) \quad \min_x \sum_{s=1}^{N_s} \sum_{ij} \gamma^s \left\| R_{ij}^s x - D^s \alpha_{ij}^s \right\|^2_2 + \nu \left\| F_u x - y \right\|^2_2
\]  

(5.4)

Problem (P2m) is a least squares problem whose solution can be obtained by a procedure similar to the one adopted for the single scale dictionary case (Chapter 3).

The normal equation for this case is

\[
\left( \sum_{s=1}^{N_s} \sum_{ij} \gamma^s R_{ij}^s T R_{ij}^s + \nu F_u^H F_u \right) x = \sum_{s=1}^{N_s} \sum_{ij} \gamma^s R_{ij}^s T D^s \alpha_{ij}^s + \nu F_u^H y
\]  

(5.5)

The matrix \( \sum_{s=1}^{N_s} \sum_{ij} \gamma^s R_{ij}^s T R_{ij}^s \in \mathbb{C}^{P \times P} \) is diagonal with the entries corresponding to image pixel locations and their values are equal to the number of overlapping patches contributing at those pixel locations (when \( \gamma^s = 1, \forall s \)) from all scales. \( \sum_{s=1}^{N_s} \sum_{ij} \gamma^s R_{ij}^s T R_{ij}^s = \beta I_P \) when the patches wrap around at image boundaries at all scales. In particular, \( \beta = \sum_{s=1}^{N_s} \gamma^s n_s \) when the overlap stride \( r = 1 \) for the patches. The term \( \sum_{s=1}^{N_s} \sum_{ij} \gamma^s R_{ij}^s T D^s \alpha_{ij}^s \) with the additional scaling of \( \frac{1}{\beta} \) represents the multiscale patch-averaged result. The patches, having been approximated by the learned dictionary at different scales, are averaged at their respective locations in the image. The intensity value at each pixel is obtained by averaging (simple averaging when \( \gamma^s = 1, \forall s \), weighted averaging otherwise) the contributions of the various patches that cover it at different scales.

The solution to (5.5) is now the same as for the single scale case. The various terms are transformed from the image space to Fourier space with \( F x \) representing the full k-space data. The multiscale patch averaged result is transformed to the Fourier domain, producing

\[
S_m = \frac{F \sum_{s=1}^{N_s} \sum_{ij} \gamma^s R_{ij}^s T D^s \alpha_{ij}^s}{\beta}
\]
The solution to (5.5) is then

\[ F_x(k_x, k_y) = \begin{cases} 
  \frac{S_m(k_x, k_y)}{S_m(k_x, k_y) + \nu S_0(k_x, k_y)}, & (k_x, k_y) \notin \Omega \\
  S_0(k_x, k_y), & (k_x, k_y) \in \Omega 
\end{cases} \tag{5.6} \]

Equation (5.6) uses the multiscale dictionary-based interpolated values for the non-sampled Fourier frequencies, and fills back the sampled frequencies albeit with averaging in the presence of noise. The reconstruction, \( x \), is the IFFT of \( F_x \). The algorithm is summarized in Figure 5.4.

5.4 Algorithm Properties

5.4.1 Convergence

The multiscale algorithm alternates between learning the multiscale dictionary and sparse representations, and estimating the reconstruction. The solution to Problem (P1m) is obtained by solving \( N_s \) sub-problems (P1*). Solving Problems (P1m) and (P2m) iteratively leads to monotonic decrease in the cost function of (P0m). Since the cost function is non-negative, it converges. Thus, beginning with the zero-filled result, better multiscale sparse reconstructions (in the sense of (P0m)) are learned at each iteration. Empirically, the iterates \( x^k \) (indexed by iteration number) converge as well. Moreover, they converge faster than the iterates for the single scale dictionary case. This is shown in Section 5.5.

5.4.2 Multiscale Parameters

A study of algorithmic parameters was conducted in Chapter 4 for the single scale dictionary based algorithm. Parameters in the single scale dictionary framework such as the size of patches, sparsity threshold for each patch, the number of atoms of the dictionary, fraction of patches used for training, patch overlap stride, and the factor \( \lambda \) in the data consistency weight were studied. The multiscale algorithm has a few additional parameters such as the number of scales \( N_s \), scale weights \( \gamma^s \), patch sizes at those scales \( (\sqrt{n_s} \times \sqrt{n_s}) \), sparsity thresholds \( T^s_0 \), and the number of atoms of the dictionary at each
scale \((K_s)\). We study the effects of tuning some of these parameters in the next section.

5.4.3 Computational Complexity

The multiscale algorithm alternates between the image domain and k-space. The solution to Problem (P1m) involves solving \(N_s\) sub-problems (P1s) each corresponding to a particular scale. The sub-problem at scale \(s\) (P1s) involves learning the dictionary \(D^s\) from a fraction \(\delta\) of all \(M_s\) patches (total number of patches at that scale) and using it to obtain sparse approximations of the \(M_s\) patches. The dictionary \(D^s\) is learned using the K-SVD algorithm. The computation for learning is dominated by sparse coding which scales as \(O(\delta M_s K_s n_s T_0^s J)\) (\(J\) - number of K-SVD iterations). The cost of sparse coding all \(M_s\) patches using the learned dictionary is \(O(K_s n_s T_0^s M_s)\). The costs of the learning and the final sparse coding step at scale \(s\) are balanced by choosing \(\delta J = 1\). The total cost involved in solving problem (P1m) is then the sum of the complexities of the sub-problems at the various scales, \(O\left(\sum_{s=1}^{N_s} K_s n_s T_0^s M_s\right)\). The computational complexity of the reconstruction update step (P2) is dominated by 2 FFTs at cost \(O(P \log P)\). The other computations for (P2), namely patch averaging across scales and sampled frequency averaging, are fast, low complexity operations.

Under the assumptions of patch wraparound and overlap stride \(r = 1\), the total number of patches at each scale \(M_s = P\), and \(O\left(P \sum_{s=1}^{N_s} K_s n_s T_0^s\right) \gg O(P \log P)\) typically. This indicates that Problem (P1m) dominates the computational cost with the main speed bottleneck being the various sparse coding steps within (P1m). Suggestions similar to the ones described in Chapter 3 can be used to obtain speedups. Since the sub-problems (P1s) in the multiscale framework are independent, they can be solved in parallel. It will be shown in the next section that despite the complexity involved in the algorithm, the multiscale structure for the dictionary can provide better reconstructions at a lower complexity compared to single scale dictionaries.
5.5 Multiscale Experiments

5.5.1 Framework

In this section, the performance of the proposed additive multiscale image reconstruction algorithm is demonstrated at a variety of undersampling factors. The images used in the experiments are in vivo MR scans of size $512 \times 512$ (courtesy: [24]). Sampling schemes used in the experiments include 2D random sampling [19], and pseudo radial sampling [60]. The CS data acquisition was simulated by subsampling the 2D DFT of the MR images. The multiscale reconstruction method is compared with the reconstruction strategy in Chapter 3 involving single scale dictionaries. All implementations were coded in Matlab v7.8 (R2009a). Computations were performed with an Intel Core i5 CPU at 2.27 GHz and 4 GB memory, employing a 64-bit Windows 7 operating system.

In the experiments, the nominal values of the various parameters were set as $N_s = 3, \gamma^s = 1 \forall s, n_1 = 9, n_2 = 16, n_3 = 25, K_s = n_s, T^s_0 = 0.15 \times n_s, \lambda = 140$. We worked with maximum patch overlap, $r = 1$. The learning stage (K-SVD) employed 10 iterations, $200 \cdot K_s$ patches, and a sparsity of $T^s_0$ at each scale. The K-SVD algorithm was initialized with the left singular vectors of the training data at each scale. After learning, the overlapping patches were sparse coded using the sparsity threshold of $T^s_0$ at scale $s$. The algorithm was run for about 10-70 iterations in the experiments where the higher iteration count was needed for cases with worse initializations (zero-filled reconstruction) for $x$.

The parameters for the single scale case used in the comparisons were set as $K = n, T_0 = 0.15 \times n, \lambda = 140$, where $n$ values of 25 and 36 were used. The other parameters, such as learning iterations, were set similarly as in the multiscale case. The quality of the image reconstruction is quantified using PSNR and HFEN. Only complex-valued dictionaries are used in this section.

5.5.2 Results

We present several examples to demonstrate the promise of the multiscale framework for MRI. Figure 5.5 shows the performance of the algorithm on the reference image of Figure 5.1 employing pseudo radial sampling of k-
space at 7.19-fold undersampling. The sampling mask is shown along with the zero-filled Fourier reconstruction which is used as the initialization for the algorithm. The zero-filled Fourier reconstruction has significant undesirable artifacts due to aliasing and has a PSNR of 27.33 dB. The results with the multiscale dictionary and the single scale dictionary \((n = 36)\) both display reduction in aliasing compared to the initialization. Similar to Figure 5.1, the improvement in error for multiscale reconstruction compared to the single scale one was computed as \(|x_s - x_{ref}| - |x_m - x_{ref}|\). The brighter regions in yellow indicate the superior performance of the multiscale formulation at the pixel level (especially at edges). Aliasing removal is also better for the multiscale approach. The variation in PSNR over the iterations of the algorithm shows the multiscale algorithm overtaking the single scale one within a couple of iterations. The final PSNR of the single scale result was 32.92 dB while that of the multiscale reconstruction was 34.23 dB. Thus, an improvement of 1.31 dB in PSNR is achieved with the diversity of scales provided by the additive multiscale formulation.

Figure 5.6 shows the zero-filled reconstruction used as initialization for \(x\) in Figure 5.1. The aliasing artifacts are “incoherent” here due to the variable density random sampling employed. The variation of PSNR shows the multiscale dictionary converging to a much higher PSNR than the single scale ones \((n = 25\) and \(n = 36\)) although more slowly than in Figure 5.5.

Figures 5.7 and 5.8 demonstrate the performance of our algorithm on the reference image of Figure 5.1 using pseudo radial sampling at 7.19-fold undersampling. Zero-mean complex white Gaussian noise of standard deviation \(\sigma = 8\) was added in k-space. The image reconstructed from the noisy fully sampled k-space data is shown in Figure 5.7 along with the magnitude of the reconstructed image from fully sampled noise only, to help assess visually the noise level. The reconstructions are shown in Figure 5.8. The multiscale formulation again performs better than the single scale formulation \((n = 36)\). The improvement in error for the multiscale formulation compared to the single scale one is computed as \(|x_s - x_{ref}| - |x_m - x_{ref}|\). Yellow regions in the result, especially along edges, show the superior performance of the multiscale formulation in reconstructing those regions. The norm of the reconstruction difference between successive iterations \(||x^k - x^{k-1}||_2\) is plotted over iterations for both methods and is seen to converge quickly. It also decreases to a smaller value for the multiscale algorithm compared to
the single scale one. This shows the better convergence of the iterates $x^k$ for the multiscale algorithm.

PSNR and HFEN are plotted over iterations in Figure 5.8 and the multiscale algorithm converges to better values for these quantities. The initial zero-filled Fourier reconstruction had a PSNR of 27.27 dB. While the single scale algorithm had a final reconstruction PSNR of 32.54 dB, the multiscale formulation was better at 33.40 dB. The smaller final value of HFEN for the multiscale algorithm indicates better reconstruction of edges and fine features.

We evaluate some of the algorithmic parameters working with the example (sampled noisy data) of Figure 5.7. Figure 5.9 shows the results of the evaluation of two parameters, $\gamma^s$ and $N_s$. In one case, we performed the multiscale reconstruction varying only the $\gamma^s$. The $\gamma^s$ at scale $s$ was set proportional to the patch size at that scale. This is interesting since we typically get better “single scale” reconstructions at higher patch sizes. However, the PSNR with such a setting for the $\gamma^s$ yielded a reconstruction PSNR of 33.17 dB which is lower than the 33.4 dB obtained with $\gamma^s = 1 \forall s$ in Figure 5.7, suggesting that a uniform weighting is slightly better here.

In the second case in Figure 5.9, we varied the number of scales $N_s$ keeping all the other parameters at the same values as in Figure 5.7. Using only two scales ($n_1 = 25, n_2 = 16$) produced a reconstruction PSNR of 33.23 dB, which is lower than the 33.4 dB obtained with 3 scales. The value with the two scales is also close to the one obtained in the first case with skewed values for $\gamma^s$ indicating that setting $\gamma^s$ such that the highest scale gets the most weighting works more like a two-scale scenario. The PSNR using two scales is quite a bit higher than the PSNR with one scale ($n = 25$ which is close to $n = 36$ in performance), and adding the third scale of $n_3 = 9$ to the two-scale dictionary produces a smaller improvement (0.17 dB). Adding more scales such as $n = 36$ to the three-scale dictionary produced only small improvements in the result, whereas the computational cost became prohibitive with too many scales.

Figure 5.10 demonstrates the performance of our algorithm on an image (600 × 600 pixels) of the brain (courtesy of [67]) using variable density 2D random sampling at 10-fold undersampling. The multiscale formulation is seen to perform better than the single scale formulation ($n = 36$) for this case. The improvement in reconstruction error for the multiscale formula-
tion compared to the single scale one is computed as $|x_s - x_{ref}| - |x_m - x_{ref}|$. The bright regions of this result indicate the superior performance of the multiscale formulation in reconstructing those regions (especially the finer features). PSNR and HFEN are plotted over iterations in Figures 5.10 and 5.11 respectively. The initial zero-filled Fourier reconstruction had a PSNR of 25.36 dB. While the single scale algorithm had a final reconstruction PSNR of 35.75 dB, the multiscale formulation was better at 36.61 dB. The smaller final value of HFEN for the multiscale algorithm again indicates better reconstruction of edges and fine features. The norm of the reconstruction difference between successive iterations ($||x^k - x^{k-1}||_2$) is plotted over iterations for both methods in Figure 5.11 and can be seen to converge to a smaller value (also faster) for the multiscale algorithm compared to the single scale one. This demonstrates the better convergence properties of the multiscale algorithm.

Figure 5.12 demonstrates the performance of our algorithm on the reference image of Figure 5.1 using variable density 2D random sampling at 10-fold undersampling. The multiscale reconstruction algorithm provides better reconstructions than the single scale ($n = 36$) case. The improvement in reconstruction error for the multiscale formulation compared to the single scale one is computed as $|x_s - x_{ref}| - |x_m - x_{ref}|$. The bright regions of this result indicate the superior performance of the multiscale formulation in reconstructing those regions. PSNR is plotted over iterations. The initial zero-filled Fourier reconstruction had a PSNR of 24.76 dB. While the single scale algorithm had a final reconstruction PSNR of 35.19 dB, the multiscale formulation was better at 35.9 dB. The CSMRI result of [16] (not shown here) had a PSNR of 26.74 dB for this case. Thus, it can be seen that dictionary learning leads to promising improvements over non-adaptive dictionaries and multiscale dictionaries provide even more promising reconstruction quality compared to single scale ones.

5.5.3 Comparison of Computational Complexities

When the algorithmic parameters such as $K_s$ and $T_0^s$ are proportional to $n_s$ at each scale (i.e. $K_s = \tau_1 n_s$, $T_0^s = \tau_2 n_s$ with proportionality factors $\tau_1$ and $\tau_2$ being invariant to scale), the complexity of the multiscale algorithm
is approximately proportional to $\sum_{s=1}^{N_{\mathbf{s}}} n_s^3$. For the single scale dictionary case, this factor would simply amount to $n^3$ (under the same proportionality assumptions). When the patch sizes used for the multiscale algorithm, i.e. $n_1 = 9, n_2 = 16, n_3 = 25$, along with the single scale patch size of $n = 36$, are substituted into the complexity factors, it can be observed that the cost for the single scale case scales as $9^3 + 16^3 + 25^3$ over that of the multiscale framework. This amounts to a reduction in cost by a factor of 2.3 for the multiscale case compared to the single scale case ($n = 36$). Thus, the multiscale dictionary not only gives better reconstructions but also does that with lower computation. When, a single scale patch size of $n = 25$ is used, the complexity for the multiscale case is worse by a factor of 1.3. However, the single scale dictionary with $n = 25$ typically performs worse than the case involving $n = 36$.

5.6 A Sequential Multiscale Algorithm

In the experiments reported so far in this chapter, we have used the zero-filling solution to initialize the multiscale algorithm. While such an initialization is shown to work well, one could also start off with a better initialization that leads to faster convergence to the optimal solution. Here, we present one such alternative methodology which we call the sequential multiscale algorithm.

In this alternative version, the first few iterations of the alternating algorithm are run utilizing only the smallest scale ($n_1 = 9$) after which the next scale ($n_2 = 16$) is added. The highest scale ($n_3 = 25$) is added only in the last few iterations. Thus, the algorithm initially works like a single scale algorithm before switching to a double scale version and then finally a triple scale version. If there were more scales, these would also be added sequentially. This implies that the full multiscale algorithm utilizing all the scales would be initialized with a reasonably good solution obtained from previous iterations with smaller number of scales. Thus, convergence is expected to be quicker with the complete set of scales.

Figure 5.13 demonstrates the performance of the sequential multiscale algorithm on the reference image of Figure 5.1 using variable density 2D random sampling at 7.14-fold undersampling. The alternating reconstruction
algorithm was run for 65 iterations, of which the first 19 were run with only the smallest scale. Iterations 20 to 39 were run with two scales and the full multiscale version was used only from the 40th iteration. The reconstruction of this approach is shown along with the case when the full multiscale algorithm is run for all the 65 iterations. The plot of PSNR over iterations shows the sequential multiscale scheme initially lagging behind the full multiscale version but catching up by the 50th iteration. The final PSNR for the sequential multiscale scheme was slightly higher (37.67 dB) than that for the full multiscale version (37.53 dB). Thus, the sequential multiscale version converges to a better reconstruction and also does so faster since it utilizes only the smaller scales for about two thirds of the iterations, thereby requiring fewer computations than the full multiscale version for those iterations. Thus, a better initialization is seen to lead to both better reconstruction and faster convergence. The zero-filled reconstruction had a PSNR of only 25.26 dB for this case.
5.7 Figures

Figure 5.1: A comparison of multiscale dictionaries versus single scale dictionaries for DLMRI: (a) Axial T2-weighted reference image of the brain, (b) Sampling mask in k-space with 7-fold undersampling, (c) DLMRI reconstruction employing a patch size of $5 \times 5$, (d) Regions of high reconstruction error ($>4.8\%$) for (c) shown in white, (e) DLMRI reconstruction employing a patch size of $6 \times 6$, (f) Regions of high reconstruction error ($>4.8\%$) for (e) shown in white.
Figure 5.2: A comparison of multiscale dictionaries versus single scale dictionaries for DLMRI using the k-space data of Figure 5.1: (a) Multiscale DLMRI reconstruction employing patch sizes of $3 \times 3$, $4 \times 4$, and $5 \times 5$, (b) Regions of high reconstruction error for (a), (c) Improvement in reconstruction error for (a) compared to Figure 5.1 (c) (positive values (bright pixels) denote reduced error, red denote no change, and negative values (dark pixels) denote increased error), (d) Improvement in reconstruction error for (a) compared to Figure 5.1 (e).
Figure 5.3: CSMRI reconstruction with non-adaptive dictionary using the k-space data of Figure 5.1: (a) CSMRI result of [16] with wavelets and total variation, (b) Regions of high reconstruction error (> 4.8%) for (a) shown in white.
Algorithm 2

**Input**: $y$ - k-space measurements  

**Output**: $x$ - Reconstructed MR image  

**Initialization**: $x = x_0 = F_u^H y$

**Iteration**:

1. Learn dictionaries, sparse representations for patches of $x$ at the multiple scales
2. Update $x$: Each pixel value obtained by averaging contributions of all patches that cover it at various scales
3. $S \leftarrow FFT(x)$
4. Restore Sampled frequencies to update $S$
5. $x \leftarrow IFFT(S)$

Figure 5.4: Algorithm to reconstruct MR images from undersampled k-space measurements using multiscale adaptive dictionaries.
Figure 5.5: Radial sampling: (a) Sampling mask in k-space with 7-fold undersampling, (b) Zero-filled reconstruction, (c) DLMRI reconstruction employing a single patch size of $6 \times 6$, (d) Multiscale DLMRI reconstruction employing patch sizes of $3 \times 3$, $4 \times 4$, and $5 \times 5$, (e) Improvement in reconstruction error for (d) compared to (c) (positive values (bright pixels) denote reduced error, red denote no change, and negative values (dark pixel) denote increased error), (f) PSNR vs. iterations for (c) and (d).
Figure 5.6: Initialization and PSNR variation for Figure 5.1: (a) Zero-filled reconstruction (b) PSNR vs. iterations for multiscale and single scale reconstructions.
Figure 5.7: Noisy data: (a) The noisy fully sampled image, (b) Noise in (a), (c) Sampling mask in k-space with 7-fold undersampling, (d) Zero-filled reconstruction.
Figure 5.8: Reconstructions for the noisy data of Figure 5.7: (a) DLMRI reconstruction employing a single patch size of 6 × 6, (b) Multiscale DLMRI reconstruction employing patch sizes of 3 × 3, 4 × 4, and 5 × 5, (c) Improvement in reconstruction error for (b) compared to (a), (d) Norm of the reconstruction difference between successive iterations, (e) PSNR vs. iterations for (a) and (b), (f) HFEN vs. iterations for (a) and (b).
Figure 5.9: Parameter evaluation for the data of Figure 5.7: (a) Multiscale DLMRI reconstruction employing patch sizes of $3 \times 3$, $4 \times 4$, and $5 \times 5$ and scale dependant $\gamma^s$, (b) Improvement in reconstruction error for (a) compared to the single scale reconstruction of Figure 5.7, (c) Multiscale DLMRI reconstruction employing 2 scales/patch sizes of $4 \times 4$, and $5 \times 5$ and uniform $\gamma^s$, (b) Improvement in reconstruction error for (c) compared to the single scale reconstruction of Figure 5.7.
Figure 5.10: Multiscale reconstruction with 2D random sampling: (a) Reference image of the brain, (b) Sampling mask in k-space with 10-fold undersampling, (c) DLMRI reconstruction employing a single patch size of $6 \times 6$, (d) Multiscale DLMRI reconstruction employing patch sizes of $3 \times 3$, $4 \times 4$, and $5 \times 5$, (e) Improvement in reconstruction error for (d) compared to (c), (f) PSNR vs. iterations for (c) and (d).
Figure 5.11: Multiscale reconstruction with 2D random sampling for the data of Figure 5.10: (a) HFEN vs. iterations for the single scale and multiscale algorithms, (b) Norm of the reconstruction difference between successive iterations for the single scale and multiscale algorithms.
Figure 5.12: 2D random sampling: (a) Sampling mask in k-space with 10-fold undersampling, (b) Zero-filled reconstruction, (c) DLMRI reconstruction employing a single patch size of 6 × 6, (d) Multiscale DLMRI reconstruction employing patch sizes of 3 × 3, 4 × 4, and 5 × 5, (e) Improvement in reconstruction error for (d) compared to (c), (f) PSNR vs. iterations for (c) and (d).
Figure 5.13: 2D random sampling at 7-fold undersampling: (a) Multiscale DLMRI reconstruction employing patch sizes of $3 \times 3$, $4 \times 4$, and $5 \times 5$, (b) Multiscale DLMRI reconstruction employing patch sizes of $3 \times 3$, $4 \times 4$, and $5 \times 5$ where the scales are added sequentially, (c) Sampling mask in k-space, (d) PSNR vs. iterations for (a) and (b).
CHAPTER 6
CONCLUSIONS

CSMRI methods employing analytical sparsifying transforms such as wavelets and finite differences can perform poorly at high undersampling factors. In this thesis, a novel adaptive reconstruction framework exploiting image patch-based sparsity has been presented and shown to be very beneficial for highly undersampled CSMRI. The patch-based dictionary is obtained directly using the undersampled k-space data and is thus adapted to the specific image instance. Learning directly from the CS measurements also removes the need for fully sampled reference images.

The dictionary learning itself can be done at either one scale or multiple scales. When only a single scale is used, the alternating reconstruction algorithm presented in this work learns the dictionary and removes aliasing and noise in the image domain in one step, and enforces data fidelity in k-space in the next step. Various experimental results demonstrate the superior performance of such an algorithm in both noiseless and noisy scenarios as compared to previous CSMRI methods. The performance is demonstrated using a variety of sampling trajectories and k-space undersampling factors. The algorithm usually converges in a small number of iterations and provides highly accurate reconstructions at high undersampling factors. It is also robust to parameter selection. The dictionary learning step of the algorithm can also be additionally initialized with a dictionary learned from reference image(s), in order to reduce the number of iterations required for convergence, thus accelerating the algorithm further.

The promise of (additive) multiscale adaptive dictionaries for MRI is also demonstrated in this work. The additive multiscale framework learns dictionaries at several scales and then combines the image patch approximations at those scales to enable superior reconstructions. The reconstruction algorithm for this formulation alternates between the image domain and k-space similar to the single scale dictionary case. Experimental results demonstrate the su-

72
prior performance of multiscale dictionaries compared to single scale ones. The former provide both better reconstructions and lower computational complexity. A sequential multiscale version of the reconstruction algorithm is shown to work even faster. As mentioned earlier in Chapter 5, work on a truly multiscale formulation is ongoing and will be presented in the future.

The design of optimal sampling schemes for dictionary learning based MRI is a subject for future study. The proposed adaptive framework for image reconstruction employing either single scale or multiscale dictionaries can also be extended to other imaging applications. Parallel imaging and dynamic imaging in MRI as well as other imaging techniques such as computed tomography (CT) may benefit from such an adaptive framework.
REFERENCES


