



Universidade de Brasília

**Instituto de Ciências Exatas
Departamento de Ciência da Computação**

The application of analysis filters in compressed sensing algorithms for magnetic resonance imaging reconstruction

Jonathan A. S. Lima

Tese apresentada como requisito parcial para
conclusão do Doutorado em Informática

Orientador

Prof.^a Dr.^a Mylène Christine Queiroz de Farias

Coorientador

Prof. Dr. Cristiano Jacques Miosso

Brasília
2019



Universidade de Brasília

**Instituto de Ciências Exatas
Departamento de Ciência da Computação**

The application of analysis filters in compressed sensing algorithms for magnetic resonance imaging reconstruction

Jonathan A. S. Lima

Tese apresentada como requisito parcial para
conclusão do Doutorado em Informática

Prof.^a Dr.^a Mylène Christine Queiroz de Farias (Orientador)
CIC/UnB

Prof. Dr. Bruno L. M. Espinoza Prof. Dr. Adson F. da Rocha
University of Brasília University of Brasília

Prof. Dr. Eduardo M. A. M. Mendes
University of Minas Gerais

Prof. Dr. Bruno L. M. Espinoza
Coordenador do Programa de Pós-graduação em Informática

Brasília, 09 de agosto de 2019

Agradecimentos

Agradeço a CAPES pelo apoio financeiro. Agradeço principalmente à Professora Mylène, que fez eu me apaixonar por processamento de sinais desde aquelas primeiras aulas de Processamento de Sinais Multimídia em 2011, e com isso influenciou todo o resto da minha trajetória acadêmica. Agradeço ao professor Cristiano, por sua coorientação, me ajudando nos problemas, nas decisões, me apresentando Compressed Sensing e biomédica, expandindo ainda mais meus conhecimentos. Ambos acreditaram em mim como ninguém, e me motivaram a chegar até aqui. E claro, as orientações, as reuniões, a compreensão, a paciência, (as idas ao El Negro). Agradeço também à tantos outros professores, Lineu, Célius, João Luiz, Assis, Márcio, Carla, Alba, Mia, Ayala, Vidal, Lucero, que me permitiram olhar mais longe. Aos membros da banca, pela avaliação crítica e justa da tese, que adicionou muito ao trabalho. Agradeço ao pessoal do mestrado, Amanda, Paula, Ariane, Gabi, Lucas, Helard, Jeremias, Gustavo, Nilson, Daniel. Dividir aulas, estudos e momentos com essas pessoas queridas me deixou mais à vontade para lidar com os desafios acadêmicos. Agradeço aos colegas de GPDS que dividi sala, Luciana, Wellington, Gustavo, Muhammad. As vezes eram os primeiros a ver um avanço, um gráfico recém saído do forno, inclusive com suas opiniões e sugestões. E também Dário, Henrique, Helard, Pedro, Sana, Felipe, pelo convívio no GPDS, pelas apresentações e discussões do grupo. Agradeço ao departamento de ciência da computação, pela ajuda com as burocracias. Em especial a Paula (que nem está mais lá), que foi comigo sei lá onde na reitoria e conseguiu sair de lá comigo matriculado no doutorado. E também à Carol, que resolveu os trâmites da tese. Agradeço ao Felipe e von Borries, pela ajuda no intercâmbio em El Paso, inclusive ajudando a me sentir mais a vontade lá. E pelas trocas de ideias, discussões, colaborações. Agradeço à minha família, pelo apoio, e a compreensão e paciência com o tempo que um doutorado leva. Agradeço Juli, por todo o incentivo durante esses anos, sempre motivando para eu conseguir concluir o trabalho, e as correções de texto. São todos uns fofos!

Resumo Expandido

Métodos de filtragem digital em compressed sensing para reconstrução de imagens de ressonância magnética

Introdução

Imageamento por ressonância magnética (RM) permite a visualização de órgão internos e o diagnóstico de patologias, tratamentos, acompanhamento de procedimentos médicos, planejamento de cirurgias, etc. Ao contrário de raio-X e tomografia computacional, RM não emite radiação ionizante e gera imagens com um ótimo contraste para tecidos moles [1]. Entretanto, um típico exame de RM pode demorar horas, o que aumenta o custo e diminui a abrangência de seu uso. Além disso, o exame requer que o paciente se mova o mínimo possível, muitas vezes tendo que prender a respiração por 15 segundos. Um exame de RM mais rápido pode diminuir esses custos, aumentar a abrangência do uso, aumentar o conforto do paciente e permitir novos tipos de exame e evitar artefatos de movimento nas imagens.

Uma das formas de diminuir o tempo do exame é reconstruir as imagens com um número menor de medidas. *Compressed Sensing* (CS) [2, 3] é um método de reconstrução de sinais subamostrados que sejam esparsos em algum domínio conhecido. Entre os métodos de CS, a minimização da variação total (TV) [4, 2] é um método que reconstrói imagens com gradiente esparso. Em imagens de RM, estruturas com mesma propriedade apresentam um nível de cinza muito próximo, o que torna o gradiente da imagem bastante esparso. Portanto, a minimização TV é popular entre os métodos de reconstrução de RM por CS [5, 6, 7, 8, 9, 10]. Portanto, melhorar a qualidade gerada pela minimização TV permite melhorar o resultado de muitas técnicas de reconstrução de RM. NESTA [11] é um algoritmo rápido e robusto para resolver problemas de reconstrução por CS. O método usa aproximações convexas e diferenciáveis das funções a se minimizar e obtém as soluções por um método iterativo de primeira ordem. Particularmente na solução para a TV, o método aplica o filtro de diferenças finitas verticais e horizontais na aproximação da função TV.

Outro método utilizado para o problema de reconstrução subamostrada de RM com CS é a pré-filtragem [12]. No método, primeiro é aplicado uma série de filtros diretamente nas medidas. O método então usa as versões filtradas das medidas para reconstrução por CS. As reconstruções obtidas correspondem as versões filtradas da imagem original. Por fim, o método recombina os espectros reconstruídos junto com as medidas originais para obter a reconstrução final. Os filtros testados foram de diferenças finitas na vertical, horizontal e diagonal, e o resultado obtido apresentou maior qualidade que a minimização TV.

Portanto, a literatura apresenta dois tipos de métodos de filtragem utilizada com CS, a filtragem durante a reconstrução e a pré-filtragem. Entretanto, apenas filtros de diferenças finitas foram testados sistematicamente nestes métodos. Há possibilidade de testar outros filtros e algoritmos nas reconstruções. A hipótese é que outros filtros podem esparsificar ainda mais as imagens, e isso proporcionaria uma melhor qualidade ao resultado ao utilizar esses métodos. O objetivo desta tese é propor, simular e analisar métodos de filtragem com CS para resolver o problema de reconstrução de RM subamostrado. Para cada método, o objetivo é projetar os conjuntos de banco de filtros e analisar a qualidade de reconstrução de cada. É desejado obter imagens com maior qualidade que os métodos que usam diferenças finitas (TV e pré-filtragem com filtros de diferenças finitas), ou uma qualidade semelhante a partir de menos medidas. Isso permitiria diminuir o tempo de aquisição de medidas nos exames de RM.

Metodologia

Bancos de filtros para a pré-filtragem

Primeiramente, foi feito o teste sistemático do método de pré-filtragem com um conjunto de banco de filtros projetados com características distintas. O primeiro conjunto de banco de filtros é baseado no projeto de filtros por janelamento [13]. Os bancos de filtros são projetados de forma que cada um deles ocupe uma faixa diferente do espectro de igual tamanho. Eles são aplicados na horizontal e vertical para formar filtros 2D, sendo que o filtro passa-baixas é descartado. São projetados 20 bancos de filtros, com as combinações de 2 a 5 divisões de banda e de ordem 2 a 10 (de 2 em 2). Estes bancos de filtros são designados neste trabalho como WIN(ordem, bandas).

O outro conjunto de bancos de filtros é baseado na decomposição wavelet. Como a decomposição wavelet é formada por uma sequência de filtragens e decimações, ela foi adaptada para conter uma sequência de filtragens, usando a propriedade de decimação da transformada Z. Cada nível de decomposição corresponde a uma posição diferente do espectro. Os filtros são aplicados horizontalmente e verticalmente para formar filtros 2D,

em que o filtro passa-baixas é descartado. São projetados 21 filtros, com combinações de 7 famílias wavelet e 3 níveis de decomposição. Neste trabalho, refiro a este banco de filtros como WAV.

O primeiro experimento usa o método de pré-filtragem com o algoritmo de reconstrução IRLS [14], em que as reconstruções são minimizações ℓ_1 . O processo de tomada de medidas pela máquina de RM é simulado a partir de imagens de cortes de cérebros. As medidas correspondem a 20, 40, 60, 80 e 100 linhas radiais no espaço k. A avaliação de qualidade é feita com as métricas SNR e SSIM. São calculadas ainda a ℓ_1 média das imagens filtradas, assim como a cobertura do espectro dos bancos de filtros.

Inclusão de medidas de valor zero na pré-filtragem

Para o segundo experimento, é proposta uma modificação no método de pré-filtragem. Para cada filtro, medidas com valor zero são incluídas nas regiões de banda de rejeição no espaço de soluções, por um método de limiar. Assim, a minimização força soluções com que tenham espectro igual a zero na banda de rejeição do filtro (como o esperado para a solução correta). É considerada como banda de rejeição as regiões do espectro em que a amplitude é menor que um limiar. As medidas de valor zero são associadas a essas regiões do espectro.

Para um limiar igual a 1% da amplitude máxima de um filtro, para o banco de filtros WAV(db4,1), e para 20 linhas radiais, o número de amostras passa de 10.4% para entre 34.4% e 44.4% ao incluir as medidas abaixo do limiar. Para o experimento, foram utilizadas as mesmas imagens, os mesmos bancos de filtros do experimento anterior. O algoritmo de reconstrução NESTA ao invés do IRLS. O limiar foi escolhido empiricamente, com valores 0%, 0.5%, 1% e 5% do valor máximo de amplitude do filtro.

Norma da filtragem

Foi proposta a norma da filtragem. O método consiste em minimizar a norma ℓ_1 ou ℓ_2 da filtragem do sinal a ser reconstruído. A filtragem é composta pela aplicação de um conjunto de filtros escolhidos pelo usuário. Os filtros podem deixar o sinal esparso, e assim favorecer a reconstrução por CS. O método proposto é equivalente à TV quando os filtros utilizados são os de diferenças finitas.

Três variações do método são propostas, a norma isotrópica da filtragem, a norma anisotrópica e a norma isotrópica + anisotrópica. A norma isotrópica é baseada na norma ℓ_2 , enquanto a anisotrópica se baseia na norma ℓ_1 . A implementação foi baseada no algoritmo NESTA, utilizando os filtros para obter a aproximações dos gradientes. Finalmente, utiliza-se o mesmo algoritmo de reconstrução utilizado no NESTA,

alterando a função gradiente. Para a isotrópica + anisotrópica, o gradiente é a soma dos gradientes da isotrópica e anisotrópica.

O terceiro experimento simula reconstrução de RM utilizando a norma da filtragem, em suas formas isotrópicas, anisotrópicas e isotrópicas + anisotrópicas. São utilizadas as mesmas imagens e padrões de amostragem dos experimentos anteriores. 8 tipos de combinações de filtros são testados, que incluem diferenças finitas de primeira e segunda ordem e filtros WIN. Foram incluídos 4 níveis de ruído gaussiano nas medidas: sem ruído, 60 dB, 40 dB e 20 dB.

Pré-filtragem com decimação

Foi proposta adição de um processo de decimação na reconstrução no método da pré-filtragem. Ao realizar uma operação de decimação de ordem 2, acontece *aliasing*, o que faz com que a parte do espectro acima de $\pi/2$ sejam rebatidos, e o espectro fica mais preenchido. Isso permite diminuir a esparsidade do espectro sem afetar a esparsidade do sinal do domínio do espaço. O método aplica essa operação no domínio da frequência, pois as medidas estão neste domínio. Entretanto, para cada medida válida para a reconstrução, é necessário que as medidas nas posições i e $i + N/2$ façam parte da amostragem original, caso contrário, as duas medidas devem ser descartadas. Isso limita a aplicação da técnica pois geralmente diminui bastante o número de medidas para a reconstrução. Esta parte é um trabalho em andamento, sem resultados por enquanto.

Resultados

Bancos de filtros para a pré-filtragem

São mostrados banco de filtros que apresentam qualidade de reconstrução maior que os filtros Haar. Os bancos de filtros WIN de menor ordem correspondem aos melhores resultados em qualidade de reconstrução. A maior melhora de qualidade é para 20 linhas radiais, que mostra uma melhora de 2.8 dB/0.106 SNR/SSIM O banco de filtros WIN de ordem 2 obtiveram entre 1.2 e 2.5 dB maior SNR que os resultados com TV NESTA, e entre entre 1.4 e 3.6 dB maior SNR que o pré-filtragem com filtros de diferenças finitas.

Foi tomada como hipótese que os filtros que proporcionam a maior esparsidade nas imagens obteriam as melhores qualidades de reconstrução. Porém, não foi encontrada uma correlação entre a qualidade da imagem e a métrica ℓ_1 . Talvez a métrica utilizada não tenha sido uma boa medida de esparsidade. Há uma correlação entre a cobertura do espectro e a qualidade apenas quando as medidas originais são incluídas no cálculo da cobertura do espectro.

O tempo de reconstrução é influenciado principalmente pelo número de filtros do banco de filtros usada na reconstrução. O banco de filtros WIN(2,2) obteve uma das melhores reconstruções e tem apenas 3 filtros, sendo o melhor compromisso entre qualidade e tempo de reconstrução.

Inclusão de medidas de valor zero na pré-filtragem

No método de pré-filtragem, o algoritmo NESTA não obteve melhor resultado de reconstrução que o IRLS para os filtros WIN, e similar qualidade para os filtros WAV. O método do limiar deixou a qualidade maior ou igual em relação ao método tradicional. As maiores melhorias em qualidade ocorreram com um limiar de 5%. Entretanto, a escolha foi empírica, e parece ser a melhor abordagem para o problema, pois o resultado parece ser dependente do banco de filtros utilizado e da imagem.

Norma da filtragem

O melhor resultado foi obtido ao utilizar a norma isotrópica + anisotrópica da filtragem, com uma combinação de 8 diferentes filtros incluindo de diferenças finitas de primeira, e segunda ordem e o banco de filtros WIN(2,2). Para essa configuração, o SNR da reconstrução foi显著mente maior que ao do TV NESTA, para todos os níveis de ruído. Um teste estatístico ANOVA foi aplicado para isolar a influência da função de norma e do conjunto de filtros, e o teste apontou que ambos a função de norma isotrópica + anisotrópica e os filtros apresentaram ganhos significativos em relação aos outras funções de norma e filtros, respectivamente.

Conclusão

Neste trabalho, técnicas de filtragem em CS métodos e bancos de filtros foram propostas e avaliadas para a reconstrução de imagens de RM a partir de medidas subamostradas. Houve melhora da qualidade de reconstrução em relação a minimização TV e ao método de pré-filtragem com filtros de diferenças finitas. Todos os experimentos apresentados mostram algum tipo de melhora nos resultados. A partir do primeiro experimento, se obteve a melhor qualidade de reconstrução para poucas linhas radiais. A partir do segundo experimento, se obteve um método que melhora quase sempre a qualidade de reconstrução para a pré-filtragem. E no terceiro experimento é proposta uma nova técnica, e ela obtém melhores resultados comparado às outras para uma maior amostragem. O trabalho apresentado nesta tese abre a oportunidade para que os métodos sejam aplicados nas máquinas de RM e possam melhorar a qualidade nos problemas reais de reconstrução por RM.

Palavras-chave: compressed sensing, Imageamento por ressonância magnética, Filtragem

Abstract

Magnetic Resonance Imaging (MRI) exams usually take a long time to be performed because they require a great amount of measurements to reconstruct an image with good quality. Decreasing the acquisition time of MRI can prevent motion artifacts, make possible to perform new types of exams, and also reduce MRI costs.

Compressed Sensing (CS) techniques are able to reconstruct MRI images at a sub-Nyquist rate, provided that the signals are sparse in a known domain. A CS method known as total variation (TV) minimization, minimizes the finite differences to reconstruct the signal. This operation can be interpreted as a filtering operation that is performed in the reconstruction steps. On the other hand, the pre-filtering method reconstructs filtered versions of the image with CS and recombine their spectrum to obtain a better image quality. This method relies on the fact that (high-pass) filtered versions of the images are sparse in the pixel domain and can be reconstructed with CS using fewer measurements.

In this work, I use filtering methods with CS to improve the quality of the undersampled MRI image reconstructions. The filters provide sparsity to the images, and generate better CS reconstructions. In the pre-filtering methods, I proposed a systematical test to evaluate a large number of filter banks, which were still not tested in the pre-filtering literature. I also proposed a threshold method to include measurements in the solution space, based on the stop-band of the filters. Finally, I proposed the filtering norms, a method that uses filters in the reconstruction algorithm. This method generalizes the TV minimization for any type of filter. I simulated the methods extensively for different sampling density and on a large set of images, and use objective metrics to evaluate the reconstruction quality. The pre-filtering, for low order filters designed with windowing method obtained SNR values between 1 and 2.9 dB higher than the TV minimization. Filtering norms with a combination of filters resulted in SNR values between 1.2 and 1.5 dB higher than values obtained with the TV. In most cases, the threshold method improved the image quality results. However, the highest quality improvements were observed for poor reconstructions.

Keywords: compressed sensing, Magnetic resonance imaging, filtering

Publications

1. Jonathan Lima, Cristiano Miosso, and Mylène Farias. Per-pixel mirror-based acquisition method for video compressive sensing. In European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, 2014. [15]
2. Paula Lima, Jonathan Lima, and Priscila Solis. Rea-wsn: Intercluster routing algorithm for energy optimization in wireless sensor networks. In 2015 7th IEEE Latin-American Conference on Communications (LATINCOM), pages 1–7. IEEE, 2015. [16]
3. Jonathan Lima, Cristiano Miosso, and Mylène Farias. Per-pixel mirror-based method for high-speed video acquisition. *Journal of Visual Communication and Image Representation*, 47:23–35, 2017. [17]
4. Jonathan Lima, Cristiano Miosso, and Mylène Farias. Avaliação de filtros de decomposição wavelet para reconstrução de imagens de ressonância magnética com base em compressive sensing com pré-filtragem. In V Congresso Brasileiro de Eletromiografia e Cinesiologia e X Simpósio de Engenharia Biomédica (COBEC-SEB 2017), 2017. [18]
5. Jonathan Lima, Cristiano Miosso, Mylene Farias, and Ricardo von Borries. Evaluation of different types of filters in magnetic resonance imaging using compressive sensing with pre-filtering. In Engineering in Medicine and Biology Society (EMBC), 2018 40th Annual International Conference of the IEEE. IEEE, 2018. [19]

Contents

1	Introduction	1
1.1	Context and problem definition	1
1.2	Objectives	4
1.3	Summary of the thesis contributions	4
1.4	Thesis organization	6
2	Theoretical foundations and state-of-the-art	8
2.1	Magnetic resonance imaging	8
2.1.1	MRI Basic concepts	8
2.1.2	Frequency encoding and trajectories	11
2.2	Compressed sensing	13
2.2.1	General domain of sparsity	15
2.2.2	Total Variation minimization	16
2.3	The pre-filtering method	17
2.3.1	Non-uniform Pre-filtering	20
2.4	The NESTA Algorithm	21
2.4.1	The Nesterov's method for function minimization	21
2.4.2	NESTA	22
2.4.3	NESTA for Total Variation minimization	24
3	Filter banks for pre-filtering	25
3.1	Filter banks	25
3.1.1	Filters banks based on windowing method (WIN)	26
3.1.2	Wavelet decomposition filters banks (WAV)	27
3.2	Experimental methodology	28
3.2.1	Simulations settings	28
3.2.2	Analysis Parameters	31
3.3	Results	33
3.4	Conclusions	39

4 Improving pre-filtering by adding zero-valued measurements	42
4.1 Proposed pre-filtering modifications	42
4.2 Experimental methodology	45
4.3 Results	46
4.4 Conclusions	53
5 Isotropic and anisotropic filtering norms	57
5.1 NESTA brief recap	57
5.2 Filtering Norm Minimization	59
5.2.1 NESTA Isotropic Filtering Norm	60
5.2.2 NESTA Anisotropic Filtering Norm	61
5.2.3 NESTA Isotropic + Anisotropic Filtering Norm	62
5.2.4 The matrix \mathbf{H}	62
5.3 Applications and Simulations	64
5.3.1 Tests Using Synthetic Images	64
5.3.2 Tests Using Magnetic Resonance Images	65
5.4 Conclusions	73
6 Towards a Pre-filtering with decimation	75
6.1 Pre-filtering with decimation	77
6.2 Mathematical formulation of the pre-filtering with decimation	78
6.3 Practical implications	82
6.4 Conclusions	84
7 Conclusions	86
7.1 Future works	88
Bibliography	90

List of Figures

2.1	MRI scanner, showing also the coils that generates the magnetic fields and scans the output signal. Image source: [20].	9
2.2	Rectangular (a), spiral (b) and radial (c) samplings. Magnitude of gradient field is shown in left, and the corresponding sampling trajectories scheme in right. Equal color points represent the same time.	12
2.3	High-pass filtered versions of the original image. First row shows the original image, second row shows the spectrum of the filters and third row shows the filtered versions of the original image, filtered by the filter shown in the second row. Note that each one of the filtered images are sparser in pixel domain than the original image.	18
2.4	Reconstruction using pre-filtering procedure using vertical and horizontal finite differences filters.	19
2.5	Reconstruction using pre-filtering procedure using Haar filters.	20
3.1	Reconstruction using pre-filtering procedure for any filter bank.	26
3.2	Example: The magnitude of a WIN filter bank formed by 8 different 2D filters, with different band divisions, with order 6.	27
3.3	Wavelet decomposition simplification to the Wavelet filter bank.	29
3.4	Example: Magnitude of 2 level Biorthogonal type 2.8 wavelet filters. Only the low-pass filter is not considered.	30
3.5	Sampling positions for the reconstruction simulations, ranging from 20 to 100 radial lines projections.	30
3.6	Some slices from the first participant brain, taken from an MRI machine.	30
3.7	Spectrum coverage for 3 different filter configurations. First column is WIN filter bank design with 12 coefficients and 2x2 band divisions, second is TV filter bank and third column is Daubechies's 3 wavelet filter bank with 2 levels. First row is using 20 radial lines, while second is using 80 lines.	33
3.8	Average SNR (dB) of the reconstruction using the pre-filtering with IRLS with p=1 for the filter banks tested. It is also shown SNR of reconstruction with the NESTA TV minimization as horizontal lines.	35

3.9	Average SSIM of the reconstruction using the pre-filtering with IRLS with p=1 for the filter banks tested. The y axis is in a log scale. It is also shown the SSIM of reconstruction with the NESTA TV minimization as horizontal lines.	35
3.10	MRI reconstructions with pre-filtering. Columns corresponds to respectively 20, 40, 60, 80 and 100 radial lines.	36
3.11	Average SNR(dB) vs average ℓ_1 of the filtered images. Each point corresponds to a filter bank.	38
3.12	Average SNR(dB) vs spectrum coverage of the filter banks. First plot does not include the measurements in the spectrum coverage. The second includes the measurements of part of covered spectrum.	40
3.13	Reconstruction times (s) grouped by number of radial lines and number of filters in the filter banks.	41
4.1	New representation of the pre-filtering method.	44
4.2	Modified pre-filtering method with extra zero-valued measurements.	45
4.3	Sampled positions. White lines represent the original sampling pattern. Translucid white areas represent the extra measurements, <i>i.e.</i> , the points for which the magnitude of the filter spectrum is smaller than 1% of maximum filter magnitude.	45
4.4	Average SNR(dB) of the reconstruction using the pre-filtering with NESTA for the filter banks tested. Average SNR values for reconstructions with the NESTA TV minimization are shown as horizontal lines.	49
4.5	Average SSIM of the reconstruction using the pre-filtering with NESTA for the filter banks tested. Average SSIM values for reconstructions with the NESTA TV minimization are shown as horizontal lines.	49
4.6	Average SNR(dB) of the reconstruction using the pre-filtering with NESTA for the filter banks tested. Average SNR values for reconstructions with the NESTA TV minimization are shown as horizontal lines.	51
4.7	Average SSIM of the reconstruction using the pre-filtering with NESTA for the filter banks tested. Average SSIM values for reconstructions with the NESTA TV minimization are shown as horizontal lines.	51
4.8	MRI reconstructions with pre-filtering including zero valued measurements. 1st column: Ground truth images. 2nd to 5-th columns: Reconstructions with zero-valued measurements with threshold respective to 0%, 0.5%, 1% and 5% of the maximum value of the filter. The label in the left indicates the filter bank used and the number of radial lines of the simulation of the row.	54

5.1	Isotropic filtering norm reconstructions using 20 radial projections.	64
5.2	MRI reconstructions. 1st row: ground truth images. 2nd to 5-th rows: TV minimization; iFN(x, h^2); iaFN(x, h^6); and iaFN(x, h^8). 1st to 5th columns: 40 radial lines and 20dB of added noise; 40 radial lines and 40 dB of noise; 20 radial lines and 40 dB of noise; 20 radial lines and 20 dB of noise; and 60 radial lines and 40 dB of noise.	68
5.3	Average SNR (top) and SSIM (bottom) values, with the corresponding confidence intervals (95%), computed between the ground truth and the images reconstructed using the filtering norm with filter banks h^1, h^2, h^6 , and h^8 for the iFN, aFN and iaFN models. The plot illustrates the values for all tested radial lines values, considering the no-noise (top of the bars) and the 20dB noise (middle of the bars) cases.	71
5.4	Average SNR (top) and SSIM (bottom) values, with the corresponding confidence intervals (95%), computed between the ground truth and the images reconstructed using the NESTA TV algorithm ($iFN(x, h^1)$) and the isotropic and anisotropic filtering with h^8 ($iaFN(x, h^8)$, best configuration) for the tested samplings and levels of noise.	72
5.5	Reconstruction times(s), grouped by the total number of coefficients of the filter banks. Outliers over 120s are not shown because the interval was chosen for the best visualization.	73
6.1	A signal x (top left), its decimation $x[2n]$ (middle left) and decimated and shifted $x[2n + 1]$ (bottom left). On the right sizes, the absolute value of the DFT of the signals in the right.	76
6.2	The lazy wavelet transform decomposition and reconstruction for a decimation factor of 2.	77
6.3	Pre-filtering with decimation model.	78
6.4	The 2D lazy wavelet transform decomposition and reconstruction for a vertical and horizontal decimation factor of 2.	83
6.5	Comparison of valid sampling rates compared to the original sampling rate of b .	84

List of Tables

1.1	The two types of filtering methods used with CS. It is detailed the combination of methods and type of filters, as well the chapter that the combination is presented.	5
3.1	Pre-filtering total reconstruction time (s) for TV filter.	31
3.2	Average reconstruction SNR(dB), SSIM, and time using pre-filtering with IRLS reconstruction algorithm, classified by radial lines. It is also shown the number of filters filter bank, the average ℓ_1 of the images and the spectrum coverage. The bolded face cells correspond to the filter banks with the highest SNR or SSIM.	37
4.1	Average number of measurements of the filter bank for a percentage of the total size of the image's spectrum.	48
4.2	Average reconstruction SNR(dB), SSIM, and time using pre-filtering with NESTA with zero threshold, classified by radial lines. The number of filters in each filter bank, the average ℓ_1 of the images and the spectrum coverage are also shown. Numbers in bold correspond to filter banks with the best values.	50
4.3	Difference of the average SNR (dB) and SSIM from the best reconstruction with threshold (Figures 4.6 and 4.7), and the reconstruction with zero threshold (Figures 4.4 and 4.5), classified by filter banks and radial lines. .	52
4.4	Average SNR (dB) of the reconstruction using pre-filtering with threshold. .	55
4.5	Average SSIM of the reconstruction using pre-filtering with threshold. .	56
5.1	Description of the set of filter banks used on our experiment.	66

5.2 ANOVA test between SNR/SSIM values for the image reconstructions, isolated by filter banks (left table) and cost functions (right table). Each cell has a pair of ‘1’, ‘0’, ‘-1’, with the first value corresponding to the ANOVA test results for SNR and the second for SSIM. The ‘1’ value means that the mean value of the corresponding row parameter is statistically higher ($p < 0.1$) than the corresponding column parameter. The ‘-1’ value means that the value to the corresponding row parameter is statistically lower than the one corresponding to the column parameter. Finally, the ‘0’ value means that there is no statistical difference between the two values. Green cells corresponds to ‘1’ to both SNR and SSIM, red cells to both ‘0’ or ‘-1’, and blue cells to one of the metrics corresponding to ‘1’.	67
5.3 ANOVA test between SNR/SSIM values for the image reconstructions of combinations of filter banks and cost functions. Each cell has a pair of ‘1’, ‘0’, ‘-1’, with the first value corresponding to the ANOVA test results for SNR and the second for SSIM. The ‘1’ value means that the mean value of the corresponding row parameter is statistically higher ($p < 0.1$) than the corresponding column parameter. The ‘-1’ value means that the value to the corresponding row parameter is statistically lower than the one corresponding to the column parameter. Finally, the ‘0’ value means that there is no statistical difference between the two values. Green cells corresponds to ‘1’ to both SNR and SSIM, red cells to both ‘0’ or ‘-1’, and blue cells to one of the metrics corresponding to ‘1’.	70
5.4 Median reconstruction time(s).	73
7.1 SNR(dB) and SSIM corresponding to the best results of the presented methods.	88

Acronyms

- aFN** anisotropic filtering norm. 60
- ANOVA** analysis of variance. 66
- CS** Compressed Sensing. 2, 13
- CT** Computational Tomography. 1
- DTFT** discrete time Fourier transform. 75
- iaFN** isotropic + anisotropic filtering norm. 62
- iFFT** inverse fast Fourier transform. 11
- iFN** isotropic filtering norm. 60
- IRLS** Iterative Re-weighted Least Squares. 4, 14
- MRI** Magnetic Resonance Imaging. 1
- NESTA** Nesterov's Algorithm. 2, 21
- NUFFT** non-uniform fast Fourier transform. 20
- RF** radio frequency. 10
- RIP** Restricted Isometry Property. 15
- SNR** Signal-to-Noise Ratio. 31
- SSIM** Structural SIMilarity index. 32
- TV** Total Variation. 2, 16, 24
- WIN** Filter banks designed with the windowing method. 26

Chapter 1

Introduction

1.1 Context and problem definition

Medical imaging make it possible to obtain a visual representation of internal organs, revealing structures that are not accessible or even hidden by other structures. These visual representations allow the specialist (*e. g.* the radiologist) to observe abnormalities in organs and tissues, as well as diagnose pathologies and follow-up medical treatments. Among the non-invasive medical imaging modalities, the most generally applicable for internal organs visualization are X-ray radiography, Computational Tomography (CT), and Magnetic Resonance Imaging (MRI) [1]. However, X-ray radiography and CT submit patients to ionizing radiation. Also, X-ray is more suitable for hard tissues such as bones, while CT allows the visualization of soft tissues. But in CT, the patient is submitted to higher radiation doses, that are often linked to cancer incidence [21].

MRI, on the other hand, does not emit ionizing radiation. Instead, MRI uses strong magnetic fields and radio frequency pulses to generate images of internal organs [22]. This allows performing imaging exams in cases for which the other imaging modalities are not recommended due to the radiation, like for example exams in pregnant women and children or a repeated series of exams. Another advantage of MRI is the high contrast of the reconstructed images. Different works [23, 24, 25], which compare the use of MRI and CT images for diagnosing diseases and injuries, showed that the levels of contrast of MRI allows a better anatomical identification of internal tissues, leading to better diagnose [26]. Moreover, the importance of MRI as diagnosis tool has been demonstrated for a large amount of diseases and conditions. Considering only the brain diseases [27], MRI has been used for diagnosis and treatment of Alzheimer's [28], psychopathy [29], multiple sclerosis [30], depression [31], vertigo and imbalance [32] research, to name a few. MRI is also widely used for presurgical planning, which often requires a high image quality of the area for planning more effective and less invasive surgical procedures [33].

In fact, MRI has been used for surgery of the brain, [34, 35, 36], breasts [37, 38], urological system [39], among others.

Nevertheless, the time required to perform an MRI exam has always been a concern. For each image, the technology relies on getting thousands of k-space measurements from the scanners, which are spaced by programmable radio-frequency pulses. A complete MRI exam can last up to hours. Also, the machine's magnet is very loud and the exam is generally performed in a cold room, which can be uncomfortable for patients. The exams also require the patient to stay still and some exams require that the patient hold his/her breath for several seconds. Movements may compromise the accuracy of the MRI measurements and, as consequence, the quality of the reconstructed images, generating images with undesired motion and blur artifacts [40].

A faster MRI acquisition can help to avoid the motion and blur artifacts, allowing for new types of exams. For instance, it would make possible to exam patients with tremors. Exams on claustrophobic patients can also became easier to perform. Finally, some exams that are performed by CT due to its faster nature could be substituted by an MRI exam and spare the patient from the unnecessary. Additionally, a faster acquisition time would also reduce the time required by an exam, and, therefore, reduce costs, since more exams can be performed per day. A faster exam also increases the patients comfort, since they would spend less time in uncomfortable environment. Faster MRI exams could also be more frequently used in emergencies, giving more detailed images to doctors in cases when the patient condition is critical.

Undersampling the signal in the k-space is a common approach to decrease the time required by an MRI exam. This reduces the number measurements and, therefore, the image quality can be compromised, leading to less details and artifacts in the image. One of the ways to surpass this problem is using Compressed Sensing (CS) techniques [2, 3, 41, 42]. CS takes advantage of the sparse structures of the images for an undersampled reconstruction [43]. One of the most popular CS algorithms for MRI is the Total Variation (TV) minimization [4, 2]. TV is suitable to reconstruct gradient sparse images, a common type of image in MRI. Since similar human body tissues have the same spin relaxation times and, therefore, are represented in the same gray level in MRI images [44]. As consequence, the acquired image is sparse to the gradient operator and, thus, easier to reconstruct with CS TV minimization. Several MRI reconstruction systems use TV minimization, at least as part of the CS reconstruction problem [5, 6, 7, 8, 9, 10]. Therefore, improving the TV minimization may improve the CS MRI reconstruction techniques.

Among the advances made in CS reconstruction algorithms, one of the most notable is the development of NESTA [11]. NESTA is a fast and reliable algorithm for solving

CS optimization problems. NESTA is based on the Nesterov's method [45], which solves general convex function minimization, allowing to solve different analysis and synthesis CS problems. For instance, the NESTA algorithm implements the TV minimization using a convex approximation of the finite differences operator. Although the method presents fast and accurate results, there is space for improvement and new applications. For example, the finite differences operator can be seen as a filtering operation, and therefore, other filtering operations can be implemented and tested. In other words, different filtering operations used with NESTA can improve the reconstruction quality in different situations.

One of the CS methods for MRI reconstruction that improves the reconstruction quality of undersampled MRI is the pre-filtering method, proposed by Miosso *et al.* [12]. The method consists in first applying a set of high-pass filters transformations directly to the measurements obtained from MRI machine (in the k-space domain). The filter banks are designed in such way that they cover all the spectrum, except for the low frequencies (that can be recovered from the original measurements). The next step of the method is to use CS to reconstruct the filtered versions of the images from the filtered measurements. Finally, the last step consists of recombining all filtered image reconstructions, along with original measurements, to reconstruct a single image. The method is based on the fact that the sparsity plays a major role in the CS reconstruction. Therefore, it is worth using high-pass filtered versions of the images because these high-frequency images are generally sparse. The method originally used the finite differences operator as a high-pass filter. The results show an improvement in the image quality when compared to the TV minimization.

Recently, two works used pre-filtering method for MRI image reconstruction [46, 47]. Costa [46] used a GPU parallel implementation of pre-filtering, reducing the reconstruction time. This was possible because the internal reconstructions of pre-filtering are independent. This work also compared the effect of approximating the MRI trajectories to a Cartesian grid. The results show that the approximated and non approximated trajectories present very reconstruction quality, suggesting that the Cartesian approximation can be used in simulations without compromising the reconstruction quality. A second work by Almeida [47] proposed using prior information with the pre-filtering method. The inclusion of the prior information allowed to reconstruct MRI images with less measurements while keeping the quality of the image. These works also show that the pre-filtering method is a very important tool and has potential to be explored and improved.

However, in the pre-filtering literature, a systematical experiment to test the filters that lead to the better reconstruction quality is still missing. The original implementation

use simple finite difference filters. Also, the known implementations of the pre-filtering methods use the Iterative Re-weighted Least Squares (IRLS) algorithm. Therefore, other algorithms could lead to improvements in the reconstruction quality. The ideas of the pre-filtering can also be used for other CS reconstruction algorithms. Instead of using the filters before the reconstruction, the filters can be used in the internal steps of the reconstruction algorithm.

1.2 Objectives

An important question in MRI reconstruction is which type of filters can lead to better reconstruction quality when using CS with filtering methods. My hypothesis is that filters which promote the sparsity to a signal or image generate images with a better quality when using the CS reconstruction. Therefore, it is possible to obtain images with higher quality than the images generated by CS filtering methods that use the finite difference filters.

That said, the main objectives of this thesis are: 1) to propose, simulate and analyze digital filtering methods in CS for undersampled MRI reconstruction; and 2) for each method, to design filter banks and analyze the quality of the reconstructed image. The overall goal is to improve the reconstruction quality when compared to the filtering methods that use finite difference filters, such as the original pre-filtering and the TV minimization. It is desirable to obtain a higher image quality with a fixed measurement rate, or a similar image quality with a lower number of measurements, which incurs in a decrease in the acquisition time of the exams.

The filtering methods in CS can be divided in two classes: 1) the pre-filtering methods, which the filtering operations occur before the CS reconstruction; and 2) methods which the filtering operations occur during the CS reconstruction. Another objective of this thesis is to evaluate separately these two classes of filtering methods and compare their results. It is also an objective to evaluate the filter parameters that could better explain the results.

1.3 Summary of the thesis contributions

To summarize the thesis contributions, Table 1.1 shows the combinations of methods and filters already tested in the literature (state-of-the-art) and the combinations of methods and filters the thesis contributes. Next, I describe in more details each of the contributions of this thesis.

Table 1.1: The two types of filtering methods used with CS. It is detailed the combination of methods and type of filters, as well the chapter that the combination is presented.

	Filter banks		
Pre-filtering methods:	Finite differences	Windowing method	Wavelet decomposition
Pre-filtering IRLS	State-of-the-art	Chapter 3	Chapter 3
Pre-filtering NESTA	Chapter 4	Chapter 4	Chapter 4
Pre-filtering NESTA + zero measurements	Chapter 4	Chapter 4	Chapter 4

	Filter banks		
Filtering on reconstruction:	Finite differences	Windowing method	Combinations
TV NESTA	State-of-the-art		
Filtering Norms	Chapter 5	Chapter 5	Chapter 5

Systematic pre-filtering evaluation

I designed and tested a very large set of filter banks in the pre-filtering CS method. These filter banks are divided in 2 classes. The first class consists of filters designed using the windowing method, formed by band-pass filters divided in equal parts. The second class of filters is based on the wavelet decomposition. In my simulations, filter banks designed with the windowing method and a low order obtained statistically significant better reconstruction quality than what was obtained by the filters tested by Miosso *et al.*[12].

Using NESTA with pre-filtering

I evaluated the pre-filtering method with the NESTA algorithm for ℓ_1 minimization. I tested using a large variety of filters in the pre-filtering with IRLS, and computed the image quality of both algorithms for the different types of filters. The NESTA algorithm did not achieve the image quality of the best results of IRLS.

Inclusion of zero-valued measurements in pre-filtering

I also proposed improvements to the pre-filtering method. I used the knowledge about the filters spectrum to include measurements in the solution space of the reconstruction problem. In the spectrum positions where the amplitude of the filter is below a given threshold, the method includes zero-valued measurements. I tested the method with the filters presented. The method improves the reconstruction quality for most of the filters.

Filtering norms

I proposed a method, called *filtering norm*, that uses (user specific) filtering operations as cost functions in the CS reconstruction. Just as the pre-filtering method, in the filtering norm, the filters also sparsify the signals, improving the reconstruction quality. The filtering norm generalizes the TV minimization, *i.e.* the TV is a specific case of filtering norm when the filter is the finite differences. I also implemented isotropic and anisotropic versions of the filtering norm. Results show that this proposed method is able to better reconstruct smooth images (a difficult task for TV minimization) using second order finite differences filters. They also show that using combinations of different types of filters can significantly improve the reconstruction quality for MRI images, when compared to standard TV minimization.

Decimation applied to pre-filtering

I performed a set of preliminary studies that includes a decimation in the pre-filtering methodologies. The decimation operation allows to cover more of the spectrum of the filter used in pre-filtering, while not changing significantly the sparsity in the space domain, that affects the reconstruction quality. I developed the mathematical theory of the method and proved that the decimation operation applied in the space domain has some restrictions in the frequency domain (where the measurements are taken and the pre-processing occurs). This is a work in progress and so far have not surpassed this issue.

1.4 Thesis organization

The thesis organization are given as follows.

In Chapter 2, I present the theoretical foundation and the state-of-the-art of this work. I explore the basic concepts of MRI, including the basic physics, the frequency encoding and the types of possible trajectories in the k-space. I also present the basic concepts of compressed sensing minimization problems, explaining the linear system and the different norms used in the problem. Next, I present the stages of the pre-filtering method, detailing the algorithm and the filters used by the authors, as well as the non-uniform pre-filtering. Next, I present the Nesterov's method for function minimization and its application on the NESTA algorithm, detailing the cost functions used in the algorithm, including the TV minimization.

In Chapter 3, I perform a systematic test of 43 filter banks, using the pre-filtering method with IRLS reconstruction algorithm. I describe the design of two types of filter banks: WIN, based on the windowing method with equally spaced bands; and the WAV,

based on wavelet decomposition filters. Finally, I describe the experiment, and analyze and discuss their results.

In Chapter 4, I describe the proposal of a technique that allows the use of filter features to improve the reconstruction results. I first change the representation of the original pre-filtering diagram to incorporate the proposed modifications, then I describe the method. Finally, I describe, analyze and discuss the experimental results of the simulations, comparing it with the results of chapter 3.

In Chapter 5, I describe the filtering norm minimization, a generalization of the NESTA TV minimization. I briefly review the NESTA method for direct reference. I describe the method, emphasizing the changes from the original NESTA algorithm, and detail the isotropic and anisotropic forms. I prepared experiments using synthetic and real data.

Chapter 6 describes a method that modifies the pre-filtering to include a decimation operator. The inclusion of decimation allows to de-sparsify the frequency domain, while not changing significantly the sparsity in the space domain. We present the mathematical formulation of the presented method, and the limitations. The chapter shows some directions for future works.

Chapter 7 presents the conclusions of this thesis, as well as the future works.

Chapter 2

Theoretical foundations and state-of-the-art

This chapter present the basic concepts for this thesis. First, the basic concepts of MRI are presented. These concepts help to understand the k-space trajectories used as sampling pattern for the undersampled MRI reconstruction problem.

The chapter also presents the concepts of CS minimization problems, a set of techniques for reconstruct undersampled data given that the signal can be represented in a sparse domain. CS is used for solving the undersampled MRI reconstruction problem in this work. A special case of CS is the TV minimization, which can be interpreted as a filtering operation in the internal steps of the CS reconstruction algorithm.

The chapter details the pre-filtering method for CS reconstruction, showing the filters used in the original work [12]. This method is used in Chapters 3, 4 and 6, with different filters. Next, the NESTA algorithm is presented, which is a robust and fast algorithm for solving the CS problem, especially the TV minimization. The algorithm is used in Chapter 4, and adapted in Chapter 5 for the filtering norms.

2.1 Magnetic resonance imaging

2.1.1 MRI Basic concepts

The human body is composed by, approximately, 70% of water (H_2O). MRI machines use the hydrogen atoms of water composition in the body tissues to obtain the images. Different tissues and organs have different concentrations of H_2O and that is how the MRI process can differentiate the tissues. A hydrogen nucleus is a single proton, which is a spinning charged particle, which has its own magnetic momentum. The magnetic momentum of a particle is called ‘spin’. Normally, the spin of protons in a body are

orientated randomly. Therefore, the total net magnetization vector ($M(t)$) of all protons (the macroscopic nuclear polarization) is close to zero, since the different protons spin are canceled. When these protons are in a magnetic field B , they act like a rotating pendulum, in a movement called precession. The protons precess in the Lamour frequency, which is proportional to the gyromagnetic constant (γ) and the applied field B ($\omega = \gamma B$). For the hydrogen atom, γ is 42.6 MHz/tesla [48].

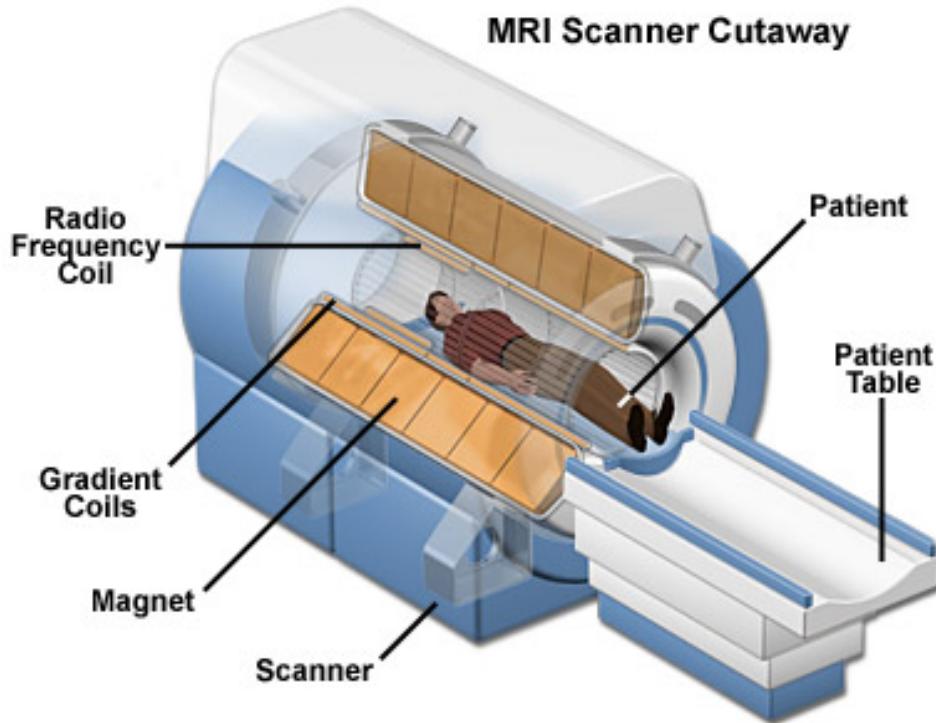


Figure 2.1: MRI scanner, showing also the coils that generates the magnetic fields and scans the output signal. Image source: [20].

An MRI machine scanner for human patients is illustrated in Figure 2.1. It contains a tube where the participant is placed during the exam. During the process, image shots of slices of participant's body are generated by taking samples from the protons reaction to the machine magnetic fields and radio frequency pulses. The machine has 3 classes of magnets. The first is a constant strong magnetic field B_0 , which has generally 1.5 to 7 tesla (by comparison, the earth magnetic field is about $0.5 \cdot 10^{-4}$ tesla). B_0 is parallel to the participant's table (z-axis, or longitudinal orientation) and is generated by a superconducting electromagnet. Therefore, when the machine is turned on, the participant body receives a $\langle 0, 0, M_0 \rangle$ magnetic field. As a result, the spin of protons align with the z axis and precess at frequency $\omega_0 = \gamma B_0$. When they reach the equilibrium, the net magnetization vector M has a predominant component: a fixed value $\langle 0, 0, B_0 \rangle$ aligned with the longitudinal axis. The longitudinal (M_z) and the transversal (M_{xy})

net magnetization, at a given time, can be obtained by solving the following differential equation [49]:

$$\frac{dM}{dt} = \gamma M_x B - i \frac{M_x}{T_2} - j \frac{M_y}{T_2} - k \frac{M_z - M_0}{T_1}, \quad (2.1)$$

whose solutions are $|M_z(t)| = M_0(1 - e^{-\frac{t}{T_1}})$ and $|M_{xy}(t)| = M_0(e^{-\frac{t}{T_2}})$. T_1 is the relaxation decay time that dictates how the spin orientation goes back to equilibrium. Over time, the protons start to precess out of synchrony, so the combined magnetization (in transversal plane) contribution decays exponentially in time. T_2 is the relaxation time related to this decay. When the equilibrium is achieved, M_{xy} converges to 0 and M_z converges to M_0 , so, the net magnetization vector comes back to $\langle 0, 0, M_0 \rangle$. The T_1 and T_2 relaxation time curves vary with the tissue's composition and structure. For example, hydrogen protons from fat tissues achieve the equilibrium faster (T_1 and T_2 are smaller) than hydrogen protons from water. Using the T_1 decay curve (sensing the signal in the M_z axis), water appears close to black in the images, while fat appears close to white. Using the T_2 decay curve, water appears close to white and fat tissues appear gray. In the database of images used in this work, the images are acquired by sensing in the M_{xy} plane, in other words, using a T_2 decay curve.

Another set of magnets in the MRI machine are the gradient magnets. They are orientated in 3 axis, allowing to change the orientation and slice localization of acquired images. The gradient field (G) alters the strength of the primary magnetic field at each $\langle x, y, z \rangle$ position. As a result, the precession frequency of protons varies with position. The precession frequency is given by $\omega(x, y, z) = \gamma(B_0 + G(x, y, z))$. We use the Gradient field to perform a slice selection and the spatial-frequency coding, as explained in the next section.

The third set of magnets in an MRI machine are radio frequency (RF) coils. They are used for transmitting the radio frequency pulses and for receiving the signals in MRI. The frequency of magnetic pulse generated by the RF coils should be in resonance with the protons spin frequency. This generates a constructive wave interference and, as consequence, amplifies the magnitude of the net magnetization vector, generating a signal that is distinguishable from noise. When the RF magnetic pulses are taken off, the protons start to relax, and the net magnetization converges to $\langle 0, 0, M_0 \rangle$, following the exponential curves $|M_z|$ and $|M_{xy}|$. The variation of the net magnetization vector induces an electrical signal on the RF receiver coils. As consequence, a signal proportional to the net magnetization can be acquired and an image can be formed.

2.1.2 Frequency encoding and trajectories

In order to obtain an axial image from a participant, the slice z_0 (of the z-axis) is chosen by tuning the G field accordingly. For example, we can choose G varying linearly in z-axis in such way that ω_0 is equal to $\gamma(B_0 + G(x, y, z_0))$ when $z = z_0$, and different for $z \neq z_0$. Thus, when a RF pulse in a narrow range around frequency ω_0 is emitted, only the protons on the selected slice resonate with the pulse.

After selecting the plane (z_0), to form an image we have to differentiate the net magnetization M_{xy} of each spatial position in this plane. Since it is not possible to do this directly, we use a frequency encoding of spatial positions. The frequency domain of an MRI is known as k-space. For a given point (k_x, k_y) in the k-space, the signal read in the transversal receivers is given by the following expression:

$$\hat{M}_{xy}(k_x, k_y) = \int_{z_0 \text{ plane}} M_{xy} e^{-2\pi j(k_x x + k_y y)} dx dy. \quad (2.2)$$

These are Fourier coefficients at frequency points (k_x, k_y) . However is not possible to obtain the image from a single Fourier coefficient. To get more k-space points, we have to vary the transversal gradient fields G_x and G_y over time and read the signal over time to get several k-space measurements. The variation of the (k_x, k_y) over time is related to the gradient fields in the following way:

$$\begin{aligned} k_x(t) &= \gamma / 2\pi \int_0^t G_x(\tau) d\tau, \\ k_y(t) &= \gamma / 2\pi \int_0^t G_y(\tau) d\tau. \end{aligned} \quad (2.3)$$

Thus, the k-space coefficients must be taken by following a curve in the z_0 plane, known in MRI as trajectories. To set the trajectory $(k_x(t), k_y(t))$ over time we have to set $G_x(t)$ and $G_y(t)$ accordingly. Figure 2.2 shows examples of $G_x(t)$ and $G_y(t)$ and the generated trajectories in k-space.

The most commonly used trajectory in MRI is the rectangular trajectory, also known as the Cartesian trajectory, which is shown in Figure 2.2 (a). The k-space points are sampled equally spaced and chosen to match the frequencies of a discrete Fourier transform. Thus, the image is reconstructed with an inverse fast Fourier transform (iFFT) if all lines have been sampled. However, undersampling some lines for faster acquisition generates aliasing. Also, natural images generally have most energy concentrated in low frequencies. Therefore, an equally spaced sampling like the rectangular trajectory is not the ideal spreading of sampling points in the k-space. Rectangular sampling also presents abrupt changes of the gradient fields in the transition between lines. The MRI machines have restrictions on the maximum variation

of the gradient fields. Therefore, there are hardware constraints for the implementation of the rectangular sampling.

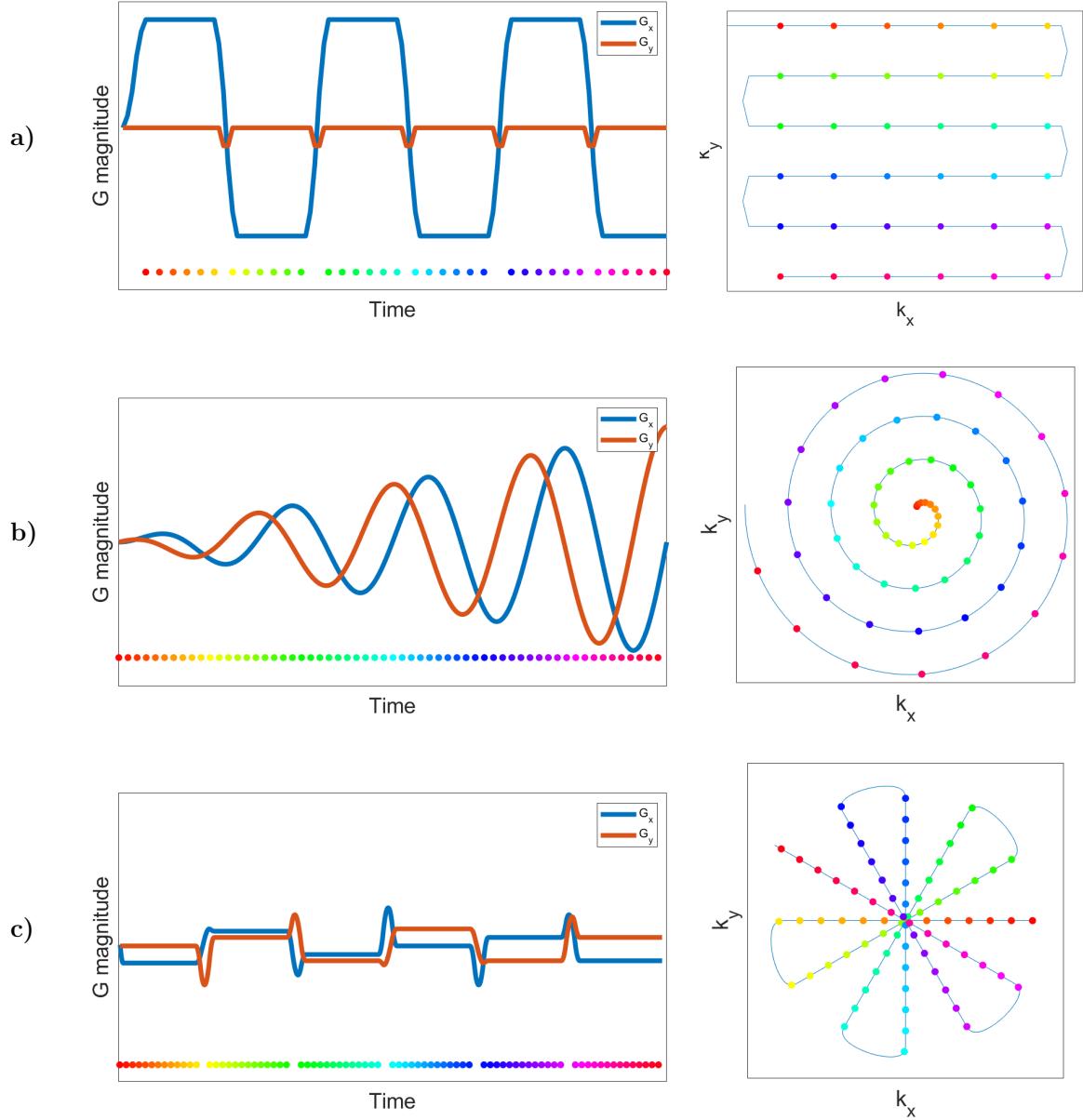


Figure 2.2: Rectangular (a), spiral (b) and radial (c) samplings. Magnitude of gradient field is shown in left, and the corresponding sampling trajectories scheme in right. Equal color points represent the same time.

Figure 2.2 (b) shows a non-Cartesian trajectory, the spiral trajectory with 1 branch. Each branch of the spiral trajectory is acquired from a single excitation/echo time and covers all regions of the spectrum. In comparison, the other sampling schemes require the time of one excitation/echo for sampling only a single line of the spectrum. If we want a denser sampling, there are also implementations of the spiral with several branches. Other variations include changing the radius of the spiral with non-linear functions. Also,

notice that the gradients $G_x(t)$ and $G_y(t)$ of the spiral sampling are smooth, therefore this sampling is more flexible in terms of hardware constraints.

Radial sampling is the third trajectory shown in Figure 2.2 (c). This trajectory was commonly used in the initial stages of the MRI technology, when the reconstruction was similar to CT reconstructions, which are based on the central slice theorem [50]. In the radial trajectory, every line passes through the center of the k-space. Therefore, it is always updated, which helps to reduce motion artifacts. Also, each radial line has equal contribution to the image reconstruction, since each line passes through all frequencies in a given angle, containing high and low frequencies. The central part (the low frequencies) of the spectrum is oversampled. Therefore, the region of the spectrum where natural images concentrate the major part of its energy is the most sampled region. These features allow real time MRI reconstruction [51, 52, 53, 54]. Additional strategies allow to increase the incoherence (defined in the next section) in a compressed sensing reconstruction. In fact, the radial sampling has been recently incorporated into commercial MRI systems [55, 56].

2.2 Compressed sensing

Nyquist-Shannon sampling theorem states that, if a signal is captured at a rate at least 2 times faster (Nyquist rate) than its bandwidth, it can be reconstructed without losses [57]. In many cases, the required Nyquist rate is too high, requiring sensors that are too expensive or demanding a long time to acquire enough measurements. Compressed Sensing (CS) [58, 59, 60, 41, 42] is a technique for acquiring and representing signals at a rate significantly lower than the Nyquist rate. The technique allows, with high probability, to reconstruct a signal from fewer measurements than the Nyquist rate, by using optimization techniques.

To correctly reconstruct the signal, CS theory states that the signal should be sparse when represented in a certain domain. A signal is sparse in a domain if it is represented by a few non-zero coefficients. By definition, a k -sparse signal has exactly k non-zero coefficients in this domain.

Let n be the length of a sparse signal x to be acquired. Suppose that the signal is k -sparse in the space domain, where $k \ll n$. Let b be a linear projection of x in a set of m ($m < n$) vectors of a sampling matrix \mathbf{M} , which defines how the signal is sampled. We can represent b as:

$$b = \mathbf{M}x, \quad (2.4)$$

where b is a vector of size m . The goal of the reconstruction stage is to find the solution x that matches with the samples $b = \mathbf{M}x$. This is an undetermined problem [61] and, therefore, there is an infinite number of solutions.

The solution of this problem relies on sparsity, *i.e.* among all the possible solutions, the sparsest solution is assumed to be the correct solution. The sparsity can be directly measured by the metric ℓ_0 , which is a metric that counts the number of non-zero elements of a vector. So, the reconstruction problem can be described by the following equation:

$$\begin{aligned} & \operatorname{argmin}_x \|x\|_0 \\ & \text{subject to } \mathbf{M}x = b \end{aligned} \quad (2.5)$$

This problem can be solved reliably for $m = 3k$, given that x is k -sparse. However, Equation 2.5 describes an unstable NP-complete problem [62], which requires the enumeration of all $\binom{N}{K}$ possible positions of non-zero entries of x .

A second solution is based on the ℓ_1 metric minimization. The ℓ_1 computes the sum of the magnitude of each vector element. It was proven [60] that for sparse signals, the minimum ℓ_0 occurs in the same points as the minimum ℓ_1 , so that minimizing ℓ_1 leads to the same solution as minimizing ℓ_0 . The metric ℓ_1 can be expressed as:

$$\begin{aligned} & \operatorname{argmin}_x \|x\|_1 \\ & \text{subject to } \mathbf{M}x = b \end{aligned} \quad (2.6)$$

Alternatively, we can use other metrics on the CS minimization, which lead to the same solutions as ℓ_0 and ℓ_1 . We can use the metric ℓ_p ($0 < p < 1$), which is defined by:

$$\|x\|_p = \left(\sum_{i=1}^N |x_i|^p \right)^{\frac{1}{p}}. \quad (2.7)$$

Despite the faster reconstruction, ℓ_1 minimization generally requires more measurements than the ℓ_0 minimization. In the ℓ_p minimization, by selecting the p value accordingly, we can choose a balance between the number of required measurements and the reconstruction time. For lower p values, less measurements are required and the reconstruction time is higher [14]. Naturally, as the p value is increased (making it closer to 1), more measurements are needed, and the reconstruction time is shorter. A common reconstruction algorithm that solves problem in Equation 2.6 is the Iterative Re-weighted Least Squares (IRLS) [14, 63].

A more generic CS problem is the quadratic constraints problem, where the goal is to search for the sparsest solution in the range $\|b - \mathbf{M}x\| < \epsilon$ for a small value ϵ . The ℓ_1

quadratic constraints problem is given by:

$$\begin{aligned} & \operatorname{argmin}_x \|x\|_1 \\ & \text{subject to } \|b - \mathbf{M}x\| < \epsilon \end{aligned} \quad (2.8)$$

This problem is more suitable for real world signals, which are frequently quasi-sparse. These signals have a few large coefficients, with the majority of coefficients below a small threshold. The quadratic constraints problem in Equation 2.8 allow to find sparse approximations of these signals even for noisy measurements¹. Notice that, when ϵ is zero, the exact CS problem is a special case of the quadratic constraints problem.

2.2.1 General domain of sparsity

Until the previous section, x was considered sparse in the space domain. However, a more general formulation can be applied to represent and reconstruct a signal x that is sparse in any domain.

Let us suppose that x is sparse in the domain \mathbf{W} , *i.e.* the signal $\alpha = \mathbf{W}x$ has only a few coefficients different from zero. The coefficients α with the minimum ℓ_1 norm in the solution space are obtained by solving the problem:

$$\begin{aligned} & \operatorname{argmin}_\alpha \|\alpha\|_1 \\ & \text{subject to } \|b - \mathbf{M}\mathbf{W}^H\alpha\| < \epsilon \end{aligned} \quad (2.9)$$

The signal x can be obtained by applying the inverse transformation \mathbf{W} to α . This problem is known as the synthesis problem [11].

However, CS requires some conditions to be satisfied in order to solve the above problem. One of these conditions is the Restricted Isometry Property (RIP) [42, 2], which states that, for any v that is $3k$ -sparse, $\Theta = \mathbf{M}\mathbf{W}^H$ must be such as:

$$1 - \epsilon \leq \frac{\|\Theta v\|}{\|v\|} \leq 1 + \epsilon, \quad (2.10)$$

for a small values of ϵ ($\epsilon > 0$). In other words, Θ must preserve the sizes of any k -sparse vectors [58]. In practice, we cannot test this property for every v vector, because this would require to test every combination of sparse signals. So, we choose a pre-defined matrix \mathbf{M} , which is proven to satisfy RIP [60].

Other necessary condition that the CS theory requires to correctly solve the problem is the incoherence property [43], which requires that the correlation between \mathbf{M} and \mathbf{W}

¹In fact, this approach is commonly used in denoising algorithms, even in fully sampled signals, where CS is not applicable.

has a low value. To satisfy the incoherence, we can choose \mathbf{M} in such a way that the representation of the lines of \mathbf{W} are not sparse. The MRI acquisition can be performed in such a way that it satisfies both RIP and incoherence properties, depending on the sampling trajectory on the k-space [9].

A different approach is known as the analysis problem [11], given by:

$$\begin{aligned} & \operatorname{argmin}_x \|\mathbf{W}x\|_1 \\ & \text{subject to } \|b - \mathbf{M}x\| < \epsilon \end{aligned} \quad (2.11)$$

In this problem, x is obtained directly. When \mathbf{W} is an invertible basis ($\mathbf{W}^{-1} = \mathbf{W}^T$), the problems in Equations 2.9 and 2.11 are equivalent and an ℓ_1 minimization in the form of the problem in Equation 2.9 correctly solves the analysis problem. However, when \mathbf{W} is not an invertible basis, we cannot use the ℓ_1 reconstruction. Therefore, specific algorithms have to be designed to solve the problem. The analysis problem is usually more useful than the synthesis one because it allows the use sparsifying functions not restricted to the invertible basis, like, for instance, overcomplete dictionaries. In fact, Elad *et al.* [64] have shown that solving an analysis problem have advantages over solving a synthesis problem.

2.2.2 Total Variation minimization

One of the analysis CS problems is the Total Variation (TV) minimization. MRI images are generally gradient sparse, *i.e.* the finite differences in the x and y axis are close to zero for most pixels. This happens because the human organs have similar properties, which leads to the same level of gray. For this reason, a popular sparsity metric is the total variation (TV) [2]. The TV reconstruction problem becomes the problem of finding an image (x) that fits the measurements (b) with the lowest TV metric.

The TV metric has been widely used in two different models, the isotropic TV [4, 2] and the anisotropic TV [65, 66, 67]. Let us first define the discrete finite differences operator:

$$\mathbf{D}x = \begin{bmatrix} D_h(x) \\ D_v(x) \end{bmatrix}, \quad (2.12)$$

where

$$\begin{aligned} D_h(x) &= x(i+1, j) - x(i, j) \\ D_v(x) &= x(i, j+1) - x(i, j) \end{aligned} \quad (2.13)$$

The isotropic TV can be defined as the ℓ_2 of the finite differences $\|\mathbf{D}x\|_2$, while the anisotropic is the $\|\mathbf{D}x\|_1$. Each metric has its advantages over the other [68]. Once

chosen the type of TV metric, the minimization problem can then be defined as:

$$\begin{aligned} & \operatorname{argmin}_x \|x\|_{TV} \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon \end{aligned} \quad (2.14)$$

In this thesis, if not explicitly mentioned otherwise, TV refers to the isotropic TV.

2.3 The pre-filtering method

Miosso *et.al* [12] proposed another CS method for reconstructing gradient sparse MRI images, which provides better results than TV minimization. In the method, the authors proposed processing the measurements before the reconstruction. The pre-processing is performed by performing finite differences filtering operations on the measurements. The method reconstructs the pre-filtered versions of the measurements, resulting in filtered versions of the original image, and recomposes the original image with the filtered reconstructions and the original measurements. The idea is that sparse versions of the signal generate good quality reconstruction.

Let h_k be a list of 2D finite difference filters. These filters are meant to sparsify a gradient sparse image in the image domain. Figure 2.3 shows the effect of filtering an image with finite difference filters. Note that the filtered versions are sparse and, therefore, can be reconstructed with an ℓ_p minimization.

Let b be the measurements of the original image, taken in the sample positions τ . The coefficients b_k of each filter h_k are obtained by:

$$b_k = (H_{k \in \tau}) \circ b, \quad (2.15)$$

where H_k is Fourier transform of h_k , \circ is an element-wise multiplication, and $H_{k \in \tau}$ is composed of the elements of H_k corresponding to the sampled positions τ . For each b_k , the reconstructed image \hat{x}_k can be obtained by minimizing the following optimization problem:

$$\begin{aligned} \hat{x}_k &= \operatorname{argmin}_{x_k} \|x_k\|_p^p \\ &\text{subject to } b_k = \mathbf{M}x_k. \end{aligned} \quad (2.16)$$

Supposing that each x_k (the filtered version of the signal x , when filtered by h_k) is sparse, an ℓ_p reconstruction in the image domain can correctly reconstruct the images \hat{x}_k .

The last step is the final image composition. This step is formed by combining the reconstructed images and the original measurements. The original image can be reconstructed by completing the spectrum of \hat{x} with either b or with the coefficients

from \hat{x}_k , as described in Algorithm 1. Notice that the algorithm divides the spectrum of X_k by the filter spectrum, normalizing the amplitude of the filtered solution and correcting any time shift generated by the filtering operation on measurements. As the finite differences are high-pass filters, the high frequency components are filled by x_k . The low frequency components are completed by b . Therefore, sampling patterns that have sampling concentration at low frequencies favor this method (for example, the radial lines sampling).

In the original work [12] tests two filters banks. First, the horizontal and vertical finite differences (TV filters):

$$h_1 = \begin{bmatrix} 1 & -1 \end{bmatrix} \text{ and } h_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad (2.17)$$

corresponding to the diagram shown in Figure 2.4.

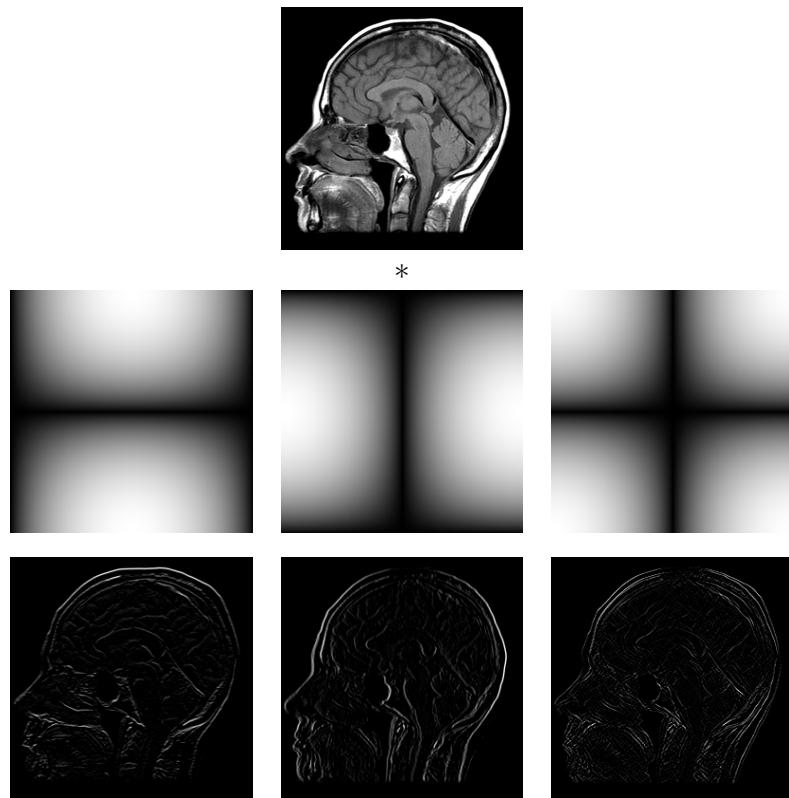


Figure 2.3: High-pass filtered versions of the original image. First row shows the original image, second row shows the spectrum of the filters and third row shows the filtered versions of the original image, filtered by the filter shown in the second row. Note that each one of the filtered images are sparser in pixel domain than the original image.

Algorithm 1 Final image composition algorithm.

Input: $b, \hat{X}_1, \hat{X}_2, \dots, \hat{X}_n, \hat{H}_1, \hat{H}_2, \dots, \hat{H}_n$

Output: \hat{X}

```

1: for each  $u, v$  position do
2:   if  $u, v$  are sampled points then
3:      $\hat{X}(u, v) = b(u, v)$ 
4:   else
5:      $k = \max\_idx(||\hat{X}_1(u, v)||, ||\hat{X}_2(u, v)||, \dots, ||\hat{X}_n(u, v)||)$ 
6:      $Y = \hat{X}_k(u, v)$ 
7:      $\hat{X}(u, v) = Y(u, v)/H_k(u, v)$ 
8:   end if
9: end for

```

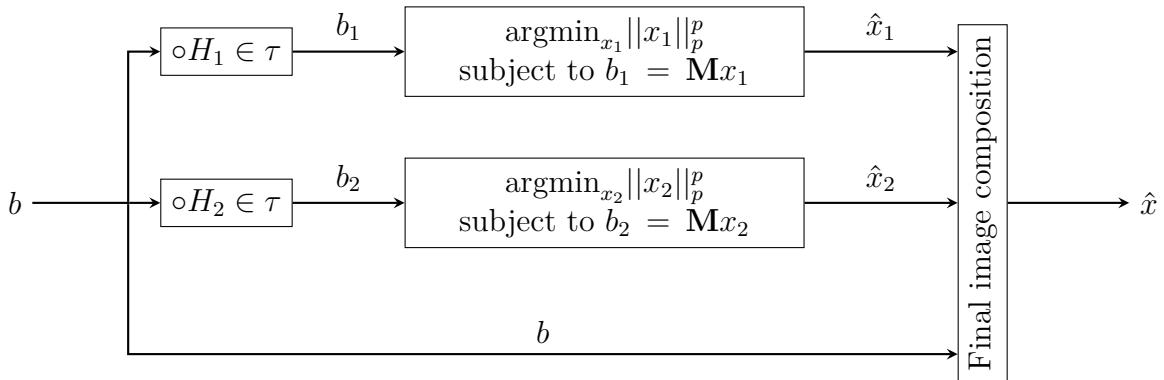


Figure 2.4: Reconstruction using pre-filtering procedure using vertical and horizontal finite differences filters.

The other tested model is a high-pass filter bank from the Haar 2D wavelet family (Haar filters):

$$h_1 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, h_2 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \text{ and } h_3 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \quad (2.18)$$

The example shown in Figure 2.3 illustrates the filtering step used by the Haar filters and Figure 2.5 shows their pre-filtering model.

Miosso *et al.*'s [12] compared the pre-filtering method using the two mentioned filter banks and the standard TV minimization (given by Equation 2.12) with the ℓ_1 -MAGIC implementation [69]. Pre-filtering with both filter banks produced better image reconstruction qualities (measured in SNR) than what was produced by the ℓ_1 -MAGIC TV minimization. More specifically, the Haar filter bank presented the best results.

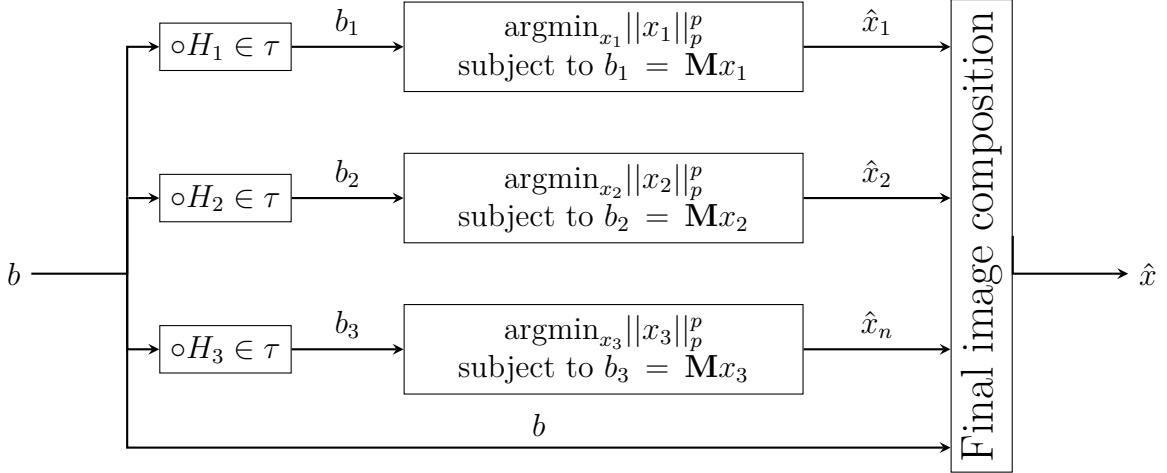


Figure 2.5: Reconstruction using pre-filtering procedure using Haar filters.

2.3.1 Non-uniform Pre-filtering

Costa [46] used a feature that was not commonly explored in the literature. He managed to reconstruct MRI images with the pre-filtering approach using a non-uniform fast Fourier transform (NUFFT) [70]. For most trajectories used in MRI, the samples taken by the MRI machines are collected in a non-uniform 2D k-space, therefore, they usually do not fit a rectangular grid. In simulated CS MRI reconstructions, spiral and radial trajectories are calculated as approximations to match the rectangular grid. This grid approximation is very popular in the CS MRI literature [71, 72, 73, 74, 5].

Costa used a non-Cartesian grid to reconstruct MRI images with the pre-filtering method. He used the NUFFT instead of the discrete Fourier transform in the CS models. Because his approach is not limited to a rectangular grid, Costa used the measurements in the original locations of the spiral trajectories and compared the results to measurements approximated by a grid. He expected better results using measurements in the original trajectory. For some images, the approximated grid present a slightly better quality, for others, the original locations. The difference between the qualities in SNR (dB) are less than 1 dB. Therefore, he concluded that reconstructions using an approximated grid are consistent with results of original locations and can be used instead.

Since Costa's results showed that the quality of reconstructions did not change drastically when using grid approximations measurements, in this work, I use only Rectangular grid approximated locations for the measurements. This approach is faster because the FFT is faster than the NUFFT.

2.4 The NESTA Algorithm

Among the advances made in CS problems in the last ten years, one of the most notable is the development of NESTA [11]. NESTA is a fast and reliable algorithm for solving CS optimization problems. When compared to other methods, the main advantage of NESTA is its flexibility. Since NESTA is based on the Nesterov's method [45] for general convex function minimization, it can solve both synthesis and analysis problems. The algorithm implements a solution to the isotropic TV minimization problem using a convex approximation of the finite differences as a cost function and a 2-overcomplete sparse matrix \mathbf{D} .

NESTA is the main algorithm I use for CS reconstruction in this thesis. In Chapter 5 I also present a CS reconstruction method based on NESTA. Therefore, it is important to describe this method in details. NESTA [11] aims to solve the ℓ_1 basis pursuit compressed sensing optimization problem, defined as:

$$\begin{aligned} & \operatorname{argmin}_x \|x\|_1 \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon, \end{aligned} \tag{2.19}$$

where \mathbf{M} is the measurement matrix, which generates b from x , and ϵ is proportional to the estimated measurement noise. The ℓ_1 norm solves the CS optimization problem in polynomial time.

2.4.1 The Nesterov's method for function minimization

NESTA is based on Nesterov's method [45], which is a fast convex minimization solver. More specifically, Nesterov's optimization method is a first-order method for finding the minimum of a smooth convex differentiable function f in a convex set Q_p , with the assumption that f is L -Lipschitz². The general minimization problem is given by:

$$\begin{aligned} & \operatorname{argmin}_x f(x). \\ & x \in Q_p \end{aligned} \tag{2.20}$$

The method iteratively calculates solutions x_k using a first-order method solution (i.e. based on $\nabla f(x_k)$) and weighted averages of the previous calculated gradients. When a selected number of k -dependent weights is used, the convergence rate goes from $O(1/k)$ to $O(1/k^2)$ [75, 76].

² $f(x)$ is Lipschitz with constant L , or L -Lipschitz, when $\|\nabla f(x) - \nabla f(y)\|_{\ell_2} \leq L\|x - y\|_{\ell_2}$ for any x and y

Later, the Nesterov method was extended to handle non-smooth functions [76]. In this case, the function $f(x)$ must be written in the following form:

$$f(x) = \max_{u \in Q_d} \langle u, \mathbf{W}x \rangle, \quad (2.21)$$

where Q_d is a convex set. To make this function smooth, a term is added to f , as follows:

$$f_\mu(x) = \max_{u \in Q_d} \langle u, \mathbf{W}x \rangle - \mu p_d(u), \quad (2.22)$$

where $p_d(u)$ is a strongly convex function in Q_d , with a convexity parameter σ_d [11]. The modified function $f_\mu(x)$ is continuously differentiable and is Lipschitz with constant $L_\mu = \|\mathbf{W}\|/\mu\sigma_d$. That said, the minimization problem can be solved for an approximation of the non-smooth $f(x)$ function for a given parameter μ with a fast convergence rate.

2.4.2 NESTA

Becker *et. al* [11] used Nesterov's method to solve compressed sensing basis-pursuit problems (refer to Equation 2.19), which is known as the NESTA algorithm. The authors re-wrote the ℓ_1 function as:

$$\begin{aligned} \|x\|_1 &= \max_{u \in Q_d} \langle u, x \rangle \\ u \in Q_d &= \{u : \|u\|_\infty \leq 1\}. \end{aligned} \quad (2.23)$$

After adding the smooth term, the minimization function becomes:

$$\begin{aligned} f_\mu(x) &= \max_{u \in Q_d} \langle u, x \rangle - \mu p_d(u) \\ u \in Q_d &= \{u : \|u\|_\infty \leq 1\}. \end{aligned} \quad (2.24)$$

In the NESTA formulation, $p_d(u) = \frac{1}{2}\|u\|_2^2$ is chosen because this function is strongly convex with a convexity parameter $\sigma_d = 1$ and a minimum value at $u = 0 \in Q_d$. With this choice, Equation 2.24 becomes the Huber function [77], that is Lipschitz with constant L_μ/μ , and its gradient is given by:

$$\nabla f_\mu(x(i)) = \begin{cases} \mu^{-1}x(i), & \text{if } |x(i)| < \mu \\ \text{sgn}(x(i)), & \text{otherwise} \end{cases}. \quad (2.25)$$

To solve the problem in Equation 2.24, we have to find the minimum solutions of f_μ that match the quadratic constraints of the measurements. In other words, the CS

problem becomes:

$$\begin{aligned} & \operatorname{argmin}_x f_\mu(x), \\ & x \in Q_p = \{x : \|b - \mathbf{M}x\|_2 \leq \epsilon\}. \end{aligned} \quad (2.26)$$

Using Nesterov's method, this problem is solved iteratively in 4 steps. Starting from a initial value x_0 , every iteration generates the next solution x_k by calculating $\nabla f(x_k)$ and 3 sequences: y_k , z_k and x_k . The first step is to calculate $\nabla f(x_k)$. The second step is to find the solution y_k using a first-order method, given by:

$$y_k = \operatorname{argmin}_{x \in Q_p} \frac{L_\mu}{2} \|x_k - x\|_2^2 + \langle \nabla f_\mu(x_k), x - x_k \rangle. \quad (2.27)$$

The term $\langle \nabla f_\mu(x_k), x - x_k \rangle$ assumes lower values when the vector $x - x_k$ points in the opposite direction of the gradient $\nabla f_\mu(x_k)$ (so, y_k tends to point to $-\nabla f(x_k)$). The minimization of the term $\|x_k - x\|_2^2$ ensures that the steps are not too large, so that the value $\nabla f_\mu(x_k)$ is accurate enough for the current iteration of the algorithm.

The next step consists of finding a solution that considers weighted averages of the previous calculated gradients $\sum_{i=1}^k \alpha_i \nabla f_\mu(x_i)$. This solution avoids the typical zig-zag that occurs in blind gradient decent methods, speeding-up the convergence. This step, calculated on the sequence z_k , consists of solving the following problem:

$$z_k = \operatorname{argmin}_{x \in Q_p} \frac{L}{\sigma_p} p_p(x) + \left\langle \sum_{i=0}^k \alpha_i \nabla f(x_i), x - x_k \right\rangle. \quad (2.28)$$

For a faster convergence rate ($O(1/k^2)$), the weighting factors $\alpha_i = \frac{k+1}{2}$ were chosen by a theoretical analysis of the problem [75, 76]. The function $p_p(x)$ is called the *prox-function* of Q_p , which is chosen to be smooth and strongly convex on Q_p , with convergence rate σ_p . In NESTA, $Q_p = \{x : \|b - \mathbf{M}x\|_2 \leq \epsilon\}$ and, therefore, a quadratic choice for $p_p(x)$ is chosen as:

$$p_p(x) = \frac{1}{2} \|x - x_0\|_2^2, \quad (2.29)$$

where x_0 is the initial solution. This factor guarantees that the solution does not move too far away from the initial central solution. Also, with this choice of $p_p(x)$, the convergence parameter σ_p is equal to 1.

Finally, NESTA finds x_k by averaging the value of y_k and z_k , as given by the following equation:

$$x_k = \tau_k y_k + (1 - \tau_k) z_k. \quad (2.30)$$

If $\mathbf{M}^H \mathbf{M}$ is an isometry, y_k and z_k do not require an iterative process and can be calculated in a single step at each inner loop. NESTA also uses a method named continuation, where it first calculates iterations with different values of μ in order to refine and speed the

process. These features make NESTA a very efficient algorithm that converges fast and can handle large data sizes.

Although it was originally designed to solve the ℓ_1 minimization (with the approximation given by Equation 2.24), by changing its cost function, NESTA can solve different problems. For example, by evaluating ∇f_μ on the point $\mathbf{W}x$ (using $\mathbf{W}x(i)$ in Equation 2.25 instead of $x(i)$), NESTA can solve the following analysis CS problem:

$$\begin{aligned} & \operatorname{argmin}_x \|\mathbf{W}x\|_1, \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon. \end{aligned} \quad (2.31)$$

2.4.3 NESTA for Total Variation minimization

Another problem that can be solved using NESTA, simply by changing the cost function, is the Total Variation (TV) minimization problem. The TV minimization problem included in NESTA is the isotropic TV.

The approximation to the isotropic TV norm used in NESTA is expressed in the following form:

$$\|x\|_{TV} = \max_{u \in Q_d} \langle u, \mathbf{D}x \rangle, \quad (2.32)$$

where Q_d is $\{u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} : u_1^2 + u_2^2 \leq 1\}$. As Nesterov's method assumes the function is smooth, the authors added the smoothing term $p_d(u) = \frac{1}{2}\|u\|_2^2$. Therefore, the TV norm approximation that NESTA minimizes is the following:

$$\|x\|_{TV} \approx f_\mu = \max_{u \in Q_d} \langle u, \mathbf{D}x \rangle - \frac{\mu}{2}\|u\|_2^2, \quad (2.33)$$

and its gradient is given by:

$$\nabla f_\mu = \begin{cases} \mu^{-1} \mathbf{D}^H \mathbf{D}x, & \text{if } \|\mathbf{D}x\| < \mu \\ \|\mathbf{D}x\|_2^{-1} \mathbf{D}^H \mathbf{D}x, & \text{otherwise} \end{cases}. \quad (2.34)$$

The algorithm to solve this problem is the same used for the ℓ_1 problem. The only change is the calculation of ∇f_μ in the first inner step. The calculations of y_k , z_k , and x_k are performed in the same way, but using Equation 2.34 instead of Equation 2.25.

Chapter 3

Filter banks for pre-filtering

In this chapter, I propose a methodology to test a large number of filter banks using the pre-filtering method. In particular, I want to analyze filter parameters that lead to a better (or worse) quality of the reconstructed images, considering a constant number of measurements.

Furthermore, this chapter presents the set of 2D filter banks used in our tests, which consists of filters designed with the windowing method and walevet decomposition filters. Part of the results corresponding to the wavelet filters were published in [78], while part of the results for both types of filters were published in [19].

Additionally, I propose an experimental methodology to systematically evaluate these filter banks for the pre-filtering method. I present the parameters of analysis related to the quality of reconstructed images, the spectrum coverage of the filter banks and the average sparsity of the filtered images. Finally, I present the results obtained following the proposed experimental methodology.

3.1 Filter banks

The pre-filtering model presented in Figures 2.4 and 2.5 are specific for a limited number of filters. Let us first generalize the model for any filter bank. Figure 3.1 shows a generic pre-filtering model that will be used in this chapter. The filter banks presented are labeled as H_1 to H_n in this figure. In particular, a filter H_k affects the original measurements only in the sampling positions, *i.e.*

$$b_k = (H_k \in \tau) \circ b, \quad (3.1)$$

as seen in the Section 2.3.

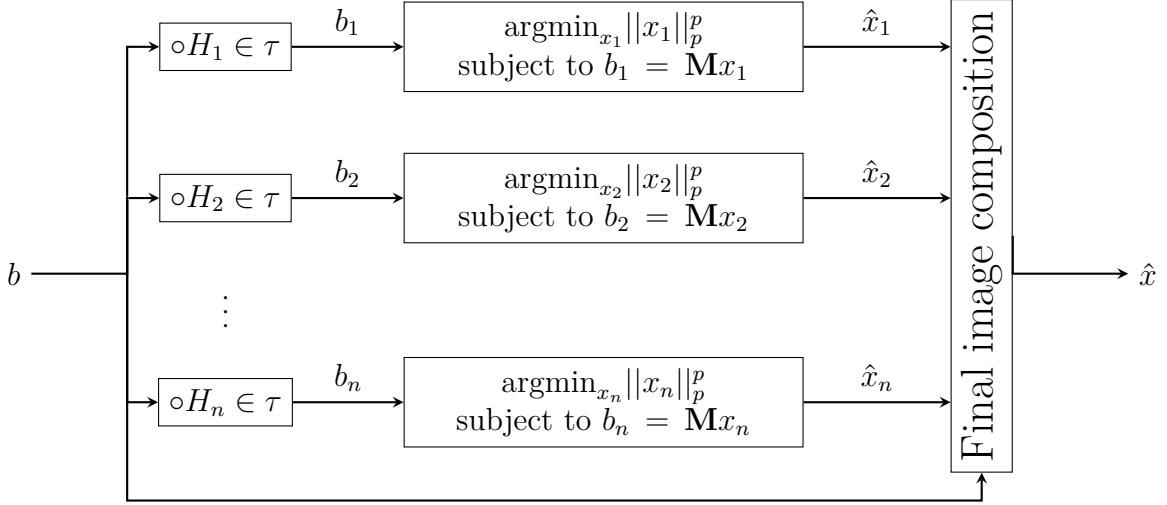


Figure 3.1: Reconstruction using pre-filtering procedure for any filter bank.

The filters used in the pre-filtering are usually a set of complementary band-pass filters with different bands. In this thesis, we refer as a set of filters with this property as ‘filter bank’. In the following sections, I present the filter banks designed for this work. First, I present the filter design based on the windowing method, which I refer as **WIN** and, next, I present the filter design based on the wavelet decomposition, which is referred as **WAV**.

3.1.1 Filters banks based on windowing method (WIN)

High frequency components of natural images are generally sparse in the image domain. I use this information to guide the filter design, preserving the high frequency components and dividing it among several bands. The goal is to design filters that increase the sparsity of the images. First, I designed a set of filters banks using the windowing method, using a Hamming window [13]. In this work, I refer to them as WIN.

First, I design n 1D WIN filters, for which I divide the spectrum in equally spaced bands. For each of the n spectrum bands, only one is selected as a passing band for each filter. More specifically, for 0 to $n-1$ filters, the selected band for the i -th filter is $[i\pi/n, (i+1)\pi/n]$. Next, I create the 2D FIR filters by combining the designed 1D WIN filters in the vertical and horizontal axis, generating n^2 2D filters. Finally, I discard the low-pass band (in the two dimensions) filters (only $n^2 - 1$ filters remain), because of the hypothesis that the image filtered with a low-pass component is not sparse for a proper CS reconstruction. Figure 3.2 shows an example of a WIN filter bank for $n = 3$, showing the spectrum magnitude of each filter. Notice that the spectrum bands have complementary magnitudes.

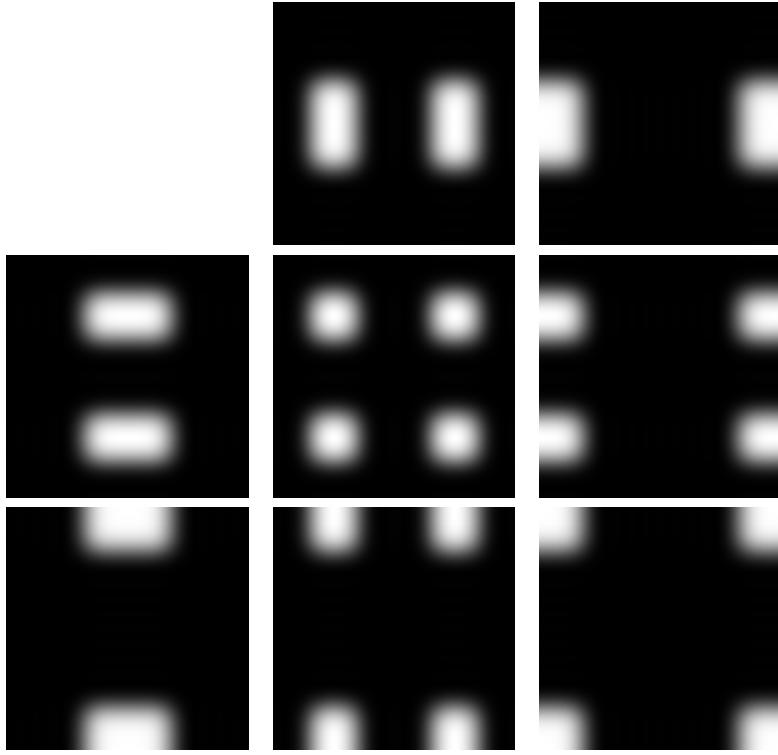


Figure 3.2: Example: The magnitude of a WIN filter bank formed by 8 different 2D filters, with different band divisions, with order 6.

The filter design allows the variation of the order of the 1D filters, which can be used to check the effect of the filter order on the quality of the image reconstructions. Filters designed with less coefficients have a smoother response decay, which could disfavor the sparsity of the filtered images. Meanwhile, filters with more coefficients usually present more rapid transitions, which may cause ringing artifacts due to the uncertainty principle [2]. In this work, I refer to a WIN filter bank of order o and n band divisions as $\text{WIN}(o, n)$.

3.1.2 Wavelet decomposition filters banks (WAV)

Other class of filter banks that are tested in this work are based on the Wavelet [79] decomposition filters (referred as WAV filter banks). I design filters that simulate the process of a 1D wavelet decomposition and, then, combine the 1D filters to form 2D filters. The pre-filtering method requires using filters, therefore, I rewrite the wavelet decomposition levels as a single filtering step by using the Z-transform decimation property [57].

Let $H_0(z)$ be a 1D low-pass filter and $H_1(z)$ the high-pass. In a wavelet decomposition, as depicted in Figure 3.3a, the signal is filtered by $H_0(z)$ and $H_1(z)$ and then it is decimated by 2 (keeping only the even samples). I use the Z-transform decimation property to express the whole wavelet decomposition into a single filtering step. This property states that if

the Z-transform of a signal x is $X(z)$, the Z-transform of this signal decimated by 2 is $X(z^2)$, as shown in Figure 3.3b. When applied to a cascade of decimation and filtering operations, the decimation operation can be swapped with the $H(z)$ filtering operation, if $H(z)$ is substituted by $H(z^2)$. This process is illustrated in Figure 3.3b.

By repeating iteratively the decimation swaps, the wavelet decomposition process can be represented by a single filtering step followed by a single higher order decimation, as seen in Figure 3.3c. In this work, I implement the $H(z^{2^k})$ by introducing zeros between samples of the coefficients of $H(z)$ in the space domain (using the Z-transform time expansion property [57]). Notice that the number of coefficients grows exponentially with number of levels.

The 1D filter bank is obtained by the filtering steps of the described wavelet decomposition (Figure 3.3d). The 2D filter bank is generated by composing vertical and horizontal implementations of the 1D filter bank. Finally, the low-pass filter is disregarded.

Figure 3.4 shows an example of the spectrum of a 2D wavelet filter bank. The filter bank corresponding to a wavelet family fam and with l number of levels is referred as $\text{WAV}(fam, l)$.

3.2 Experimental methodology

Using a database composed by MRI images, I simulate the process of reconstructing the images using each filter bank. First, I simulate the process of taking measurements from an MRI generated image. In this work, I choose a rectangular grid approximation of the radial trajectory, varying the number of radial lines taken. The reconstruction is performed using pre-filtering reconstruction model, as illustrated in Figure 3.1. This is the basic procedure of our experimental methodology, which is repeated for each image, filter bank, and for different numbers of radial lines.

3.2.1 Simulations settings

In the simulation tests, I use 55 images generated from T2 weighted brain images, from 11 axial slices of the brain of 5 participants. These images were taken from the BIRN database [80], which were generated by an MRI machine. For a faster reconstruction, I scaled these images to 256×256 pixels. Figure 3.5 shows a small subset of these images, which are slices of different positions of a brain scan.

The sets of filter banks used in the simulation include the state-of-the-art TV and Haar filters (described in Section 2.3), the WIN filter banks (Section 3.1.1), and the Wavelet filter banks (Section 3.1.2). Due to the high reconstruction times and the

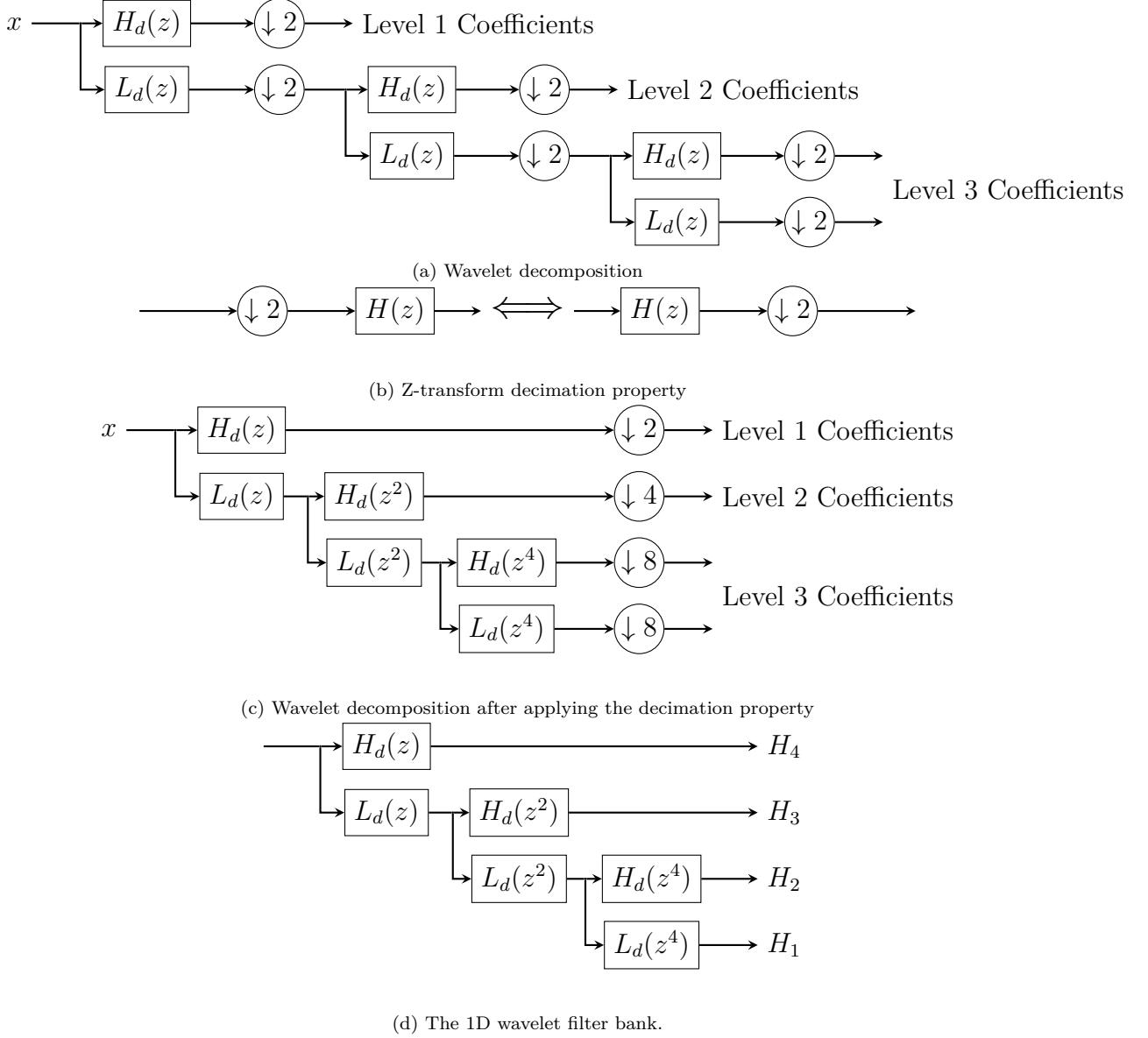


Figure 3.3: Wavelet decomposition simplification to the Wavelet filter bank.

massive generated data, I restricted the number of filter banks to the most representative configurations. I choose to use the WIN filters with the following configurations: 2 to 5 band divisions and orders 2 to 10. Preliminary test showed that using filters with more band divisions and higher order did not affect the results considerably. The Wavelet filters configurations considered are: Daubechies 1 and 4, Coiflets 1, Symlets 2, and reverse biorthogonal 2.2, 2.8 and 6.8 [81], some of the wavelet families implemented in MATLAB. The number of levels are from 1 to 3, considering that using more levels created very long filters. I choose 20, 40, 60, 80, and radial 100 lines for simulating the radial trajectory sampling. The chosen numbers of radial lines are fewer smaller than what is commonly used in an actual MRI machine, leading to a faster

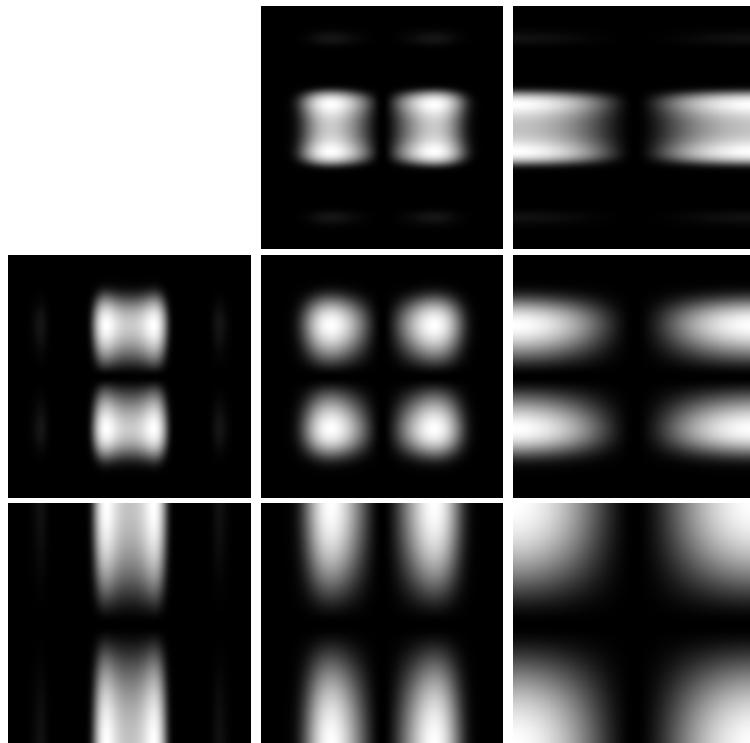


Figure 3.4: Example: Magnitude of 2 level Biorthogonal type 2.8 wavelet filters. Only the low-pass filter is not considered.

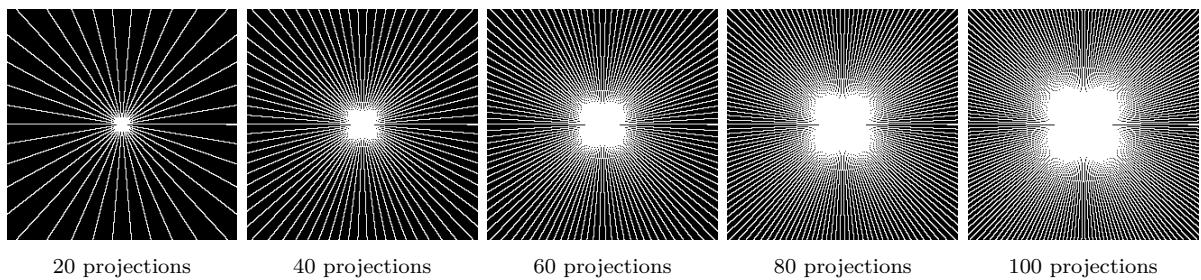


Figure 3.5: Sampling positions for the reconstruction simulations, ranging from 20 to 100 radial lines projections.

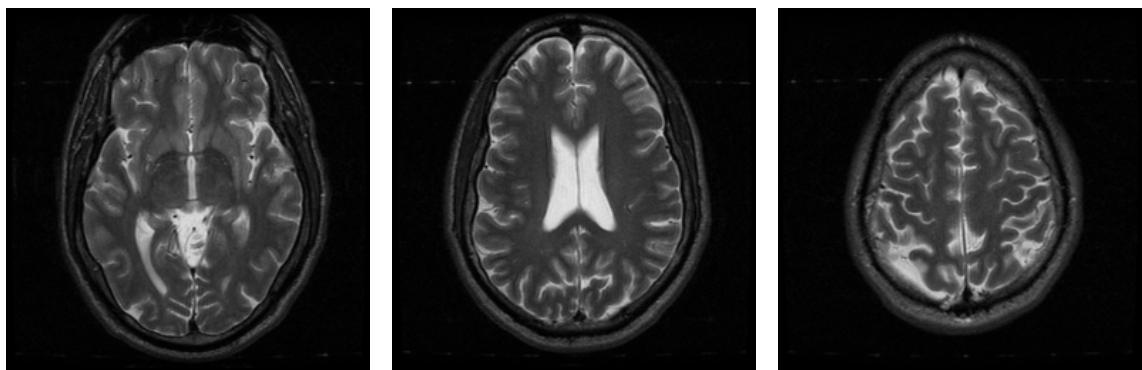


Figure 3.6: Some slices from the first participant brain, taken from an MRI machine.

Table 3.1: Pre-filtering total reconstruction time (s) for TV filter.

sampling	Value of p in the ℓ_p metric									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
20 lines	1786.3	1939.8	1552.9	1523.4	1999.9	1479.3	221.7	101.3	52.9	23.7
40 lines	3372.6	3046.5	2865.1	2935.1	3168.9	1217.7	221.4	105.4	54.2	24.7
60 lines	4808.5	4761.1	4170.5	4045.8	4350.1	740.2	235.7	114.3	54.7	25.5
80 lines	6815.5	6053.4	5959.7	5150.3	6458.8	1001.4	280.4	116.6	58.1	28.5
100 lines	8026.7	8133.2	7998.4	6597.8	9592.5	2017.8	323.7	135.4	61.8	28.0

reconstruction time than what is possible real MRI systems.

The reconstruction step is to solve the minimization problem given by:

$$\begin{aligned} & \operatorname{argmin}_{x_n} \|x_n\|_p^p \\ & \text{subject to } b_n = \mathbf{M}x_n. \end{aligned} \quad (3.2)$$

The same IRLS algorithm used by Miosso *et al.* [12] is used here. However, the value of p used in the original work was 0.1, which requires a long time to reconstruct the images. Table 3.1 shows the total reconstruction time for one image, using pre-filtering with TV filters, for different values of p (from 0.1 to 1). To test several filter banks in a feasible amount of time, the value of $p = 1$ was selected.

The simulation was performed for each of the input parameters, 5 different sampling schemes, 55 images, 43 filter banks composed of 437 filters, which result in a total of 120,175 reconstructions. The experiment was ran on a computer cluster in which the reconstructions were distributed among several nodes. Each node of the cluster has two Intel Xeon E5430 2.7 GHz CPU and the implementation is performed in MATLAB, based on IRLS implementation of the original pre-filtering work [12].

3.2.2 Analysis Parameters

To analyze the reconstruction quality, two full reference objective quality metrics are used. The first is the Signal-to-Noise Ratio (SNR) [82], which gives measures in decibels (dB). The SNR is calculated between the ground truth image x and the reconstructed image y , using the following equation:

$$\text{SNR}(dB) = 10 \log_{10} \left(\frac{\sum_{i,j} x(i,j)^2}{\sum_{i,j} (x(i,j) - y(i,j))^2} \right). \quad (3.3)$$

The range of SNR values normally varies between zero and infinity. The higher the SNR value, the lower the error between the reconstructed and reference image and the quality of the reconstructed image is higher.

The other image quality metric is the Structural SIMilarity index (SSIM) [82]. SSIM uses the fact that natural images pixels are highly correlated to the nearby pixels. This dependency carries important information about the structures that the human visual system is trained to identify [83, 84]. Given a reference image x and the reconstructed image y , SSIM estimates the luminance $l(x, y)$, the contrast $c(x, y)$, and the structure $s(x, y)$ between two images using the following equations:

$$\begin{aligned} l(x, y) &= \left(\frac{2\mu_x\mu_y+C_1}{\mu_x^2\mu_y^2+C_1} \right), \\ c(x, y) &= \left(\frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2\sigma_y^2+C_2} \right), \\ \text{and } s(x, y) &= \left(\frac{\sigma_{xy}\mu_y+C_3}{\sigma_x^2\sigma_y^2+C_3} \right), \end{aligned} \quad (3.4)$$

where μ_x and μ_y are the local means of x and y ; σ_x and σ_y are the standard deviations of x and y ; σ_{xy} is the cross-covariance between x and y ; and C_1 , C_2 and C_3 are small constants that are meant to keep the denominators different from zero. The SSIM index is obtained by combining l , c , and s , using the following equation:

$$\text{SSIM}(x, y) = l(x, y)^\alpha c(x, y)^\beta s(x, y)^\gamma. \quad (3.5)$$

In this thesis, the most common implementation of SSIM index metric is used, which is the popular Matlab implementation provided by the authors [82], with $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$. SSIM generates as output a value between ‘0’ and ‘1’, with higher values corresponding to a better quality. In other words, since SSIM has maximum and minimum values, it can be more useful in the analysis than SNR.

Other features I want to analyze and correlate to the image quality are related to the filter banks. One of them is the sparsity of the filtered images. The used sparsity metric was calculated by taking an average of the ℓ_1 value of all the filtered versions of the original image. In other words, the ℓ_1 based sparsity metric for the filter bank H_1 to H_n and the image x can be defined as:

$$\sum_{k=1}^n \sum_{i=1}^r \sum_{j=1}^c |x_k(i, j)| / (n), \quad (3.6)$$

where x_k is the $r \times c$ filtered version of x , when filtered by the k -th filter H_k ; r is the number of rows and c is the number of columns of the images. Finally, I calculate the metric for every image in the database and take the average value.

The other feature of the filter banks I considered is the spectrum coverage. I defined the spectrum coverage of a filter bank as the union of the passing band of all filters of the filter bank. In order to separate the regions of the Fourier spectrum between pass and stop bands, I use Otsu’s threshold technique [85] on the amplitude of the frequency

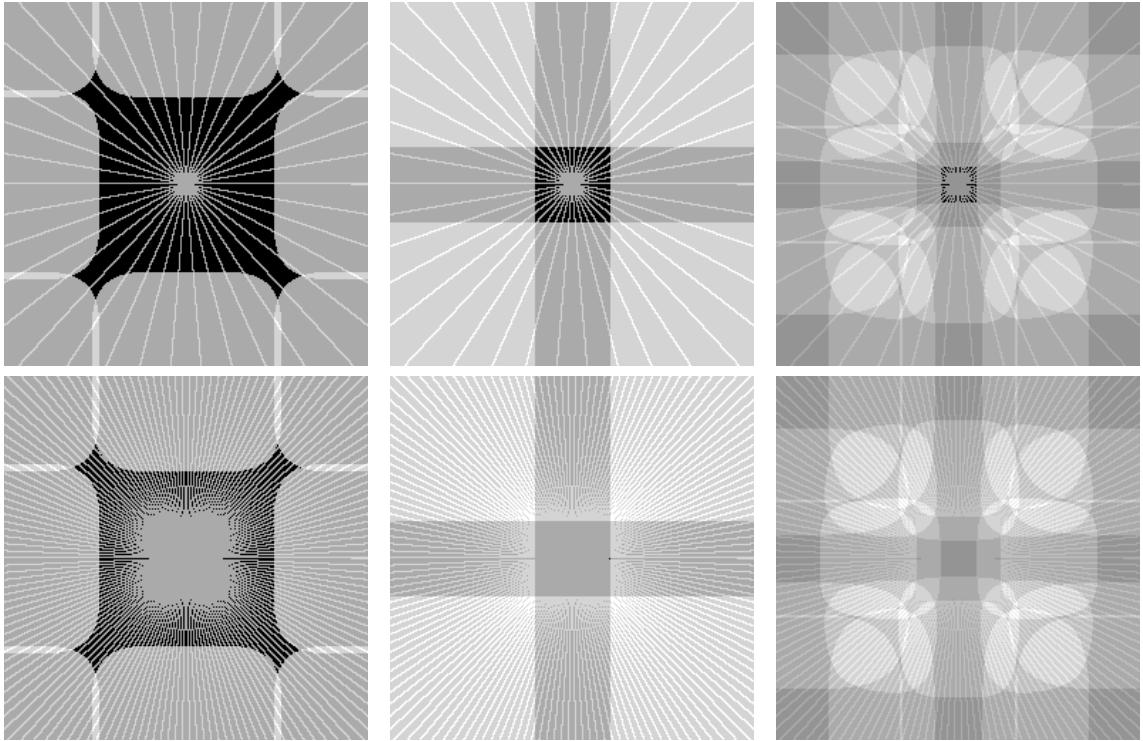


Figure 3.7: Spectrum coverage for 3 different filter configurations. First column is WIN filter bank design with 12 coefficients and 2x2 band divisions, second is TV filter bank and third column is Daubechies’s 3 wavelet filter bank with 2 levels. First row is using 20 radial lines, while second is using 80 lines.

spectrum of each filter. Then, I apply the ‘or’ morphological operation to the thresholded spectrum of all filters for obtaining the total coverage of this filter bank (the process works as a union of the passing bands regions). Optionally, I also complement the spectrum coverage with the position of where radial lines were sampled. Figure 3.7 depicts the spectrum coverage for different filter banks. The black values represent the uncovered spectrum.

3.3 Results

The average SNR and SSIM values for the reconstructions obtained in all simulations are presented, respectively, in Figures 3.8 and 3.9. The figures show some horizontal lines that correspond to the SNR and SSIM corresponding to the NESTA TV minimization (Section 2.4.3). Table 3.2 shows the average SNR and SSIM values, but also shows the number of filters in the filter banks, the average ℓ_1 of the filter banks and the spectrum coverage.

In our results, the TV and Haar filter banks produce lower image quality than NESTA TV. However, Miosso’s work [12] produced a higher quality than what is produced by the

ℓ_1 -MAGIC TV. The two possible explanation is or due to the different value of p or is due to the reconstruction algorithm.

Some of the proposed filter banks present statistically significant (with $p=0.05$) higher image quality than the NESTA TV. Figures 3.8 and 3.9 and Table 3.2 show that, for 20 radial lines WIN filters with order 2, 4 and 6 produce a significantly higher quality (around 0.1 SSIM and 2.5 dB SNR difference) than what is obtained with NESTA TV. However, for 40 or more radial lines, only WIN filters with order 2 show a systematically higher image quality than NESTA TV. The best performing wavelet filters performed slightly better than the Haar filters.

A possible explanation for these results is the spectrum of the individual filters of the filter banks. Filters with low order have a smooth spectrum decay response and can cover large areas of the spectrum, while high order filters have a sharp spectrum decay response. A sharp filter could have a small spectrum, depending on the number of band divisions and, therefore, the filtered image would be sparse in both space and frequency domains, leading to a poor CS reconstruction.

Figure 3.10 shows the pre-filtering reconstructions images for a single image, for several filter banks and different numbers of radial lines. Notice that filter banks WIN(2,2) and WIN(2,3) present a higher quality than the other filters, including the state-of-the-art Haar filter. In fact, WIN(2,2) presents images with visually less noise, and WIN(2,3) presents images with more details. The filter bank WIN(4,2) shows high quality results for 20 radial lines, however, the quality does not increase significantly when increasing the number of radial lines.

Some wavelet filter banks also produce higher image quality than what is obtained with Haar filters, specially for a higher number of radial lines. Examples are WAV(coif1,2) (that shows less noise than Haar) and WAV(coif1,3) (that shows more details than Haar). Albeit these filter banks do not produce results as good as what is obtained with WIN(2,2) and WIN(2,3). However, there are also wavelet filter banks that produce low quality results, such as WAV(db4,1).

Figure 3.11 shows the relation between the ℓ_1 -based sparsity metric and the SNR of the image reconstructions for all tested filter banks, separated by the number of radial lines. SNR is used, instead of SSIM, because SNR is easier to visualize. Higher values of sparsity corresponds to the lower values of ℓ_1 in the x axis of Figure 3.11. Each point in the plot corresponds to the average SNR and sparsity values for all images, corresponding to a single filter bank.

Some of the points with the highest ℓ_1 values correspond to reconstruction quality values (SNR) below 10 dB. These points correspond to the filter banks WAV(rbio6.8,1), WAV(rbio6.8,2), WAV(rbio2.8,1), WAV(rbio2.8,2), WAV(db4,1), WIN(8,2) and

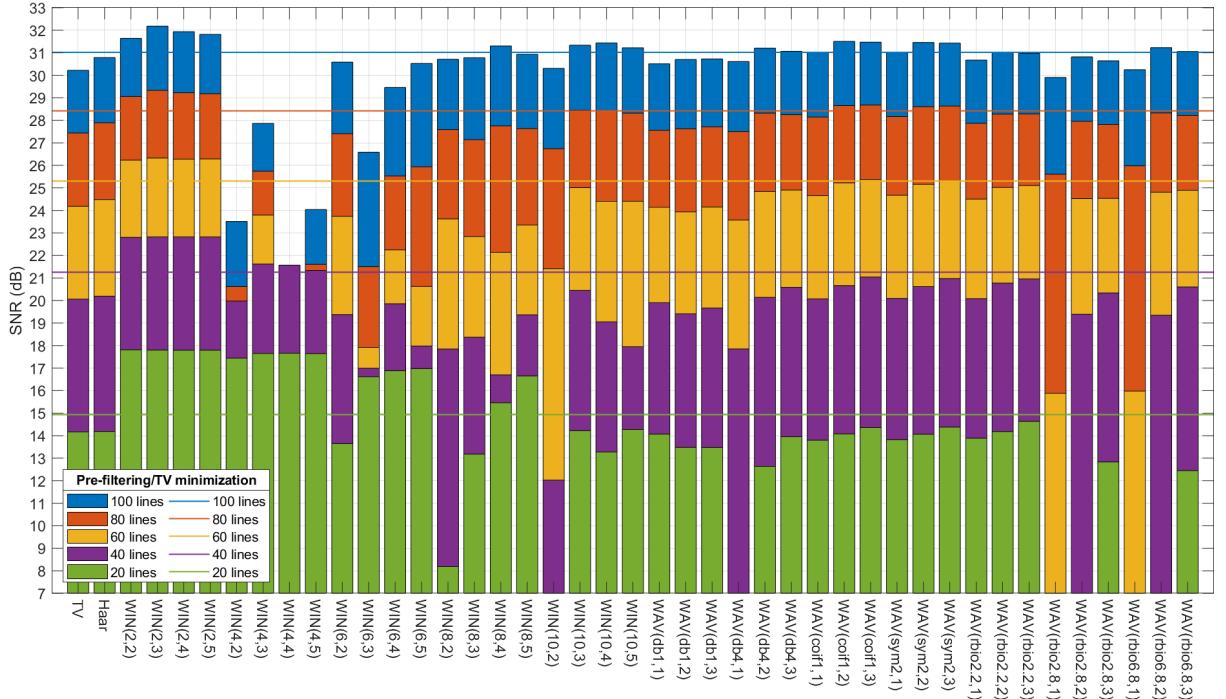


Figure 3.8: Average SNR (dB) of the reconstruction using the pre-filtering with IRLS with $p=1$ for the filter banks tested. It is also shown SNR of reconstruction with the NESTA TV minimization as horizontal lines.

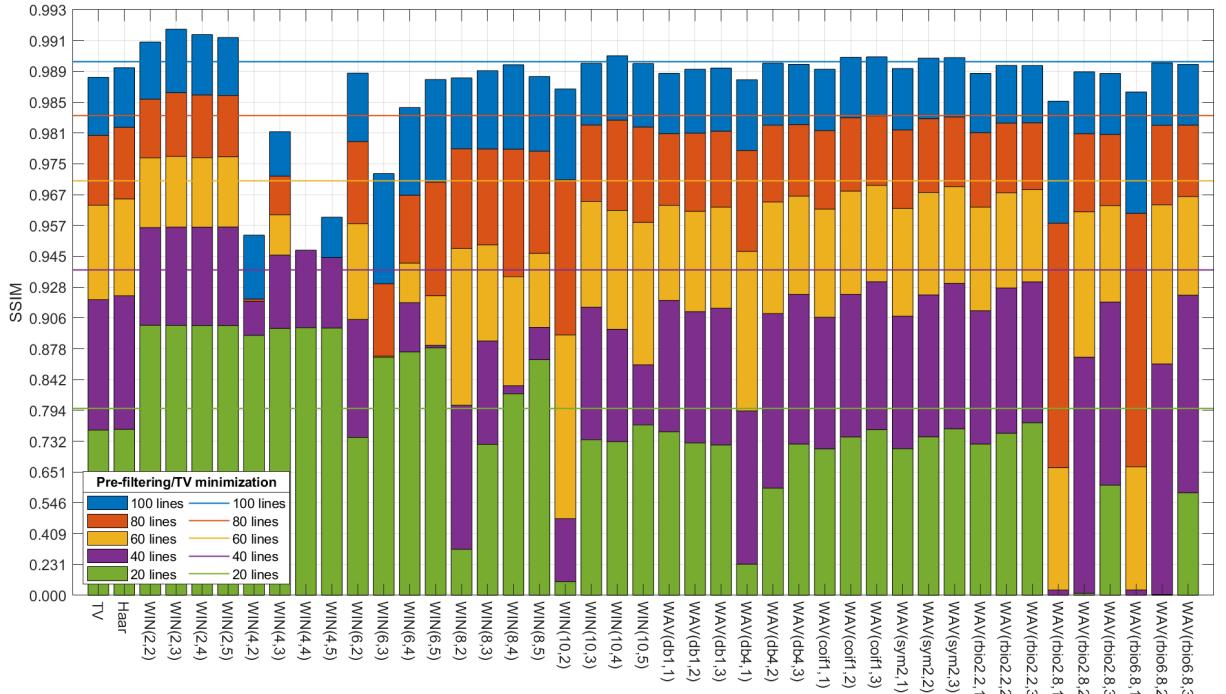


Figure 3.9: Average SSIM of the reconstruction using the pre-filtering with IRLS with $p=1$ for the filter banks tested. The y axis is in a log scale. It is also shown the SSIM of reconstruction with the NESTA TV minimization as horizontal lines.

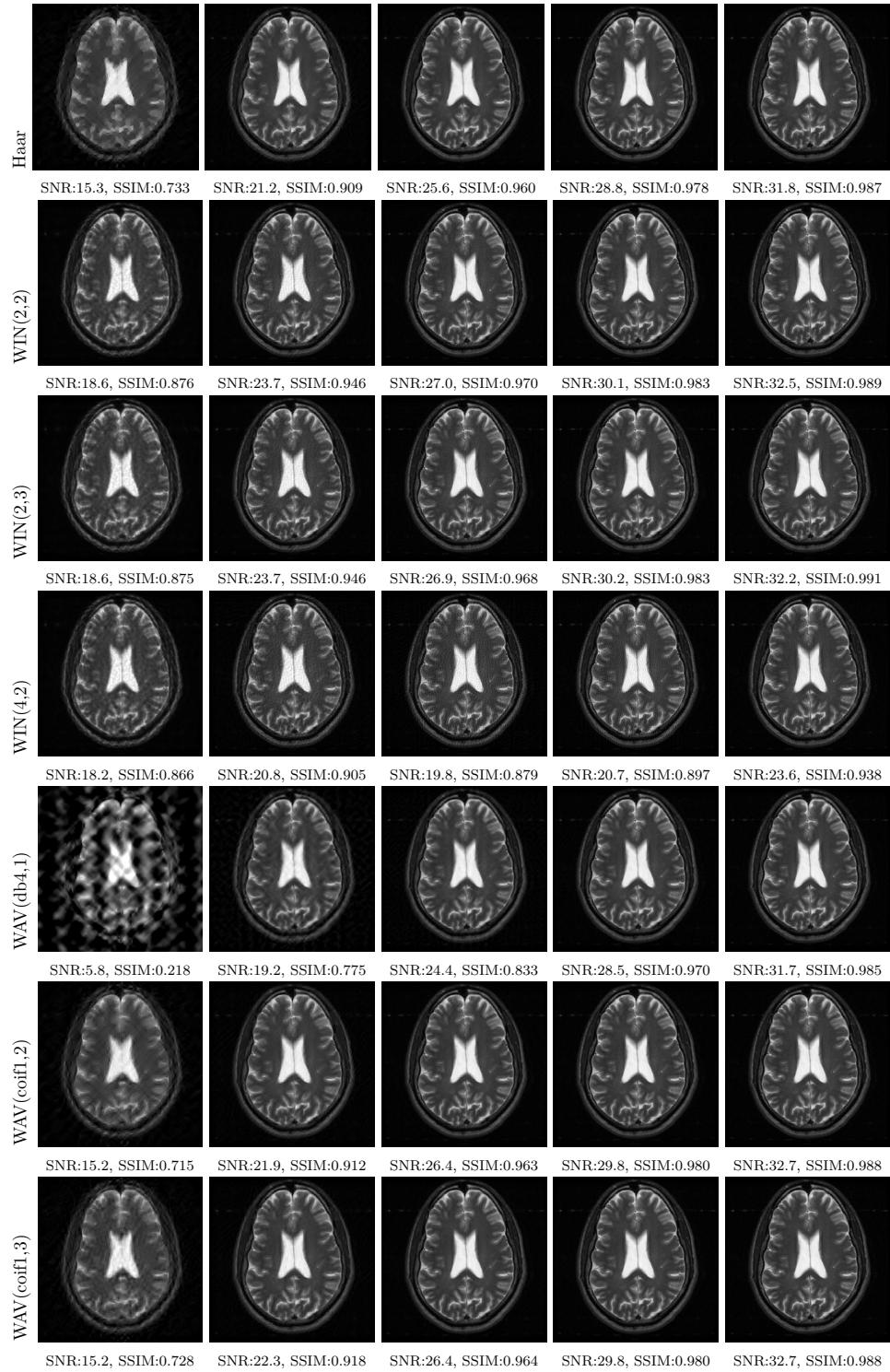


Figure 3.10: MRI reconstructions with pre-filtering. Columns corresponds to respectively 20, 40, 60, 80 and 100 radial lines.

Table 3.2: Average reconstruction SNR(dB), SSIM, and time using pre-filtering with IRLS reconstruction algorithm, classified by radial lines. It is also shown the number of filters filter bank, the average ℓ_1 of the images and the spectrum coverage. The bolded face cells correspond to the filter banks with the highest SNR or SSIM.

	20 radial lines			40 radial lines			60 radial lines			80 radial lines			100 radial lines			num fils	avg ℓ_1	spec. cov.
	SNR	SSIM	time	SNR	SSIM	time												
TV	14.2	0.756	19.2	20.1	0.920	21.8	24.2	0.964	24.1	27.4	0.980	25.9	30.2	0.988	27.4	2	31.6	95.7
Haar	14.2	0.757	27.7	20.2	0.922	32.4	24.5	0.966	36.3	27.9	0.982	39.0	30.8	0.989	41.8	3	32.4	99.0
WIN(2,2)	17.8	0.900	24.9	22.8	0.957	35.1	26.2	0.976	37.3	29.1	0.986	33.9	31.6	0.991	34.1	3	78.2	90.9
WIN(2,3)	17.8	0.900	68.4	22.8	0.957	97.2	26.3	0.976	94.9	29.3	0.986	94.1	32.2	0.992	93.1	8	79.3	100.0
WIN(2,4)	17.8	0.900	128.5	22.8	0.957	179.4	26.3	0.976	178.1	29.2	0.986	176.6	31.9	0.992	177.5	15	79.8	100.0
WIN(2,5)	17.8	0.900	207.1	22.8	0.957	288.6	26.3	0.976	289.1	29.2	0.986	280.8	31.8	0.991	282.6	24	80.1	100.0
WIN(4,2)	17.4	0.891	25.9	110.0	0.919	32.2	19.7	0.904	34.9	20.6	0.920	37.3	23.5	0.954	39.0	3	64.6	78.6
WIN(4,3)	17.7	0.898	69.0	21.6	0.945	86.9	23.8	0.961	91.9	25.7	0.972	97.2	27.9	0.981	104.4	8	79.9	100.0
WIN(4,4)	17.7	0.898	130.5	21.6	0.947	165.2	21.6	0.939	177.1	20.8	0.922	187.9	21.5	0.932	200.7	15	84.0	100.0
WIN(4,5)	17.6	0.898	208.1	21.3	0.944	263.0	21.3	0.936	282.2	21.6	0.938	298.1	24.0	0.960	320.8	24	85.5	100.0
WIN(6,2)	13.6	0.740	25.9	19.4	0.905	30.2	23.7	0.958	34.3	27.4	0.979	37.1	30.6	0.988	39.8	3	35.6	75.7
WIN(6,3)	16.6	0.869	68.3	17.0	0.849	83.0	17.9	0.870	91.9	21.5	0.930	100.2	26.6	0.973	106.3	8	66.3	99.1
WIN(6,4)	16.9	0.875	127.5	19.9	0.918	153.5	22.2	0.941	170.8	25.5	0.967	185.3	29.5	0.984	200.5	15	79.7	100.0
WIN(6,5)	17.0	0.879	204.9	18.0	0.882	243.5	20.6	0.922	270.1	25.9	0.971	293.1	30.5	0.988	316.5	24	85.5	100.0
WIN(8,2)	8.2	0.325	25.7	17.8	0.803	30.9	23.6	0.948	35.0	27.6	0.978	37.6	30.7	0.988	40.0	3	32.6	74.8
WIN(8,3)	13.2	0.725	67.3	18.4	0.886	79.8	22.8	0.950	90.1	27.1	0.978	98.5	30.8	0.989	105.2	8	42.6	95.4
WIN(8,4)	15.5	0.821	128.2	16.7	0.833	152.9	22.1	0.934	171.6	27.8	0.978	186.3	31.3	0.989	198.5	15	66.2	100.0
WIN(8,5)	16.7	0.866	200.9	19.4	0.898	241.3	23.4	0.946	271.9	27.6	0.977	296.4	30.9	0.988	315.3	24	79.5	100.0
WIN(10,2)	3.7	0.111	25.6	12.0	0.480	30.4	21.4	0.892	34.8	26.7	0.971	37.5	30.3	0.987	39.9	3	33.3	74.5
WIN(10,3)	14.2	0.735	67.0	20.5	0.915	80.4	25.0	0.965	90.8	28.4	0.982	98.5	31.3	0.989	104.5	8	33.0	93.6
WIN(10,4)	13.3	0.731	125.7	19.1	0.897	147.8	24.4	0.963	167.3	28.4	0.983	182.1	31.4	0.990	195.4	15	47.9	98.2
WIN(10,5)	14.3	0.767	203.9	18.0	0.860	243.1	24.4	0.959	273.5	28.3	0.982	295.2	31.2	0.989	314.7	24	67.0	100.0
WAV(db1,1)	14.1	0.753	27.4	19.9	0.919	32.5	24.1	0.964	36.0	27.6	0.981	38.7	30.5	0.988	41.1	3	33.0	98.3
WAV(db1,2)	13.5	0.728	75.2	19.4	0.911	89.2	23.9	0.962	99.4	27.6	0.981	106.4	30.7	0.989	113.6	8	39.1	99.9
WAV(db1,3)	13.5	0.723	144.3	19.7	0.914	170.6	24.2	0.964	190.8	27.7	0.981	205.7	30.7	0.989	221.0	15	50.9	100.0
WAV(db4,1)	5.1	0.233	27.4	17.8	0.793	32.6	23.6	0.947	36.6	27.5	0.978	38.8	30.6	0.988	41.5	3	32.1	91.4
WAV(db4,2)	12.6	0.600	75.5	20.1	0.910	88.7	24.8	0.965	99.4	28.3	0.982	105.8	31.2	0.989	111.8	8	38.6	98.5
WAV(db4,3)	14.0	0.725	143.5	20.6	0.923	169.4	24.9	0.967	188.5	28.2	0.982	201.1	31.1	0.989	213.2	15	51.8	99.7
WAV(coif1,1)	13.8	0.713	27.1	20.1	0.907	32.5	24.7	0.963	36.3	28.2	0.981	38.6	31.0	0.989	41.6	3	31.6	95.0
WAV(coif1,2)	14.1	0.741	74.9	20.7	0.923	88.9	25.2	0.968	100.1	28.7	0.983	106.4	31.5	0.990	113.3	8	37.6	99.4
WAV(coif1,3)	14.4	0.757	143.2	21.0	0.931	170.8	25.4	0.970	189.9	28.7	0.983	204.3	31.5	0.990	218.1	15	50.7	99.9
WAV(sym2,1)	13.8	0.714	27.0	20.1	0.908	32.5	24.7	0.963	36.3	28.2	0.981	38.7	31.0	0.989	41.6	3	31.7	95.3
WAV(sym2,2)	14.1	0.742	75.1	20.6	0.923	88.7	25.2	0.968	99.9	28.6	0.983	106.1	31.5	0.990	113.0	8	37.9	99.4
WAV(sym2,3)	14.4	0.759	143.3	21.0	0.930	170.0	25.3	0.969	189.8	28.6	0.983	203.1	31.4	0.990	216.7	15	50.9	99.9
WAV(rbio2,2,1)	13.9	0.725	28.1	20.1	0.912	33.1	24.5	0.964	37.4	27.9	0.981	39.8	30.7	0.988	42.4	3	32.7	97.2
WAV(rbio2,2,2)	14.2	0.749	76.3	20.8	0.928	90.6	25.0	0.968	101.0	28.3	0.982	108.6	31.0	0.989	114.8	8	42.0	99.7
WAV(rbio2,2,3)	14.6	0.771	145.4	21.0	0.931	172.9	25.1	0.969	193.3	28.3	0.982	207.3	31.0	0.989	220.6	15	57.1	100.0
WAV(rbio2,8,1)	0.0	0.000	27.9	1.2	0.044	33.1	15.9	0.663	37.4	25.6	0.958	39.6	29.9	0.985	42.0	3	33.3	88.3
WAV(rbio2,8,2)	0.4	0.016	75.5	19.4	0.869	89.1	24.5	0.962	98.2	28.0	0.981	104.0	30.8	0.989	109.1	8	40.7	97.8
WAV(rbio2,8,3)	12.8	0.610	137.9	20.3	0.918	163.1	24.5	0.964	181.2	27.8	0.980	191.8	30.6	0.988	202.7	15	55.3	99.6
WAV(rbio6,8,1)	0.0	0.000	25.8	1.2	0.045	30.4	16.0	0.667	34.1	26.0	0.962	36.6	30.2	0.986	39.1	3	32.6	88.9
WAV(rbio6,8,2)	0.2	0.008	70.4	19.3	0.861	82.9	24.8	0.964	92.7	28.3	0.982	98.7	31.2	0.989	103.8	8	38.7	97.9
WAV(rbio6,8,3)	12.5	0.583	129.7	20.6	0.923	153.3	24.9	0.967	169.1	28.2	0.982	180.4	31.1	0.989	189.7	15	52.7	99.6

WIN(10,2). Specially for 20 radial lines, we can see two different patterns. Below an average ℓ_1 value of 60, the SNR is always below 15 dB (for 20 radial lines). Above an average ℓ_1 value of 60, all points (except for one) are above 15 dB, *i.e.* less sparsity corresponds to higher image quality for 20 radial lines. These results suggests that the ℓ_1 -based sparsity metric does not always lead to an increased reconstruction quality.

Other important factor to consider is the spectrum coverage. Figure 3.12 shows the relation between the spectrum coverage and the reconstruction quality (in SNR) for each filter bank and each measurement scheme. In the first plot, the radial lines are not included as part of the spectrum covered (these values are represented in the last column of Table 3.2). Notice that the points for different radial lines are vertically aligned, showing the same spectrum coverage. We can observe points with poor reconstruction with either a

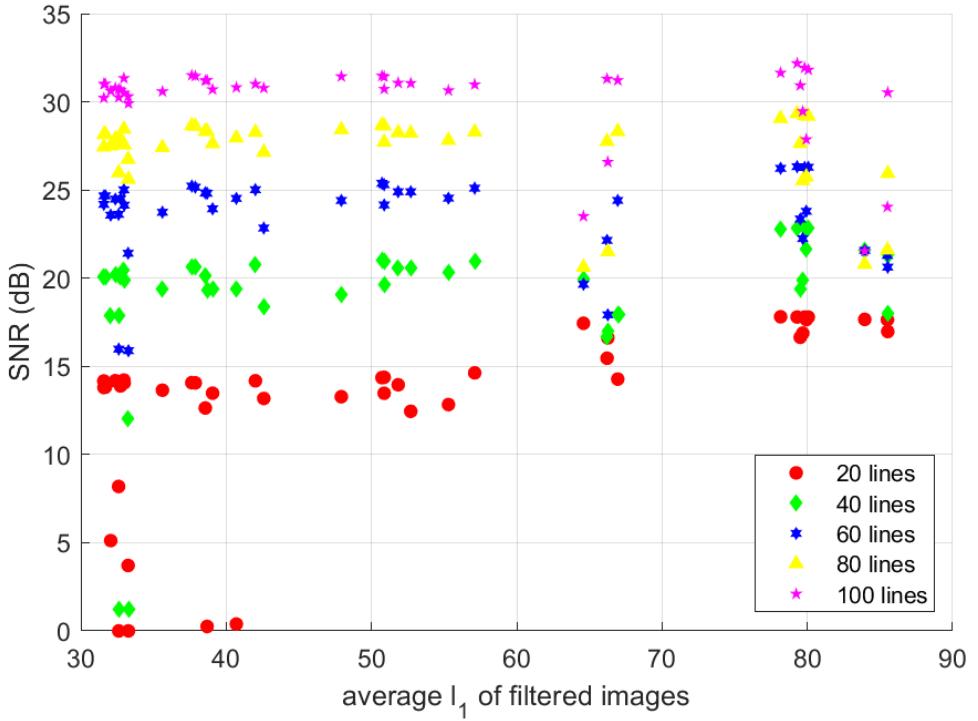


Figure 3.11: Average SNR(dB) vs average ℓ_1 of the filtered images. Each point corresponds to a filter bank.

high or a low spectrum coverage. Also, the highest SNRs corresponds to 90.9% and 100% spectrum coverage.

Let us look at the points corresponding to filters banks with the worst reconstruction quality for 20 radial lines: WIN(8,2) and WIN(10,2), WAV(dB4,1), WAV(rbio6.8,1), WAV(rbio6.8,2), WAV(rbio2.8,1), WAV(rbio2.8,2). WIN(8,2) and WIN(10,2) have only 74.8% and 74.5% spectrum coverage, what is far below the average and could explain the poor reconstruction. WAV(dB4,1), WAV(rbio2.8,1), and WAV(rbio6.8,1) have between 88.3% and 91.4% spectrum coverage, what is still below the average. However, WAV(rbio2.8,2) and WAV(rbio6.8,2) have 97.8% and 97.9% of the spectrum coverage, and there are plenty of filter banks with worst spectrum coverage and higher reconstruction quality. In summary, spectrum coverage can explain some of the results, but not all of them.

The second plot of Figure 3.12 shows the relation between SNR and the spectrum coverage percentage, including the measurements. Since the measurements are concentrated in the lower frequencies of the spectrum, they help to improve the spectrum coverage, specially for a higher number of radial lines. Notice points following an increasing curve in the plot, which means that higher spectrum coverage (including the measurements) leads to better reconstruction quality. However, we also observe points (remember, each point represents a filter bank) with full or almost full spectrum

coverage that are not among the best SNR values. Although the data suggests that spectrum coverage is a good indicative of a good reconstruction, the result also depends on the amount of measurements.

The average reconstruction time is shown in Table 3.2. The fastest reconstruction time was for the TV filter banks, because this is the smallest filter bank. The reconstruction time is largely determined by the number of filters in the bank, because the number of filters is equal to the number of individual reconstructions needed. Figure 3.13 shows a boxplot of the average reconstruction times per-image for all filters, grouped by the number of filters in the bank and by the number of radial lines. We observe that the total reconstruction time is indeed linearly correlated to the number of filters in the filter bank. Also, for the chosen experiment settings, although the number of measurements also increased the reconstruction time, it has a smaller impact on the reconstruction time than the number of filters.

Notice that the filter bank $\text{WIN}(2,2)$ has only 3 filters, resulting in a fast reconstruction time. This filter bank also produces the overall best image quality for 20 radial lines, and very close to the best quality for a higher number of radial lines. To obtain a balance between reconstruction time and reconstruction quality, $\text{WIN}(2,2)$ seems to be the best option. The filter bank $\text{WIN}(2,3)$ presents the best reconstruction quality for 40, 60, 80 and 100 radial lines. However, it takes around 8/3 more time to reconstruct it than with $\text{WIN}(2,2)$.

3.4 Conclusions

In this chapter I proposed a systematic evaluation of the pre-filtering method with a simulation of MRI reconstruction using several filter banks. I presented the model, described the filter banks and the simulation details and the analysis parameters.

With our quantitative and qualitative analysis, I found filter banks that provide images with superior quality than the Haar filters. The filter banks designed with the windowing method with the lowest order provided images with the best quality. The improvement of quality is higher for 20 radial lines, showing an improvement of 2.8dB SNR/0.106 SSIM when compared to NESTA TV minimization and 3.6 dB SNR/0.143 SSIM when compared to pre-filtering with Haar filter bank. Some wavelet filters produced slightly better image quality than the Haar filter bank.

Results show an overall positive correlation between the image quality and the spectrum coverage, when including the measurements. The combination of the spectrum of the filters and the number of measurements leads to good results. However, when

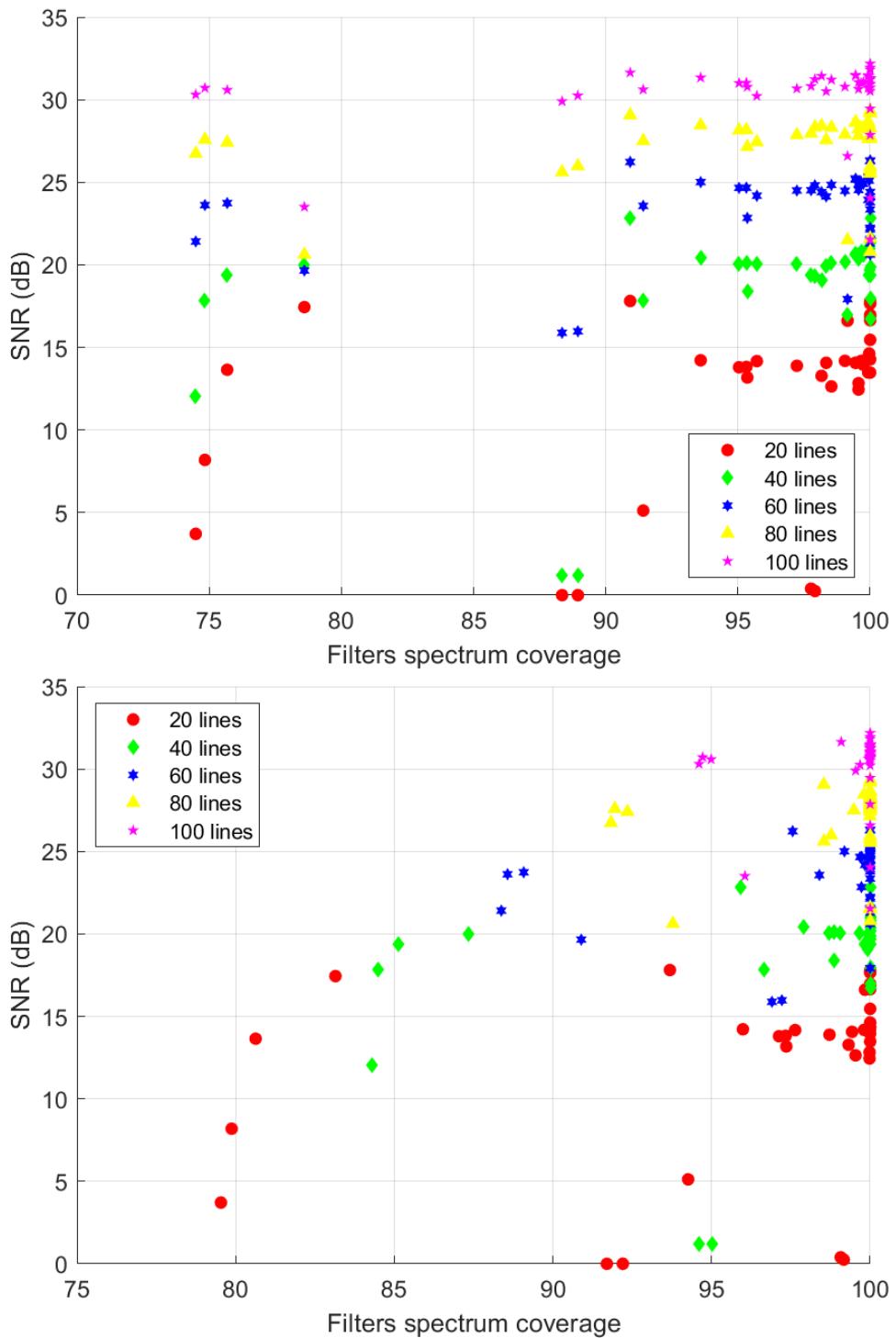


Figure 3.12: Average SNR(dB) vs spectrum coverage of the filter banks. First plot does not include the measurements in the spectrum coverage. The second includes the measurements of part of covered spectrum.

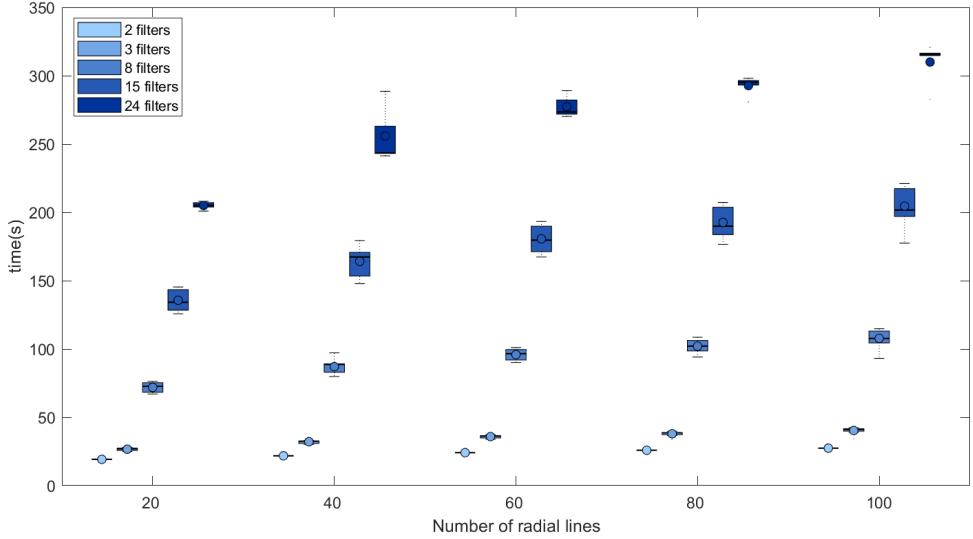


Figure 3.13: Reconstruction times (s) grouped by number of radial lines and number of filters in the filter banks.

only the spectrum of the filters is considered, the results are not conclusive. It seems that we cannot evaluate the spectrum coverage ignoring the measurements.

I hypothesized that a high sparsity would lead to a high image quality. However, the results using the ℓ_1 -based sparsity metric do not support this hypothesis. The possible explanation is that the filters that sparsify the space domain also sparsify the frequency domain, thus, frequency information is lost.

Finally, the reconstruction time is mostly influenced by the number of filters in the filter bank. Using a filter bank based on windowing method, with order 2 and 2 band divisions (WIN(2,2)) provides a good compromise between reconstruction time and reconstruction quality. This filter bank is among the filters with highest image quality, and it has only 3 filters, resulting in a fast reconstruction time.

Chapter 4

Improving pre-filtering by adding zero-valued measurements

Since the behavior of the filters is known before reconstruction, one can take advantage of this information to improve the reconstruction step of the pre-filtering method. One of the possible approaches is to include measurements to the reconstruction step, based on the spectrum of the filter banks. In this chapter, I redefine the pre-filtering method using matrices to represent some processes of the method. This allows to introduce the proposed modification to the pre-filtering method. The modified pre-filtering method uses different sampling matrices for the original measurements and the reconstruction. The process consists of including measurements with zero values in spectrum positions that have amplitude below a threshold. With this, the number of measurements in the reconstruction is increased.

Additionally, I observed that the pre-filtering method was only presented and tested in the literature using the IRLS algorithm for reconstruction. In this chapter, I also use the NESTA algorithm in the reconstruction step. I present an experiment with a setting similar as in Chapter 3 to evaluate the results of the two reconstruction algorithms, as well the results of applying the proposed modification on the prefiltering method.

4.1 Proposed pre-filtering modifications

To formally introduce the proposed modification, the original formulation of pre-filtering is redefined. Let x be a N -length signal to be reconstructed, \mathbf{F} denotes the N point discrete Fourier transform (DFT) operation. Let the m -length vector \mathbf{s} represent the indices of the sampling pattern, *i.e.* the positions of the measurements in the spectrum. The operator \mathbf{S} selects the \mathbf{s} elements of the input. \mathbf{S} can be represented by a matrix formed by a selection rows with index \mathbf{s} of the identity matrix \mathbf{I} . In a submatrix notation,

$\mathbf{S} = \mathbf{I}_{s,*}$. Note that \mathbf{S}^T , when applied to a m -length signal, generates an N -length signal with the content of the original signal at the sampling pattern, filling with zeros the non-sampled positions. The original measurements b can be represented by

$$b = \mathbf{SF}x, \quad (4.1)$$

where \mathbf{SF} is the sampling matrix (\mathbf{M} in previous chapters). As seen in previous chapters, the first step of the pre-filtering method consists of filtering the measurements with a list of filters h_k . In this chapter, this process is represented in a slightly different way than presented in Chapter 2.

Let h_k be the coefficient of the k -th filter used in the pre-filtering method. Let \mathbf{H}_k be the $N \times N$ matrix that applies the element-wise multiplication in the spectrum domain of the filter h_k , *i.e.* $\mathbf{H}_k = \text{diag}(\mathbf{F}h_k)$. To filter the measurements b , the non-sampled position are filled with zeros (*i.e.*, applying \mathbf{S}^T) and, then, the matrix \mathbf{H}_k is applied. Applying $\mathbf{H}_k\mathbf{S}^T$ to b results in an N -length spectrum with zeros in non-sampled positions. To obtain the filtered measurements b_k , as described in pre-filtering method, the samples in the sampling pattern (applying \mathbf{S}) are selected. Therefore, b_k is obtained using the following expression:

$$b_k = \mathbf{SH}_k\mathbf{S}^T b. \quad (4.2)$$

The pre-filtering in this representation is shown in Figure 4.1. The reconstructions solve the problems in the following form:

$$\begin{aligned} & \underset{x_k}{\operatorname{argmin}} \|x_k\|_p^p \\ & \|b_k - \mathbf{SF}x_k\|_2 < \epsilon. \end{aligned} \quad (4.3)$$

The solutions have a tolerance of ϵ . The final composition is the same as the one presented in Section 2.3.

Adding measurements according to the filters

We can use information about the filters to improve the effective number of measurements in the reconstruction process. For this, the solution space is changed by adding measurements to the system.

Let \mathbf{z}_k be the vector of indices whose magnitudes of \mathbf{H}_k are smaller than a threshold t_k , *i.e.*:

$$\mathbf{z}_k = \{i | 0 < i \leq N, \|\mathbf{H}_k(i, i)\|_2 < t_k\} \quad (4.4)$$

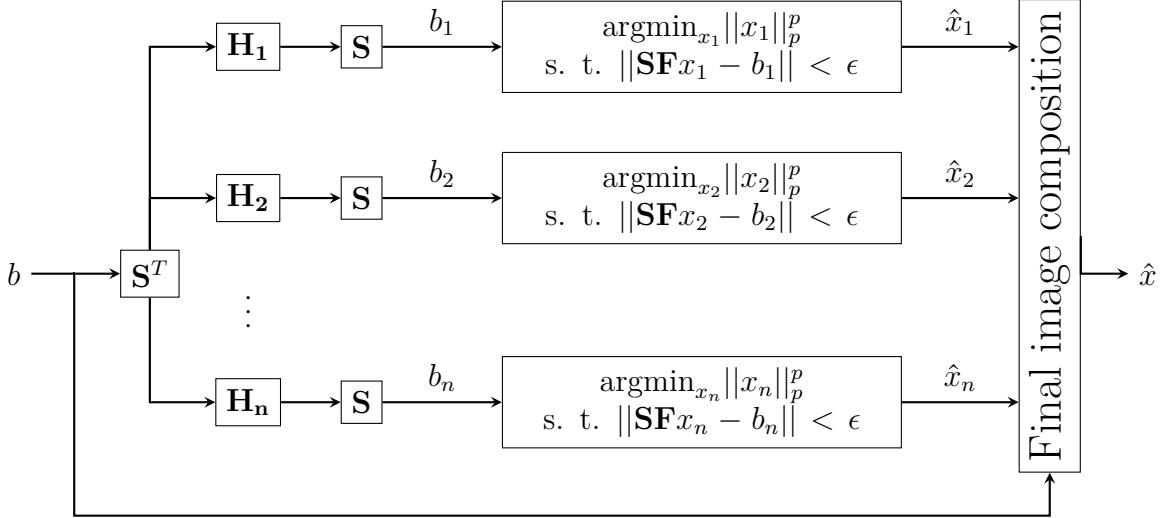


Figure 4.1: New representation of the pre-filtering method.

Let us define the operator \mathbf{Z}_k as the new sampling pattern, which selects entries of both original sampling pattern \mathbf{s} and \mathbf{z}_k , represented by:

$$\mathbf{Z}_k = \mathbf{I}_{\mathbf{s} \cup \mathbf{z}_k, *}. \quad (4.5)$$

To obtain the new b_k , we have to select the samples using \mathbf{Z}_k . Therefore, we can obtain b_k with the following equation:

$$b_k = \mathbf{Z}_k \mathbf{H}_k \mathbf{S}^T b. \quad (4.6)$$

The matrix \mathbf{S}^T , zero-fills the input in all the non-sampled positions. The matrix \mathbf{Z}_k resamples with both sampled positions and non-sampled positions below a given threshold. Therefore, the zero-valued measurements are still present in b_k .

The new reconstruction model is also modified accordingly:

$$\begin{aligned} & \text{argmin}_{x_k} ||x_k||_p^p \\ & \quad ||b_k - \mathbf{Z}_k \mathbf{F} x_k||_2 < \epsilon. \end{aligned} \quad (4.7)$$

The solution space is modified, forcing a solution that has zeros in the stop-band response of the filters. For each different filter, there is a different solution space because different filters have different passing and stopping bands. The modified model is depicted in Figure 4.2.

Figure 4.3 shows the sampled positions in the k-space for the Haar filters, with a threshold of 1% of maximum magnitude of the filters. This diagram shows how the number of measurements can be increased using this process. It also shows that different

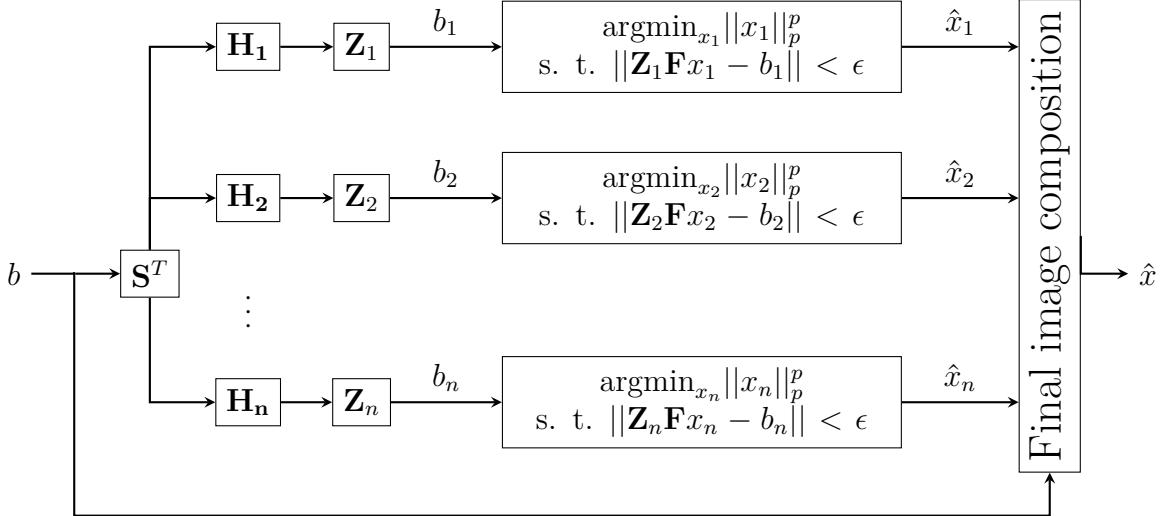


Figure 4.2: Modified pre-filtering method with extra zero-valued measurements.

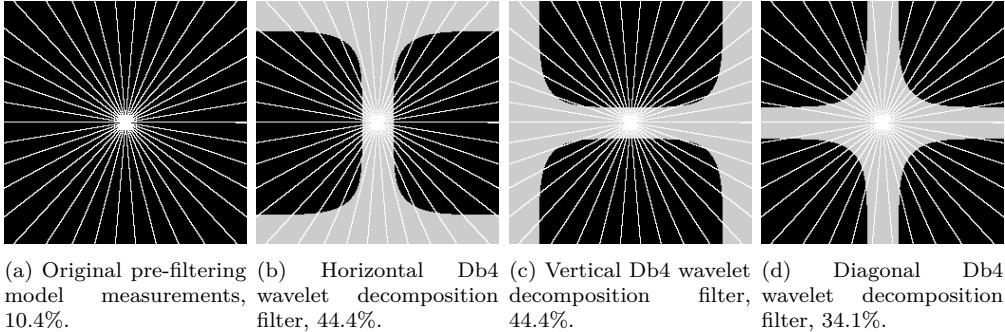


Figure 4.3: Sampled positions. White lines represent the original sampling pattern. Translucid white areas represent the extra measurements, *i.e.*, the points for which the magnitude of the filter spectrum is smaller than 1% of maximum filter magnitude.

filters have zero-valued measurements in different positions, leading to different solution spaces.

There is a balance on the tolerance of the stop-band filters threshold t_k . If t_k is small, the extra measurements may be too few to make a difference. If t_k is large, the reconstruction can be compromised because the extra measurements may increase the error.

4.2 Experimental methodology

I use the NESTA algorithm to solve reconstruction problems in the following form:

$$\begin{aligned} & \text{argmin}_{x_k} \|x_k\|_1 \\ & \text{subject to } \|\mathbf{Z}_k \mathbf{F} x_k - b_k\| < \epsilon. \end{aligned} \tag{4.8}$$

Notice that the ℓ_1 metric is used in this equation, as NESTA does not solve the ℓ_p norm for $p < 1$. To compare the reconstruction results to the last chapter results, a ϵ equal to zero is used in our simulations.

The region of zero-valued measurements included for each reconstruction is determined by 2 factors: the filter and the threshold. In this chapter, threshold values of 0%, 0.5%, 1%, and 5% of the maximum amplitude of the corresponding filter are used. These values are chosen empirically. Table 4.1 shows the average percentage of measurements for all filters of each filter bank of our experiment, including zero-valued measurements in spectrum positions below the threshold. The original measurements formed by the radial lines are also included. The number of measurements greatly increases depending on the features of the filter bank. Notice that the first threshold value is zero, which implies that only measurements with exact zero value are considered. This allows to compare these results with results of the previous chapter, *i.e.* to compare the pre-filtering method with the IRLS and NESTA algorithms.

In the experiment performed in this chapter, the same images of Chapter 3 are used, as well as the same sampling schemes (20 to 100 radial lines) and filter banks (TV, Haar, and all WIN and WAV filters). The simulation is repeated for each input parameter. Since 4 levels of threshold are included, the total number of reconstructions is 4 times the number of reconstructions performed in Chapter 3, totaling 480,700 reconstructions. The experiment was run on a computational cluster, which was described in Chapter 3. The experiment was implemented in MATLAB, based on an available NESTA implementation. The quality performance metrics are the SNR and SSIM of the reconstructed image when compared to the ground truth. The spectrum coverage and the ℓ_1 of the filter banks are also computed.

4.3 Results

Figures 4.4 and 4.5 show the average of the SNR and SSIM of the reconstructions for all simulations, considering a threshold of 0%. The figures also show horizontal lines corresponding to the TV minimization using the NESTA TV minimization, without using pre-filtering (section 2.4.3). Table 4.2 shows all average values. We can compare these results directly with the ones presented in Figures 3.7 and 3.8 and Table 3.2. This allow to evaluate the difference between the average image quality provided by IRLS and NESTA algorithms in the pre-filtering method.

In this experiment, WIN filters with lower order present the highest image quality for 20 radial lines, with a gain of more than 2 dB when compared to the state-of-the-art (Haar filter bank). These results are similar to the results obtained with the IRLS

algorithm. However, the image quality using NESTA for WIN filters is, in general, lower than the quality using IRLS. The maximum SNR value for 20 radial lines is 18.9 dB for the filter bank WIN(4,2), while the maximum SNR value for IRLS is 19.5 for the filter bank WIN(2,2). The pre-filtering with NESTA also did not match the IRLS average image quality (in SNR) for 40 or more radial lines for these filter banks. However, the maximum value obtained is very close to IRLS. For 100 radial lines, the filter bank WIN(8,4) obtained a maximum SNR value of 34.8 dB with NESTA, while for the same settings with IRLS the maximum SNR was 34.8 dB.

The WAV filter banks present results that are similar to the results obtained with IRLS, with some of them outperforming NESTA TV in terms of image quality. The maximum average SNR for a WAV filter using NESTA pre-filtering is 31.5 dB, which is the same SNR value obtained with IRLS. However, this value is still below the 32.2 dB obtained with the best performing WIN filter bank with IRLS algorithm.

Figures 4.6 and 4.7 show the average SNR and SSIM values of the reconstructions with the best image quality, for all thresholds. The reconstruction quality is always equal or superior to the plots in Figures 4.4 and 4.5. Table 4.3 shows the improvements in image quality of the results presented in Figures 4.6 and 4.7, when compared to the results in Figures 4.4 and 4.5.

For most filter banks, the improvement in image quality is small. However, the filter banks that generated a bad reconstruction with a zero threshold showed the greatest improvements. For instance, the filter bank WAV(rbio2.8,1) increased the average SNR/SSIM from 0.0/0.000 to 11.6/0.581 for 20 radial lines; from 1.2/0.045 to 16.5/0.768 for 40 radial lines; 16.0/0.666 to 21.4/0.895 for 60 lines; 25.7/0.959 to 26.2/0.966 for 80 lines and 29.9/0.985 to 30.2/0.986 for 100 radial lines. These values are still not as good as the best filter banks, but the threshold method improved the reconstruction of filter banks that performed poorly. Notice also that the method presents the highest improvement for a small number of radial lines, when the reconstruction is generally poor. In the mentioned example, the filter bank WAV(rbio2.8,1), with the zero threshold and 60 radial lines achieved 16.0/0.666 average SNR/SSIM, while the best threshold at 40 radial lines presented 16.5/0.768 average SNR/SSIM. In other words, using the proposed method can improve the results more than increasing the number of radial lines. So, in this specific scenario, the acquisition time decreased while the quality of the reconstructed image increased.

Additionally, the improvements are more prevalent in WAV filter banks than in WIN. WIN filter banks with order 2 and 4 presented no significant improvement in SNR and SSIM (as seen in Table 4.3). Other ten WIN filter banks presented average SNR/SSIM improvements between 0.1/0.002 and 0.6/0.015. None of these filter banks presented

Table 4.1: Average number of measurements of the filter bank for a percentage of the total size of the image's spectrum.

	no measurements					20 radial lines					40 radial lines					60 radial lines					80 radial lines					100 radial lines				
	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%		
TV	2.0	12.4	22.4	35.1	12.4	22.7	32.9	44.9	20.8	21.1	23.1	30.5	30.8	32.4	39.5	39.8	41.2	48.0	48.2	48.2	49.4	49.4	49.4	49.4	49.4	49.4	49.4	49.4		
Haar	3.7	8.3	12.6	23.9	14.1	18.5	22.5	32.7	24.4	28.2	31.7	40.6	30.5	32.2	37.5	39.5	40.4	41.1	45.6	48.0	48.8	49.4	53.4	53.4	53.4	53.4	53.4			
WIN(2,2)	0.0	3.5	6.9	10.2	10.4	13.9	17.3	20.6	20.8	24.2	27.7	30.9	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5	30.5			
WIN(2,3)	0.0	0.0	0.0	0.0	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4			
WIN(2,4)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
WIN(2,5)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0		
WIN(4,2)	1.1	4.5	7.9	14.3	11.5	14.9	18.3	24.0	21.8	24.8	27.8	33.0	30.5	30.5	32.3	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5	39.5		
WIN(4,3)	0.0	0.0	0.4	6.5	10.4	10.4	10.7	16.1	20.8	21.0	25.7	30.5	30.5	30.7	34.7	39.5	39.7	43.1	48.0	48.0	48.0	48.2	51.1	51.1	51.1	51.1	51.1			
WIN(4,4)	0.0	0.2	1.0	8.6	10.4	10.6	11.2	18.0	20.8	20.9	21.4	27.3	30.5	30.6	31.0	36.2	39.5	39.6	40.0	44.5	48.0	48.1	48.4	52.3	52.3	52.3	52.3	52.3		
WIN(4,5)	0.0	0.3	0.9	10.1	10.4	10.7	11.1	19.3	20.8	21.0	21.4	28.6	30.5	30.7	31.0	37.3	39.5	39.7	39.9	45.4	48.0	48.1	48.4	53.1	53.1	53.1	53.1	53.1		
WIN(6,2)	10.9	17.9	24.1	50.6	21.3	27.5	33.2	56.7	31.7	36.6	41.3	61.5	30.5	33.2	35.7	51.7	39.5	41.9	44.0	57.8	48.0	50.2	52.0	63.6	63.6	63.6	63.6	63.6		
WIN(6,3)	0.0	18.9	24.4	37.0	10.4	26.9	31.7	43.0	20.8	35.0	39.2	49.2	30.5	42.9	46.5	55.1	39.5	50.3	53.4	60.8	48.0	57.3	60.1	66.2	66.2	66.2	66.2	66.2		
WTN(6,4)	0.0	13.8	21.0	36.4	10.4	22.6	28.9	42.7	20.8	31.4	36.9	49.1	30.5	39.6	44.5	55.2	39.5	47.4	51.6	60.9	48.0	54.8	58.5	66.4	66.4	66.4	66.4	66.4		
WTN(6,5)	0.0	12.8	19.0	39.7	10.4	21.7	27.2	45.7	20.8	30.6	35.5	51.8	30.5	39.0	43.3	57.6	39.5	46.9	50.6	63.1	48.0	54.4	57.6	68.4	68.4	68.4	68.4	68.4		
WTN(8,2)	14.8	42.1	48.6	64.6	25.3	49.1	55.0	69.4	35.6	56.0	60.9	73.2	30.5	46.9	50.0	59.2	39.5	53.7	56.3	64.2	48.0	60.2	62.4	69.0	69.0	69.0	69.0	69.0		
WTN(8,3)	0.0	31.3	37.2	51.3	10.4	38.0	43.2	55.9	20.8	44.8	49.4	60.6	30.5	51.3	55.3	65.1	39.5	57.4	60.9	69.5	48.0	63.4	66.4	73.7	73.7	73.7	73.7	73.7		
WTN(8,4)	0.0	40.4	45.5	56.9	10.4	46.2	50.8	61.0	20.8	52.2	56.2	65.3	30.5	57.9	61.5	69.4	39.5	63.3	66.5	73.4	48.0	68.5	71.2	77.2	77.2	77.2	77.2	77.2		
WTN(8,5)	0.0	33.0	40.8	57.1	10.4	39.7	46.7	61.4	20.8	46.5	52.7	65.6	30.5	53.0	58.4	69.8	39.5	59.1	63.8	73.7	48.0	64.9	68.9	77.5	77.5	77.5	77.5	77.5		
WTN(10,2)	16.9	48.8	58.4	72.3	27.4	55.3	63.8	76.3	37.7	61.7	68.9	79.7	30.5	50.1	55.2	63.4	39.5	56.4	60.8	67.7	48.0	62.4	66.2	71.9	71.9	71.9	71.9	71.9		
WTN(10,3)	0.0	44.3	50.3	63.0	10.4	49.6	54.9	66.3	20.8	55.0	59.7	69.9	30.5	60.3	64.4	73.3	39.5	65.4	68.9	76.7	48.0	70.2	73.2	79.9	79.9	79.9	79.9	79.9		
WTN(10,4)	0.0	47.8	53.3	66.3	10.4	53.0	57.8	69.5	20.8	58.1	62.4	72.8	30.5	63.1	66.9	76.0	39.5	67.9	71.2	79.1	48.0	72.4	75.2	82.1	82.1	82.1	82.1	82.1		
WTN(10,5)	0.0	57.0	60.6	69.4	10.4	61.2	64.4	72.3	20.8	65.5	68.4	75.4	30.5	69.7	72.2	78.4	39.5	73.6	75.8	81.2	48.0	77.4	79.3	83.9	83.9	83.9	83.9	83.9		
WAV(db1,1)	7.9	16.2	21.5	35.7	18.3	26.1	31.2	43.9	24.4	31.7	36.3	47.9	30.5	34.8	36.5	44.2	39.5	43.5	45.0	51.8	48.0	51.6	53.0	59.0	59.0	59.0	59.0			
WAV(db1,2)	0.0	9.2	12.9	29.0	10.4	18.8	22.1	36.5	20.8	28.4	31.4	44.1	30.5	37.4	40.1	51.4	39.5	45.7	48.1	58.2	48.0	53.5	55.6	64.4	64.4	64.4	64.4	64.4		
WAV(db1,3)	0.0	14.6	19.9	41.6	10.4	23.8	28.6	48.0	20.8	32.9	37.3	54.6	30.5	41.4	45.3	60.7	39.5	67.9	71.2	79.1	48.0	72.4	75.2	82.1	82.1	82.1	82.1	82.1		
WAV(db4,1)	16.1	49.2	54.3	66.8	26.6	55.7	60.5	71.7	34.0	59.5	63.7	73.4	30.5	51.5	53.7	60.5	39.5	73.6	75.8	81.2	48.0	77.4	79.3	83.9	83.9	83.9	83.9			
WAV(db4,2)	0.0	44.4	49.1	63.3	10.4	50.1	54.2	67.0	20.8	56.0	59.6	70.8	30.5	61.7	64.9	74.6	39.5	67.1	69.9	76.7	48.0	72.2	74.6	81.8	81.8	81.8	81.8	81.8		
WAV(db4,3)	0.0	57.8	62.3	74.5	10.4	62.4	66.4	77.2	20.8	67.2	70.7	80.1	30.5	71.8	74.8	82.9	39.5	76.0	78.6	85.5	48.0	79.9	82.1	87.9	87.9	87.9	87.9	87.9		
WAV(coif1,1)	0.0	17.4	21.2	34.2	10.4	33.3	38.0	53.2	20.8	41.3	45.4	58.8	30.5	47.0	52.7	64.2	39.5	55.9	59.1	69.3	48.0	57.8	59.7	65.6	65.6	65.6	65.6	65.6		
WAV(coif1,2)	0.0	26.1	31.4	48.6	10.4	33.8	38.6	53.9	20.8	41.7	45.9	59.3	30.5	56.7	52.9	64.7	39.5	56.3	59.5	69.7	48.0	62.8	65.6	74.4	74.4	74.4	74.4	74.4		
WAV(coif1,3)	0.0	36.8	43.4	62.3	10.4	43.6	49.6	66.5	20.8	50.7	55.9	70.8	30.5	57.2	61.9	74.9	39.5	63.3	67.4	78.6	48.0	68.9	72.4	82.1	82.1	82.1	82.1			
WAV(sym2,1)	0.0	16.9	20.8	33.6	10.4	25.4	28.8	39.9	20.8	34.0	37.0	46.6	30.5	42.4	44.9	53.1	39.5	50.2	52.4	58.7	62.5	64.9	66.2	68.4	68.4	68.4	68.4			
WAV(sym2,2)	0.0	25.6	30.8	47.9	10.4	33.3	38.0	53.2	20.8	41.3	45.4	58.8	30.5	48.9	52.5	64.2	39.5	55.9	59.1	69.3	48.0	62.5	65.2	74.1	74.1	74.1	74.1	74.1		
WAV(sym2,3)	0.0	36.1	42.6	61.7	10.4	43.0	48.9	66.0	20.8	50.1	55.3	70.3	30.5	56.7	61.3	74.5	39.5	62.8	66.9	78.3	48.0	68.5	72.0	81.8	81.8	81.8	81.8			
WAV(rbio2,1)	0.0	18.5	22.6	35.3	10.4	27.0	30.6	41.7	20.8	35.7	38.8	48.4	30.5	44.0	46.7	55.0	39.5	51.8	54.2	61.2	48.0	59.0	61.1	67.1	67.1	67.1	67.1			
WAV(rbio2,2)	0.0	29.3	54.2	10.4	36.8	42.3	59.2	20.8	44.6	49.5	64.5	30.5	32.4	44.9	53.1	39.5	50.2	52.4	58.7	62.5	64.9	66.2	68.4	68.4	68.4	68.4	68.4			
WAV(rbio2,2,3)	0.0	43.4	51.1	66.1	10.4	49.8	56.8	70.0	20.8	56.3	62.5	74.1	30.5	37.5	39.5	39.5	39.5	39.5	39.5	67.8	81.3	48.0	72.8	77.0	84.4	84.4	84.4	84.4		
WAV(rbio2,8,1)	0.0	38.3	42.1	53.0	10.4	43.6	47.1	56.9	20.8	52.3	61.0	74.1	30.5	54.9	57.5	65.1	39.5	60.5	62.7	69.1	48.0	65.9	67.8	73.2	73.2	73.2	73.2	73.2		
WAV(rbio2,8,2)	0.0	47.8	52.9	67.9	10.4	52.6	57.3	71.0	20.8	57.7	61.9	74.1	30.5	62.8	66.4	74.6	39.5	67.7	70.											

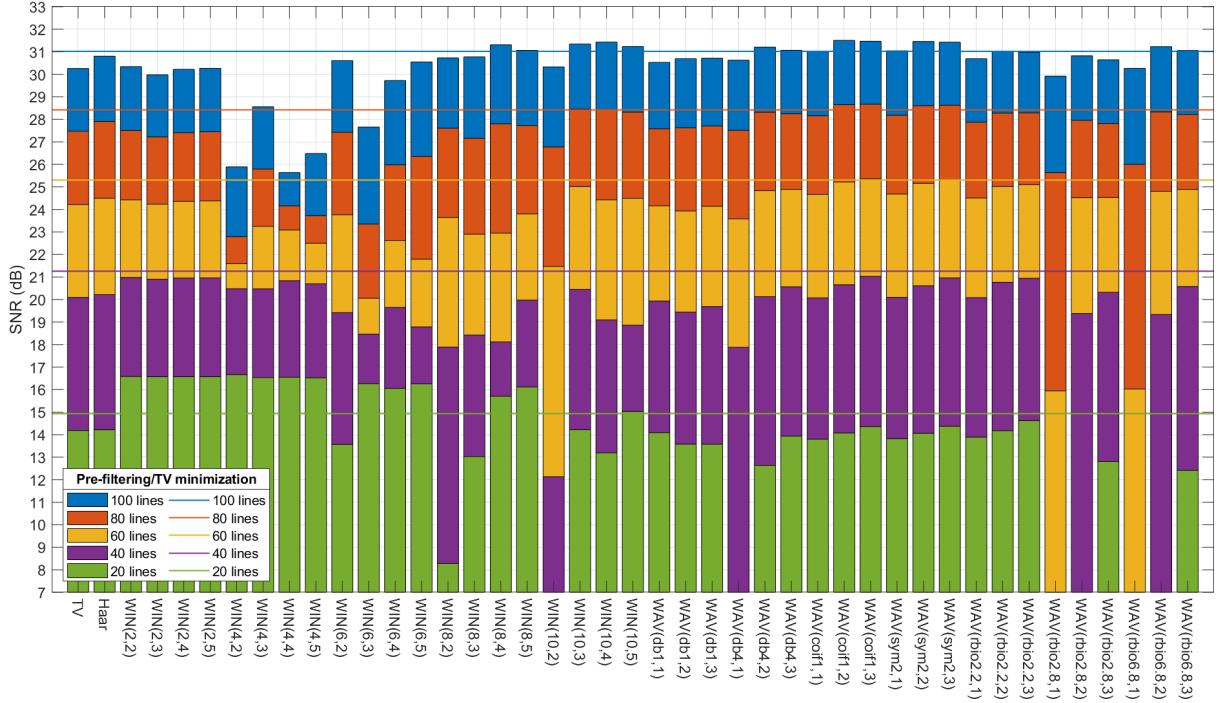


Figure 4.4: Average SNR(dB) of the reconstruction using the pre-filtering with NESTA for the filter banks tested. Average SNR values for reconstructions with the NESTA TV minimization are shown as horizontal lines.

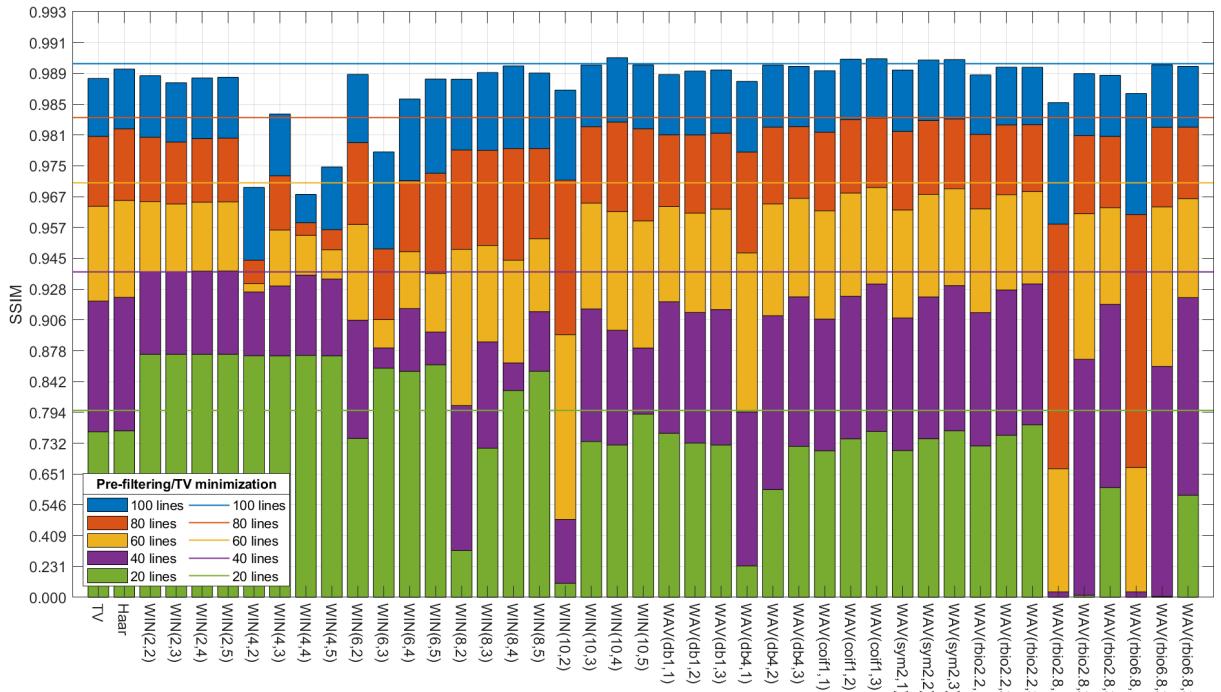


Figure 4.5: Average SSIM of the reconstruction using the pre-filtering with NESTA for the filter banks tested. Average SSIM values for reconstructions with the NESTA TV minimization are shown as horizontal lines.

Table 4.2: Average reconstruction SNR(dB), SSIM, and time using pre-filtering with NESTA with zero threshold, classified by radial lines. The number of filters in each filter bank, the average ℓ_1 of the images and the spectrum coverage are also shown. Numbers in bold correspond to filter banks with the best values.

	20 radial lines			40 radial lines			60 radial lines			80 radial lines			100 radial lines			avg ℓ_1	spec. cov.
	SNR	SSIM	time	SNR	SSIM	time											
TV	14.2	0.757	25.4	20.1	0.920	16.1	24.2	0.964	12.1	27.5	0.980	9.8	30.3	0.988	8.5	31.6	95.7
Haar	14.2	0.759	29.7	20.2	0.923	27.0	24.5	0.966	20.3	27.9	0.982	16.0	30.8	0.989	13.0	32.4	99.0
WIN(2,2)	16.6	0.874	4.1	21.0	0.938	4.3	24.4	0.966	4.5	27.5	0.980	4.6	30.3	0.988	4.7	78.2	90.9
WIN(2,3)	16.6	0.874	11.0	20.9	0.938	11.6	24.2	0.965	12.3	27.2	0.979	12.5	210.0	0.988	12.9	79.3	100.0
WIN(2,4)	16.6	0.874	20.9	20.9	0.938	21.6	24.4	0.966	23.5	27.4	0.980	23.5	30.2	0.988	24.1	79.8	100.0
WIN(2,5)	16.6	0.874	33.7	21.0	0.938	35.0	24.4	0.966	37.7	27.5	0.980	37.8	30.3	0.988	38.7	80.1	100.0
WIN(4,2)	16.7	0.873	7.8	20.5	0.926	15.6	21.6	0.931	20.6	22.8	0.944	22.3	25.9	0.970	18.6	64.6	78.6
WIN(4,3)	16.5	0.873	29.6	20.5	0.930	54.2	23.2	0.957	53.5	25.8	0.973	44.4	28.6	0.984	35.7	79.9	100.0
WIN(4,4)	16.5	0.873	66.5	20.8	0.936	88.8	23.1	0.954	85.7	24.2	0.959	79.3	25.6	0.968	71.4	84.0	100.0
WIN(4,5)	16.5	0.873	108.8	20.7	0.934	135.7	22.5	0.949	140.3	23.7	0.957	130.4	26.5	0.975	110.1	85.5	100.0
WIN(6,2)	13.6	0.742	34.6	19.4	0.906	21.2	23.8	0.959	15.8	27.4	0.979	13.3	30.6	0.988	11.6	35.6	75.7
WIN(6,3)	16.3	0.859	42.1	18.5	0.881	50.2	20.1	0.907	51.0	23.4	0.949	44.7	27.7	0.978	38.3	66.3	99.1
WIN(6,4)	16.1	0.855	125.9	19.7	0.915	102.4	22.6	0.948	84.3	26.0	0.971	73.7	29.7	0.986	64.2	79.7	100.0
WIN(6,5)	16.2	0.863	184.6	18.8	0.896	171.2	21.8	0.937	139.4	26.4	0.973	121.0	30.5	0.988	105.1	85.5	100.0
WIN(8,2)	8.3	0.330	18.8	17.9	0.805	15.0	23.6	0.949	12.9	27.6	0.978	11.3	30.7	0.988	10.5	32.6	74.8
WIN(8,3)	13.0	0.720	73.7	18.4	0.887	56.2	22.9	0.950	44.8	27.2	0.978	37.2	30.8	0.989	32.5	42.6	95.4
WIN(8,4)	15.7	0.829	92.3	18.1	0.865	92.6	23.0	0.944	79.8	27.8	0.978	70.8	31.3	0.989	61.5	66.2	100.0
WIN(8,5)	16.1	0.855	185.2	110.0	0.913	139.8	23.8	0.953	122.4	27.7	0.978	107.7	31.1	0.989	94.8	79.5	100.0
WIN(10,2)	3.8	0.113	18.9	12.1	0.486	14.7	21.5	0.894	13.0	26.8	0.972	11.4	30.3	0.987	10.6	33.3	74.5
WIN(10,3)	14.2	0.736	50.1	20.5	0.915	39.1	25.0	0.965	32.7	28.5	0.982	29.8	31.3	0.989	26.7	33.0	93.6
WIN(10,4)	13.2	0.728	125.1	19.1	0.898	94.2	24.4	0.963	77.6	28.4	0.983	67.1	31.4	0.990	58.3	47.9	98.2
WIN(10,5)	15.0	0.791	149.4	18.9	0.881	132.9	24.5	0.960	115.2	28.3	0.982	101.3	31.2	0.989	90.9	67.0	100.0
WAV(db1,1)	14.1	0.754	35.2	19.9	0.920	25.1	24.2	0.964	18.9	27.6	0.981	15.2	30.5	0.988	13.0	33.0	98.3
WAV(db1,2)	13.6	0.732	81.2	19.4	0.912	63.4	23.9	0.962	53.8	27.6	0.981	48.5	30.7	0.989	39.0	39.1	99.9
WAV(db1,3)	13.6	0.728	170.4	19.7	0.914	123.9	24.1	0.964	104.1	27.7	0.981	90.1	30.7	0.989	73.8	50.9	100.0
WAV(db4,1)	5.2	0.236	24.8	17.9	0.795	19.5	23.6	0.947	16.2	27.5	0.978	13.8	30.6	0.988	12.8	32.1	91.4
WAV(db4,2)	12.6	0.601	70.1	20.1	0.910	56.3	24.8	0.965	52.7	28.3	0.982	42.6	31.2	0.989	35.9	38.6	98.5
WAV(db4,3)	13.9	0.725	165.8	20.6	0.923	113.2	24.9	0.967	99.6	28.2	0.982	84.3	31.1	0.989	70.9	51.8	99.7
WAV(coif1,1)	13.8	0.714	27.7	20.1	0.907	20.8	24.7	0.963	16.7	28.2	0.981	14.2	31.0	0.989	12.7	31.6	95.0
WAV(coif1,2)	14.1	0.741	74.2	20.7	0.923	58.4	25.2	0.968	54.0	28.7	0.983	43.2	31.5	0.990	36.2	37.6	99.4
WAV(coif1,3)	14.4	0.757	168.3	21.0	0.931	116.7	25.4	0.970	100.2	28.7	0.983	84.2	31.5	0.990	70.4	50.7	99.9
WAV(sym2,1)	13.8	0.714	27.6	20.1	0.908	20.5	24.7	0.963	16.5	28.2	0.981	14.2	31.0	0.989	12.5	31.7	95.3
WAV(sym2,2)	14.1	0.742	74.9	20.6	0.923	58.4	25.2	0.968	54.3	28.6	0.983	43.4	31.5	0.990	36.6	37.9	99.4
WAV(sym2,3)	14.4	0.759	167.8	21.0	0.930	116.6	25.3	0.969	100.0	28.6	0.983	84.7	31.4	0.990	70.6	50.9	99.9
WAV(rbio2,2,1)	13.9	0.725	32.2	20.1	0.912	23.8	24.5	0.964	18.8	27.9	0.981	15.5	30.7	0.988	13.3	32.7	97.2
WAV(rbio2,2,2)	14.2	0.749	85.2	20.8	0.927	64.0	25.0	0.968	56.4	28.3	0.982	48.4	31.0	0.989	39.2	42.0	99.7
WAV(rbio2,2,3)	14.6	0.771	175.6	20.9	0.931	122.7	25.1	0.969	106.3	28.3	0.982	90.3	31.0	0.989	75.2	57.1	100.0
WAV(rbio2,8,1)	0.0	0.000	26.4	1.2	0.045	20.5	16.0	0.666	17.2	25.6	0.959	14.9	29.9	0.985	12.7	33.3	88.3
WAV(rbio2,8,2)	0.4	0.017	76.9	19.4	0.869	61.3	24.5	0.962	54.2	28.0	0.981	44.6	30.8	0.989	37.8	40.7	97.8
WAV(rbio2,8,3)	12.8	0.608	165.2	20.3	0.918	121.6	24.5	0.964	101.0	27.8	0.980	86.6	30.6	0.988	73.8	55.3	99.6
WAV(rbio6,8,1)	0.0	0.000	25.0	1.2	0.045	19.4	16.0	0.670	16.7	26.0	0.962	13.7	30.3	0.986	12.7	32.6	88.9
WAV(rbio6,8,2)	0.3	0.009	69.9	19.3	0.861	55.1	24.8	0.964	50.8	28.3	0.982	41.3	31.2	0.989	33.6	38.7	97.9
WAV(rbio6,8,3)	12.4	0.582	153.1	20.6	0.923	104.6	24.9	0.967	89.2	28.2	0.982	73.8	31.1	0.989	60.7	52.7	99.6

statistically significant improvements (at 95% confidence level). The only two WIN filter banks that presented significant improvements are WIN(8,2) and WIN(10,2).

Figure 4.8 shows a qualitative and quantitative comparison of the the reconstructions for four different thresholds. In this Figure, each row corresponds to different experiment setting, and a different image, filter bank and sampling scheme (given by the number of radial lines); while each column indicates the level of the threshold, which varies from 0% to 5% of the maximum amplitude of the filter spectrum.

Last row of Figure 4.8 shows the reconstructed image with best quality obtained. When using the threshold method, the SNR increased from 34.8 dB to 34.9 dB for the same filter bank WIN(8,4), with a threshold of 5%. There is a clear improvement in the 2

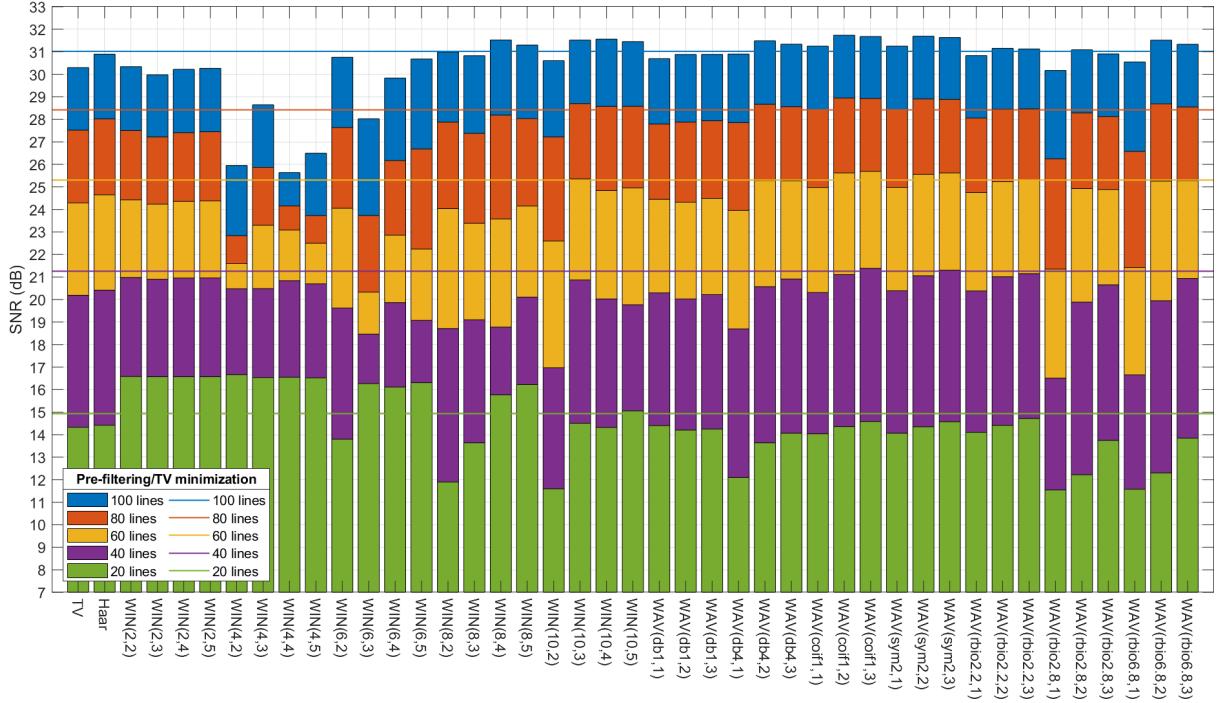


Figure 4.6: Average SNR(dB) of the reconstruction using the pre-filtering with NESTA for the filter banks tested. Average SNR values for reconstructions with the NESTA TV minimization are shown as horizontal lines.

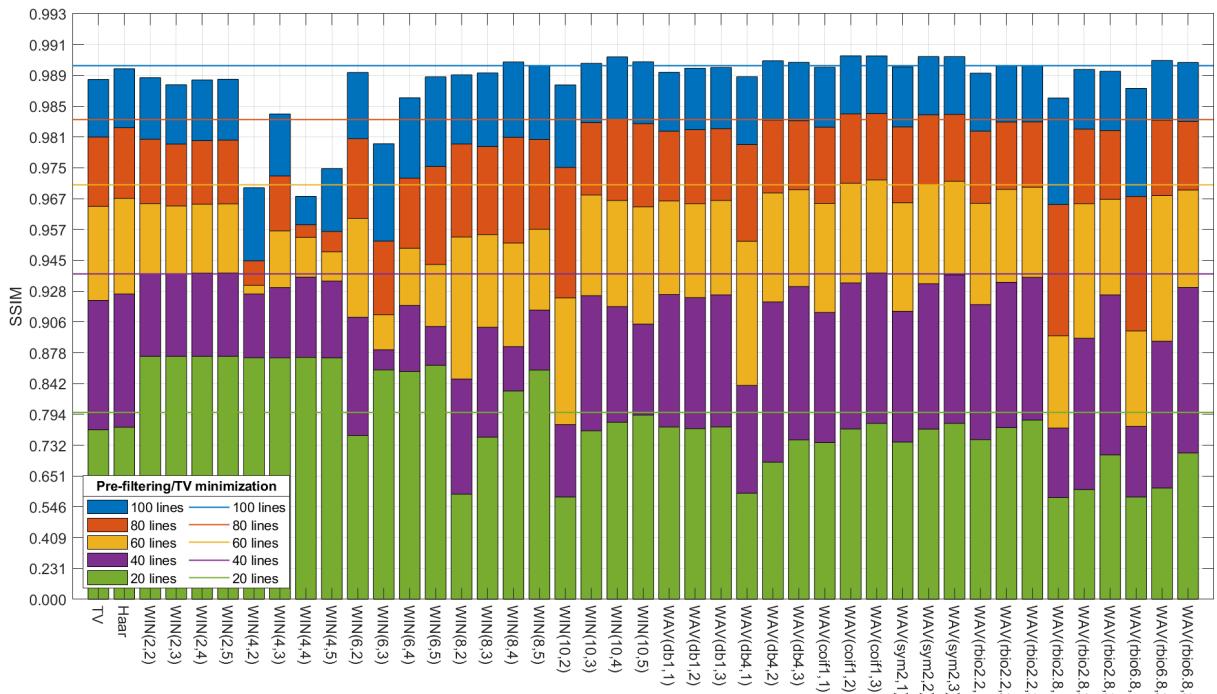


Figure 4.7: Average SSIM of the reconstruction using the pre-filtering with NESTA for the filter banks tested. Average SSIM values for reconstructions with the NESTA TV minimization are shown as horizontal lines.

Table 4.3: Difference of the average SNR (dB) and SSIM from the best reconstruction with threshold (Figures 4.6 and 4.7), and the reconstruction with zero threshold (Figures 4.4 and 4.5), classified by filter banks and radial lines.

Num radial lines	Average improvement of the SNR						Average improvement of the SSIM					
	20	40	60	80	100	average	20	40	60	80	100	average
TV	0.1	0.1	0.1	0.1	0.0	0.1	0.008	0.002	0.001	0.000	0.000	0.002
Haar	0.2	0.2	0.1	0.1	0.1	0.1	0.011	0.003	0.001	0.000	0.000	0.003
WIN(2,2)	0.0	0.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000
WIN(2,3)	0.0	0.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000
WIN(2,4)	0.0	0.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000
WIN(2,5)	0.0	0.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000
WIN(4,2)	0.0	0.0	0.0	0.0	0.1	0.0	0.000	0.000	0.000	0.001	0.000	0.000
WIN(4,3)	0.0	0.0	0.1	0.1	0.1	0.0	0.000	0.000	0.000	0.000	0.000	0.000
WIN(4,4)	0.0	0.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000
WIN(4,5)	0.0	0.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.000	0.000	0.000
WIN(6,2)	0.2	0.2	0.3	0.2	0.1	0.2	0.011	0.004	0.003	0.001	0.000	0.004
WIN(6,3)	0.0	0.0	0.3	0.4	0.4	0.2	0.000	0.000	0.005	0.004	0.002	0.002
WIN(6,4)	0.1	0.2	0.2	0.2	0.1	0.2	0.002	0.004	0.002	0.001	0.000	0.002
WIN(6,5)	0.1	0.3	0.5	0.3	0.1	0.2	0.002	0.007	0.006	0.002	0.000	0.003
WIN(8,2)	3.6	0.8	0.4	0.3	0.3	1.1	0.263	0.042	0.006	0.001	0.001	0.063
WIN(8,3)	0.6	0.7	0.5	0.2	0.1	0.4	0.029	0.015	0.005	0.001	0.000	0.010
WIN(8,4)	0.1	0.7	0.6	0.4	0.2	0.4	0.002	0.020	0.008	0.002	0.001	0.007
WIN(8,5)	0.1	0.1	0.4	0.3	0.2	0.2	0.004	0.003	0.004	0.002	0.001	0.003
WIN(10,2)	7.8	4.8	1.1	0.4	0.3	2.9	0.469	0.289	0.030	0.003	0.001	0.159
WIN(10,3)	0.3	0.4	0.3	0.2	0.2	0.3	0.027	0.010	0.003	0.001	0.000	0.008
WIN(10,4)	1.1	0.9	0.4	0.2	0.1	0.6	0.052	0.020	0.004	0.001	0.000	0.015
WIN(10,5)	0.0	0.9	0.5	0.3	0.2	0.4	0.002	0.024	0.005	0.001	0.000	0.006
WAV(db1,1)	0.3	0.4	0.3	0.2	0.2	0.3	0.017	0.006	0.002	0.001	0.000	0.005
WAV(db1,2)	0.6	0.6	0.4	0.3	0.2	0.4	0.035	0.012	0.003	0.001	0.000	0.010
WAV(db1,3)	0.7	0.5	0.3	0.2	0.2	0.4	0.043	0.011	0.003	0.001	0.000	0.012
WAV(db4,1)	6.9	0.8	0.4	0.3	0.3	1.8	0.360	0.044	0.006	0.002	0.001	0.083
WAV(db4,2)	1.0	0.5	0.4	0.3	0.3	0.5	0.089	0.011	0.004	0.001	0.001	0.021
WAV(db4,3)	0.1	0.3	0.4	0.3	0.3	0.3	0.019	0.008	0.003	0.001	0.001	0.006
WAV(coif1,1)	0.2	0.2	0.3	0.3	0.2	0.3	0.024	0.007	0.003	0.001	0.001	0.007
WAV(coif1,2)	0.3	0.5	0.4	0.3	0.2	0.3	0.025	0.009	0.003	0.001	0.000	0.008
WAV(coif1,3)	0.2	0.4	0.3	0.3	0.2	0.3	0.020	0.007	0.002	0.001	0.000	0.006
WAV(sym2,1)	0.2	0.3	0.3	0.2	0.2	0.3	0.024	0.007	0.003	0.001	0.000	0.007
WAV(sym2,2)	0.3	0.4	0.4	0.3	0.2	0.3	0.025	0.009	0.003	0.001	0.000	0.008
WAV(sym2,3)	0.2	0.3	0.3	0.3	0.2	0.3	0.019	0.007	0.002	0.001	0.000	0.006
WAV(rbio2,2,1)	0.2	0.3	0.2	0.2	0.1	0.2	0.019	0.007	0.002	0.001	0.000	0.006
WAV(rbio2,2,2)	0.2	0.2	0.2	0.2	0.1	0.2	0.020	0.006	0.002	0.001	0.000	0.006
WAV(rbio2,2,3)	0.1	0.2	0.2	0.2	0.1	0.2	0.013	0.005	0.002	0.001	0.000	0.004
WAV(rbio2,8,1)	11.6	15.3	5.4	0.6	0.2	6.6	0.581	0.723	0.228	0.007	0.001	0.308
WAV(rbio2,8,2)	11.8	0.5	0.4	0.3	0.3	2.7	0.591	0.023	0.004	0.001	0.001	0.124
WAV(rbio2,8,3)	0.9	0.3	0.4	0.3	0.3	0.4	0.100	0.008	0.003	0.001	0.001	0.023
WAV(rbio6,8,1)	11.6	15.4	5.4	0.6	0.3	6.7	0.582	0.726	0.229	0.006	0.001	0.309
WAV(rbio6,8,2)	12.0	0.6	0.5	0.4	0.3	2.8	0.604	0.029	0.004	0.001	0.001	0.128
WAV(rbio6,8,3)	1.4	0.3	0.4	0.3	0.3	0.6	0.132	0.008	0.003	0.001	0.001	0.029

first rows of Figure 4.8. The image cannot be reconstructed with a threshold of zero, given that the artifacts greatly compromise the visual quality of the image to a point that it is not recognizable. However, when a threshold of 0.5%, 1% or 5% is used, a recognizable image can be reconstructed. In these cases, the SNR and SSIM improves for a higher threshold. For 0.5%, the images seem to present ringing and noise artifacts, while with 5% the images have less noise, but are slightly blurry.

In general, the reconstruction quality increases with the threshold value, as can be seen in rows 5, 6 and 7 of Figure 4.8. However, the third row of Figure 4.8 shows an example of a setting where the threshold of 0.5% generated a higher SNR value than the threshold of 1% (and 5% is still the highest). The fourth row shows a settings where the

threshold of 1% corresponds to the best reconstruction quality. In this case, the quality decreases when using a 5% threshold, when compared to cases which the method was not used.

Table 4.4 and 4.5 present respectively the average SNR and SSIM for all threshold values. The best result among all threshold values is shown in bold. For 74.4% of the cases, a 5% threshold corresponds to the best SNR and SSIM values. A threshold of 1% produced the best quality for 11.6% of the cases, while a threshold of 0.5% rarely produced the best result (only for 2.3%¹). The best threshold value for maximizing the reconstruction quality may seem difficult to predict. An empirical approach to find it may be a good strategy, since it seems that the best threshold value depends on the filter bank, the number of radial lines and probably the content of the image. In this work, a limited number of threshold values were tested, albeit we observed that a threshold value of 5% produced the best results, in general. But, it is possible that other threshold values can lead to even better results.

4.4 Conclusions

In this chapter, I expanded the pre-filtering method. I changed the method to include more measurements with zero values in the reconstruction. The inclusion of these measurements was based on the filters, which have spectrum areas with a low energy. Therefore, the filtered image would have low energy in these positions and the measurements in these positions would be very close to zero. I tested the method using NESTA as the reconstruction algorithm, therefore, I was able to compare to the IRLS reconstruction algorithm.

In your tests, we observed that the NESTA algorithm does not outperform the IRLS, in terms of image quality. The image quality produced by NESTA and IRLS is similar for WAV filter banks, and image quality produced by IRLS is higher than NESTA's. We observed that the inclusion of the zero-valued measurement either improves the reconstruction quality or does not change it. In most cases, the highest improvements was for the 5% threshold. Other values of threshold might lead to even better results, but the best value seems to be filter and content dependent. In the worst case, using the threshold method do not change the results, but in most cases, it can improve them, therefore, it is worth using the method.

I only implemented the inclusion of zero measurements using the NESTA algorithm. Future works include to test the method with IRLS. However, the filter banks that

¹Notice that the 3 values do not sum to 100%. I am not considering the cases that all threshold values present the same reconstruction quality, because none of them shows the highest quality

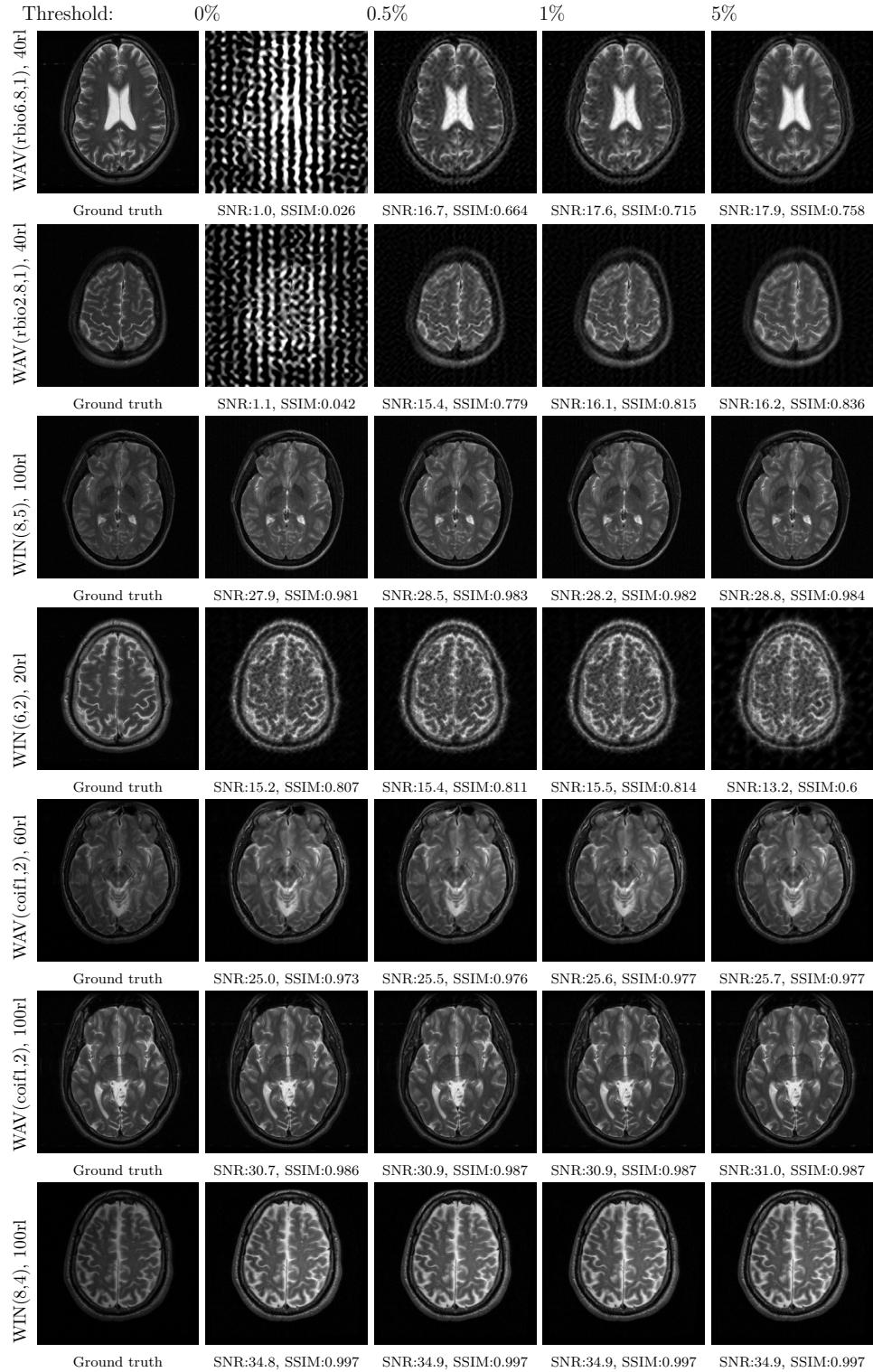


Figure 4.8: MRI reconstructions with pre-filtering including zero valued measurements. 1st column: Ground truth images. 2nd to 5-th columns: Reconstructions with zero-valued measurements with threshold respective to 0%, 0.5%, 1% and 5% of the maximum value of the filter. The label in the left indicates the filter bank used and the number of radial lines of the simulation of the row.

Table 4.4: Average SNR (dB) of the reconstruction using pre-filtering with threshold.

	20 radial lines				40 radial lines				60 radial lines				80 radial lines				100 radial lines			
	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%
TV	14.2	14.2	14.2	14.3	20.1	20.1	20.1	20.2	24.2	24.2	24.2	24.3	27.5	27.5	27.5	27.5	30.3	30.3	30.3	30.3
Haar	14.2	14.3	14.3	14.4	20.2	20.3	20.3	20.4	24.5	24.5	24.5	24.7	27.9	27.9	27.9	28.0	30.8	30.8	30.8	30.9
WIN(2,2)	16.6	16.6	16.6	16.6	21.0	21.0	21.0	21.0	24.4	24.4	24.4	24.4	27.5	27.5	27.5	27.5	30.3	30.3	30.3	30.3
WIN(2,3)	16.6	16.6	16.6	16.6	20.9	20.9	20.9	20.9	24.2	24.2	24.2	24.2	27.2	27.2	27.2	27.2	210.0	210.0	210.0	210.0
WIN(2,4)	16.6	16.6	16.6	16.6	20.9	20.9	20.9	20.9	24.4	24.4	24.4	24.4	27.4	27.4	27.4	27.4	30.2	30.2	30.2	30.2
WIN(2,5)	16.6	16.6	16.6	16.6	21.0	21.0	21.0	21.0	24.4	24.4	24.4	24.4	27.5	27.5	27.5	27.5	30.3	30.3	30.3	30.3
WIN(4,2)	16.7	16.7	16.7	20.5	20.5	20.5	20.5	21.6	21.6	21.6	21.6	22.8	22.8	22.8	22.8	25.9	25.9	25.9	25.9	
WIN(4,3)	16.5	16.5	16.5	20.5	20.5	20.5	20.5	23.2	23.2	23.2	23.2	23.3	25.8	25.8	25.8	25.9	28.6	28.6	28.6	28.7
WIN(4,4)	16.5	16.5	16.5	16.5	20.8	20.8	20.8	20.8	23.1	23.1	23.1	23.1	24.2	24.2	24.2	24.2	25.6	25.6	25.6	25.6
WIN(4,5)	16.5	16.5	16.5	16.5	20.7	20.7	20.7	20.7	22.5	22.5	22.5	22.5	23.7	23.7	23.7	23.7	26.5	26.5	26.5	26.5
WIN(6,2)	13.6	13.7	13.8	12.6	19.4	19.5	19.6	19.6	23.8	23.8	23.9	24.0	27.5	27.5	27.5	27.6	30.6	30.6	30.7	30.8
WIN(6,3)	16.3	16.3	16.3	16.3	18.5	18.5	18.5	18.4	20.1	20.1	20.1	20.3	23.4	23.5	23.6	23.7	27.7	27.8	27.9	28.0
WIN(6,4)	16.1	16.1	16.1	16.1	19.7	19.7	19.8	19.9	22.6	22.7	22.8	22.9	26.0	26.1	26.1	26.2	29.7	29.8	29.8	29.8
WIN(6,5)	16.2	16.3	16.3	16.3	18.8	18.8	18.8	19.1	21.8	21.9	21.9	22.2	26.4	26.4	26.5	26.7	30.5	30.6	30.6	30.7
WIN(8,2)	8.3	11.9	11.9	11.8	17.9	18.6	18.7	17.6	23.6	24.0	24.0	23.1	27.6	27.8	27.9	30.7	30.9	30.9	31.0	
WIN(8,3)	13.0	13.3	13.3	13.6	18.4	18.8	18.8	19.1	22.9	23.2	23.2	23.4	27.2	27.3	27.3	27.4	30.8	30.8	30.8	30.8
WIN(8,4)	15.7	15.8	15.8	15.8	18.1	18.5	18.6	18.8	23.0	23.4	23.5	23.6	27.8	28.0	28.1	28.2	31.3	31.4	31.5	31.5
WIN(8,5)	16.1	16.2	16.2	16.2	110.0	20.0	20.0	20.1	23.8	24.0	24.0	24.2	27.7	27.8	27.9	28.0	31.1	31.2	31.2	31.3
WIN(10,2)	3.8	9.7	10.9	11.6	12.1	16.2	17.0	16.7	21.5	22.2	22.6	21.5	26.8	27.2	27.2	26.3	30.3	30.6	30.4	
WIN(10,3)	14.2	14.4	14.5	13.7	20.5	20.6	20.7	20.9	25.0	25.2	25.3	28.5	28.6	28.6	28.7	31.3	31.4	31.5	31.5	
WIN(10,4)	13.2	13.8	14.0	14.3	19.1	19.7	19.8	20.0	24.4	24.7	24.7	24.8	28.4	28.5	28.6	28.6	31.4	31.5	31.6	31.6
WIN(10,5)	15.0	15.1	15.0	15.0	18.9	19.6	19.6	19.8	24.5	24.8	24.9	25.0	28.3	28.5	28.5	28.6	31.2	31.4	31.4	31.4
WAV(db1,1)	14.1	14.2	14.2	14.4	19.9	20.1	20.1	20.3	24.2	24.3	24.3	24.4	27.6	27.7	27.8	30.5	30.6	30.6	30.7	
WAV(db1,2)	13.6	13.9	14.0	14.2	19.4	19.7	19.8	20.0	23.9	24.1	24.2	24.3	27.6	27.8	27.9	30.7	30.8	30.8	30.9	
WAV(db1,3)	13.6	14.0	14.0	14.2	19.7	110.0	20.0	20.2	24.1	24.3	24.4	24.5	27.7	27.8	27.9	30.7	30.8	30.8	30.9	
WAV(db4,1)	5.2	12.1	12.1	11.9	17.9	18.7	18.7	17.8	23.6	23.9	24.0	23.5	27.5	27.8	27.8	30.6	30.8	30.8	30.9	
WAV(db4,2)	12.6	13.3	13.6	13.2	20.1	20.5	20.5	20.6	24.8	25.2	25.2	25.3	28.3	28.6	28.6	28.7	31.2	31.4	31.5	31.5
WAV(db4,3)	13.9	14.1	14.0	14.1	20.6	20.9	20.9	20.9	24.9	25.2	25.2	25.3	28.2	28.5	28.5	28.6	31.1	31.3	31.3	31.3
WAV(coifl,1)	13.8	14.0	14.0	12.9	20.1	20.3	20.3	20.3	24.7	24.8	24.9	25.0	28.2	28.3	28.3	28.4	31.0	31.2	31.2	31.2
WAV(coifl,2)	14.1	14.3	14.3	14.4	20.7	21.0	21.0	21.1	25.2	25.5	25.6	25.6	28.7	28.9	28.9	28.9	31.5	31.7	31.7	31.7
WAV(coifl,3)	14.4	14.6	14.6	14.6	21.0	21.3	21.3	21.4	25.4	25.6	25.6	25.7	28.7	28.9	28.9	28.9	31.5	31.6	31.6	31.7
WAV(sym2,1)	13.8	14.0	14.1	13.1	20.1	20.3	20.3	20.4	24.7	24.8	24.9	25.0	28.2	28.3	28.4	28.4	31.0	31.2	31.2	31.2
WAV(sym2,2)	14.1	14.3	14.3	14.3	20.6	20.9	20.9	21.1	25.2	25.5	25.5	25.6	28.6	28.8	28.8	28.9	31.5	31.6	31.6	31.7
WAV(sym2,3)	14.4	14.6	14.6	14.6	21.0	21.2	21.3	21.3	25.3	25.5	25.5	25.6	28.6	28.8	28.8	28.9	31.4	31.6	31.6	31.6
WAV(rbio2.2,1)	13.9	14.1	14.1	13.5	20.1	20.2	20.3	20.4	24.5	24.6	24.7	24.8	27.9	28.0	28.0	28.1	30.7	30.8	30.8	30.8
WAV(rbio2.2,2)	14.2	14.3	14.4	14.4	20.8	20.9	21.0	21.0	25.0	25.2	25.2	25.2	28.3	28.4	28.4	28.4	31.0	31.1	31.1	31.1
WAV(rbio2.2,3)	14.6	14.7	14.7	14.7	20.9	21.1	21.1	21.1	25.1	25.2	25.3	25.3	28.3	28.4	28.4	28.5	31.0	31.1	31.1	31.1
WAV(rbio2.8,1)	0.0	11.0	11.4	11.6	1.2	15.7	16.5	16.5	16.0	20.7	21.4	21.1	25.6	26.1	26.2	25.7	29.9	30.1	30.2	210.0
WAV(rbio2.8,2)	0.4	11.7	12.2	12.2	19.4	19.8	19.9	19.4	24.5	24.9	24.9	24.9	28.0	28.2	28.2	28.3	30.8	31.0	31.1	31.1
WAV(rbio2.8,3)	12.8	13.5	13.5	13.7	20.3	20.6	20.7	20.7	24.5	24.8	24.9	24.9	27.8	28.1	28.1	28.1	30.6	30.9	30.9	30.9
WAV(rbio6.8,1)	0.0	10.7	11.3	11.6	1.2	15.5	16.3	16.7	16.0	20.6	21.3	21.4	26.0	26.5	26.6	26.2	30.3	30.5	30.6	30.4
WAV(rbio6.8,2)	0.3	11.7	12.2	12.3	19.3	19.8	19.9	19.6	24.8	25.2	25.2	25.2	28.3	28.6	28.7	28.7	31.2	31.5	31.5	31.5
WAV(rbio6.8,3)	12.4	13.4	13.4	13.9	20.6	20.9	20.9	20.9	24.9	25.2	25.3	25.3	28.2	28.5	28.5	28.6	31.1	31.3	31.3	31.3

presented the best values with IRLS presented no improvements (in terms of reconstruction quality) when zero-valued measurements were added.

Table 4.5: Average SSIM of the reconstruction using pre-filtering with threshold.

	20 radial lines				40 radial lines				60 radial lines				80 radial lines				100 radial lines				
	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	0	0.5%	1%	5%	
TV	0.757	0.758	0.758	0.765	0.920	0.921	0.921	0.922	0.964	0.965	0.965	0.965	0.980	0.980	0.980	0.981	0.988	0.988	0.988	0.988	
Haar	0.759	0.761	0.762	0.770	0.923	0.923	0.924	0.926	0.966	0.966	0.967	0.967	0.982	0.982	0.982	0.982	0.989	0.989	0.989	0.989	
WIN(2,2)	0.874	0.874	0.874	0.874	0.938	0.938	0.938	0.938	0.966	0.966	0.966	0.966	0.980	0.980	0.980	0.980	0.988	0.988	0.988	0.988	
WIN(2,3)	0.874	0.874	0.874	0.874	0.938	0.938	0.938	0.938	0.965	0.965	0.965	0.965	0.979	0.979	0.979	0.979	0.988	0.988	0.988	0.988	
WIN(2,4)	0.874	0.874	0.874	0.874	0.938	0.938	0.938	0.938	0.966	0.966	0.966	0.966	0.980	0.980	0.980	0.980	0.988	0.988	0.988	0.988	
WIN(2,5)	0.874	0.874	0.874	0.874	0.938	0.938	0.938	0.938	0.966	0.966	0.966	0.966	0.980	0.980	0.980	0.980	0.988	0.988	0.988	0.988	
WIN(4,2)	0.873	0.873	0.873	0.872	0.926	0.926	0.926	0.926	0.926	0.931	0.931	0.931	0.931	0.944	0.944	0.944	0.944	0.970	0.970	0.970	0.970
WIN(4,3)	0.873	0.873	0.873	0.873	0.930	0.930	0.930	0.930	0.957	0.957	0.957	0.957	0.973	0.973	0.973	0.973	0.984	0.984	0.984	0.984	
WIN(4,4)	0.873	0.873	0.873	0.873	0.936	0.936	0.936	0.936	0.954	0.954	0.954	0.954	0.959	0.959	0.959	0.959	0.968	0.968	0.968	0.968	
WIN(4,5)	0.873	0.873	0.873	0.873	0.873	0.934	0.934	0.934	0.934	0.949	0.949	0.949	0.949	0.957	0.957	0.957	0.957	0.975	0.975	0.975	0.975
WIN(6,2)	0.742	0.749	0.753	0.631	0.906	0.909	0.910	0.906	0.959	0.959	0.960	0.961	0.979	0.980	0.980	0.980	0.988	0.989	0.989	0.989	
WIN(6,3)	0.859	0.859	0.859	0.859	0.881	0.881	0.881	0.881	0.907	0.909	0.909	0.912	0.949	0.951	0.951	0.953	0.978	0.979	0.979	0.980	
WIN(6,4)	0.855	0.855	0.855	0.857	0.915	0.916	0.917	0.919	0.948	0.949	0.949	0.950	0.971	0.972	0.972	0.973	0.986	0.986	0.986	0.986	
WIN(6,5)	0.863	0.863	0.864	0.864	0.896	0.897	0.897	0.903	0.937	0.938	0.938	0.943	0.973	0.974	0.974	0.975	0.988	0.988	0.988	0.988	
WIN(8,2)	0.330	0.586	0.589	0.593	0.805	0.848	0.841	0.796	0.949	0.954	0.955	0.931	0.978	0.979	0.980	0.979	0.988	0.988	0.989	0.989	
WIN(8,3)	0.720	0.732	0.734	0.750	0.887	0.895	0.896	0.902	0.950	0.953	0.954	0.956	0.978	0.979	0.979	0.979	0.989	0.989	0.989	0.989	
WIN(8,4)	0.829	0.831	0.831	0.830	0.865	0.876	0.880	0.884	0.944	0.950	0.951	0.952	0.978	0.980	0.980	0.981	0.989	0.990	0.990	0.990	
WIN(8,5)	0.855	0.857	0.857	0.859	0.913	0.914	0.914	0.915	0.953	0.955	0.956	0.958	0.978	0.979	0.979	0.980	0.989	0.989	0.989	0.989	
WIN(10,2)	0.113	0.469	0.524	0.583	0.486	0.733	0.767	0.775	0.894	0.915	0.924	0.902	0.972	0.975	0.975	0.965	0.987	0.988	0.988	0.987	
WIN(10,3)	0.736	0.753	0.763	0.686	0.915	0.920	0.922	0.925	0.965	0.967	0.967	0.968	0.982	0.982	0.983	0.983	0.989	0.990	0.990	0.990	
WIN(10,4)	0.728	0.758	0.764	0.779	0.898	0.910	0.913	0.918	0.963	0.965	0.966	0.967	0.983	0.983	0.983	0.983	0.990	0.990	0.990	0.990	
WIN(10,5)	0.791	0.792	0.792	0.792	0.881	0.900	0.901	0.905	0.960	0.964	0.964	0.965	0.982	0.983	0.983	0.983	0.989	0.990	0.990	0.990	
WAV(db1,1)	0.754	0.760	0.762	0.770	0.920	0.923	0.923	0.926	0.964	0.965	0.966	0.967	0.981	0.981	0.981	0.982	0.988	0.989	0.989	0.989	
WAV(db1,2)	0.732	0.752	0.756	0.767	0.912	0.918	0.920	0.924	0.962	0.964	0.965	0.966	0.981	0.981	0.981	0.982	0.989	0.989	0.989	0.989	
WAV(db1,3)	0.728	0.753	0.758	0.771	0.914	0.921	0.922	0.926	0.964	0.965	0.966	0.967	0.981	0.981	0.982	0.982	0.989	0.989	0.989	0.989	
WAV(db4,1)	0.236	0.581	0.593	0.596	0.795	0.839	0.831	0.801	0.947	0.952	0.953	0.940	0.978	0.979	0.979	0.979	0.988	0.988	0.988	0.988	
WAV(db4,2)	0.601	0.666	0.690	0.655	0.910	0.920	0.920	0.921	0.965	0.968	0.969	0.969	0.982	0.983	0.983	0.983	0.989	0.990	0.990	0.990	
WAV(db4,3)	0.725	0.739	0.738	0.744	0.923	0.930	0.930	0.931	0.967	0.969	0.969	0.970	0.982	0.983	0.983	0.983	0.989	0.990	0.990	0.990	
WAV(coif1,1)	0.714	0.733	0.738	0.645	0.907	0.912	0.914	0.909	0.963	0.965	0.965	0.966	0.981	0.982	0.982	0.982	0.989	0.989	0.989	0.989	
WAV(coif1,2)	0.741	0.761	0.763	0.766	0.923	0.931	0.931	0.933	0.968	0.971	0.971	0.971	0.983	0.984	0.984	0.984	0.990	0.990	0.990	0.990	
WAV(coif1,3)	0.757	0.775	0.777	0.777	0.931	0.937	0.937	0.938	0.970	0.972	0.972	0.972	0.983	0.984	0.984	0.984	0.990	0.990	0.990	0.990	
WAV(sym2,1)	0.714	0.734	0.739	0.654	0.908	0.913	0.914	0.915	0.963	0.965	0.965	0.966	0.981	0.982	0.982	0.982	0.989	0.989	0.989	0.989	
WAV(sym2,2)	0.742	0.760	0.762	0.766	0.923	0.930	0.931	0.932	0.968	0.970	0.971	0.971	0.983	0.984	0.984	0.984	0.990	0.990	0.990	0.990	
WAV(sym2,3)	0.759	0.776	0.777	0.777	0.930	0.936	0.936	0.937	0.969	0.971	0.971	0.972	0.983	0.984	0.984	0.984	0.990	0.990	0.990	0.990	
WAV(rbio2,2,1)	0.725	0.740	0.744	0.678	0.912	0.916	0.917	0.919	0.964	0.965	0.965	0.966	0.981	0.981	0.981	0.982	0.988	0.989	0.989	0.989	
WAV(rbio2,2,2)	0.749	0.763	0.765	0.769	0.927	0.932	0.932	0.933	0.968	0.969	0.969	0.970	0.982	0.983	0.983	0.983	0.989	0.989	0.989	0.989	
WAV(rbio2,2,3)	0.771	0.780	0.780	0.784	0.931	0.935	0.935	0.936	0.969	0.970	0.970	0.970	0.982	0.983	0.983	0.983	0.989	0.989	0.989	0.989	
WAV(rbio2,8,1)	0.000	0.520	0.555	0.581	0.045	0.704	0.746	0.768	0.666	0.868	0.892	0.895	0.959	0.964	0.966	0.959	0.985	0.986	0.986	0.985	
WAV(rbio2,8,2)	0.017	0.546	0.582	0.608	0.869	0.889	0.892	0.859	0.962	0.965	0.966	0.966	0.981	0.982	0.982	0.989	0.989	0.989	0.989		
WAV(rbio2,8,3)	0.608	0.683	0.678	0.708	0.918	0.925	0.925	0.926	0.964	0.967	0.967	0.967	0.980	0.982	0.982	0.988	0.989	0.989	0.989		
WAV(rbio6,8,1)	0.000	0.500	0.544	0.582	0.045	0.687	0.734	0.772	0.670	0.860	0.887	0.899	0.962	0.967	0.968	0.963	0.986	0.987	0.987	0.986	
WAV(rbio6,8,2)	0.009	0.541	0.584	0.613	0.861	0.884	0.890	0.863	0.964	0.968	0.968	0.968	0.982	0.983	0.983	0.989	0.989	0.990	0.990		
WAV(rbio6,8,3)	0.582	0.665	0.667	0.713	0.923	0.930	0.930	0.930	0.967	0.969</td											

Chapter 5

Isotropic and anisotropic filtering norms

In this chapter, I generalize the TV minimization algorithm by substituting the TV cost function by a generic filtering cost function. The proposed method is referred as the *filtering norm minimization*. The method follows the same principles of the pre-filtering CS methods, *i.e.* filtered versions of the signals generate better signal reconstructions (using CS methods) because they are usually sparser than the original signals. Differently from pre-filtering approaches, in filtering norms the filter operations are embedded in the minimization cost functions (*e.g.* in the TV minimization). I also implement isotropic and anisotropic forms (and their linear combinations) of the filtering cost functions. Our implementation is based on NESTA, which incorporates convex approximations of the isotropic and anisotropic filtering norms.

The proposed method tackles specific problems with a specific setting of parameters. For instance, using filtering norm minimization with filters with 2 vanishing moments allows to reconstruct images composed of smooth regions with better quality than with TV minimization. Also, combinations of specific approaches can be implemented. For example, the algorithm can use filters that are good for one specific type of reconstruction, while other filters (or combinations of several filters) can be used for other types of reconstruction.

5.1 NESTA brief recap

First, a brief summary of NESTA algorithm is given, focusing in the parts that are modified and directly used for the proposed method. As detailed in this previous chapter, NESTA is a first order algorithm that consists of 4 iterative steps. From the current

solution x_k , it iterates to find better solutions that minimize a function f . A description of the NESTA algorithm is given below.

Algorithm 2 Description of NESTA.

Input: x_0, μ

Output: x_n

```

for  $k \geq 0$ , until a given criteria do
2:   Compute  $\nabla f_\mu(x_k)$ 
 $y_k = \operatorname{argmin}_x \frac{L_\mu}{2} \|x_k - x\|_2^2 + \langle \nabla f_\mu(x_k), x - x_k \rangle$ , subject to  $\|b - \mathbf{M}x\|_2 \leq \epsilon$ 
4:    $z_k = \operatorname{argmin}_x L_\mu \|x_k - x_0\|_2^2 + \langle \sum_{i=0}^k \frac{k+1}{2} \nabla f(x_i), x - x_k \rangle$ , subject to  $\|b - \mathbf{M}x\|_2 \leq \epsilon$ 
 $x_k = \frac{2}{k+3}y_k + (1 - \frac{2}{k+3})z_k$ 
6: end for

```

To solve the ℓ_1 minimization problem, the authors used the approximation to the ℓ_1 function as:

$$f_\mu(x) = \max \langle u, x \rangle - \mu p_d(u) \quad (5.1)$$

$$u \in Q_d = \{u : \|u\|_\infty \leq 1\},$$

with the smoothing term given by $p_d(u) = \frac{1}{2}\|u\|_2^2$. This approximation is strongly convex and Lipschitz, with Lipschitz constant L_μ/μ . Its gradient is a known function, given by the following equation:

$$\nabla f_\mu(x(i)) = \begin{cases} \mu^{-1}x(i), & \text{if } |x(i)| < \mu \\ \operatorname{sgn}(x(i)), & \text{otherwise.} \end{cases} \quad (5.2)$$

Notice that the method could easily be adapted to handle other minimization problems by changing the function f . One of these problems is the TV minimization. The operators that apply the horizontal and vertical finite differences are:

$$D_h(x) = x(i+1, j) - x(i, j) \quad (5.3)$$

$$D_v(x) = x(i, j+1) - x(i, j),$$

and the isotropic TV minimization can be defined as:

$$\|x\|_{TV} := \|\mathbf{D}x\|_2, \quad \mathbf{D}x = \begin{bmatrix} D_h(x) \\ D_v(x) \end{bmatrix}. \quad (5.4)$$

The isotropic TV norm approximation used on NESTA is expressed in the following form:

$$\|x\|_{TV} = \max \langle u, \mathbf{D}x \rangle, \quad (5.5)$$

$$u \in Q_d$$

where Q_d is $\{u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} : u_1^2 + u_2^2 \leq 1\}$ and the added smoothing term is $p_d(u) = \frac{1}{2}\|u\|_2^2$. The TV norm approximation that NESTA minimizes is:

$$\|x\|_{TV} \approx f_\mu = \max_{u \in Q_d} \langle u, \mathbf{D}x \rangle - \frac{\mu}{2}\|u\|_2^2, \quad (5.6)$$

and its gradient is given by:

$$\nabla f_\mu = \begin{cases} \mu^{-1} \mathbf{D}^H \mathbf{D}x, & \text{if } \|\mathbf{D}x\| < \mu \\ \|\mathbf{D}x\|_2^{-1} \mathbf{D}^H \mathbf{D}x, & \text{otherwise} \end{cases}. \quad (5.7)$$

5.2 Filtering Norm Minimization

In this section, I present the method proposed in this chapter. Notice that Equation 5.3 can be expressed as:

$$\begin{aligned} D_h(x) &= x \circledast \begin{bmatrix} 1 & -1 \end{bmatrix}, \\ D_v(x) &= x \circledast \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \end{aligned} \quad (5.8)$$

where \circledast denotes the 2D convolution. In other words, the TV norm in Equation 5.4 is the norm of the filtered versions of x , using 2D filters with coefficients $[1 \ -1]$ and $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$. Although the TV minimization reconstructs non-sparse signals, the method requires the signal (usually 2D signals) to be approximately piece-wise constant, which limits the range of applications. In the work presented in this chapter, this method is generalized, making it possible to minimize the norm of filtered versions of a signal by using any 1D or 2D filter. Therefore, I use the ensemble of filters that best sparsifies the signal to generate a good reconstruction for a given application.

Suppose we have a set of filters h_1, h_2, \dots, h_n (1D or 2D), we can define the matrix \mathbf{H} as the vertical concatenation of filtering operations (similar to Equation 5.3 and Equation 5.8, but for any set of filters):

$$\mathbf{H}x = \begin{bmatrix} x \circledast h_1 \\ x \circledast h_2 \\ \vdots \\ x \circledast h_n \end{bmatrix}, \quad (5.9)$$

where \mathbf{H} is the vertical concatenation of convolutional Toeplitz matrices [86], corresponding to each filter h_k . In section 5.2.4, more details of \mathbf{H} for the 1D and 2D cases are presented.

Let us define the isotropic filtering norm (iFN) minimization problem, with quadratic constraints, by the foollowing equation:

$$\begin{aligned} & \operatorname{argmin}_x \text{iFN}(x, h) \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon. \end{aligned} \quad (5.10)$$

The anisotropic filtering norm (aFN) minimization is defined as:

$$\begin{aligned} & \operatorname{argmin}_x \text{aFN}(x, h) \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon, \end{aligned} \quad (5.11)$$

where, for a list of filters $h = h_1, h_2, \dots, h_n$, the $\text{iFN}(x, h)$ and $\text{aFN}(x, h)$ are given by:

$$\text{iFN}(x, h) := \|\mathbf{H}x\|_2 \quad \text{and} \quad \text{aFN}(x, h) := \|\mathbf{H}x\|_1. \quad (5.12)$$

In the next sections, I present the NESTA implementations of the isotropic and anisotropic filtering norm minimizations with quadratic constraints.

5.2.1 NESTA Isotropic Filtering Norm

The TV model was first formulated in the literature as an isotropic model [87, 88, 11], using the ℓ_2 norm of finite differences. Our first approach for filtering norms was using the same isotropic model based on NESTA TV minimization, but using different filter banks. To adapt NESTA TV for minimizing the isotropic filtering norm (the problem in Equation 5.10) the cost function must be smooth. Since the filtering norm iFN in Equation 5.12 is generally not smooth, I use an approximation of the filtering norm with the same smoothing term of the NESTA TV minimization of Equation 5.6. So, the approximation of $\text{iFN}(x, h)$ is given by:

$$\begin{aligned} \text{iFN}(x, h) \approx f_\mu &= \max \langle u, \mathbf{H}x \rangle - \frac{\mu}{2} \|u\|_2^2. \\ u &\in Q_d \end{aligned} \quad (5.13)$$

Here, the dual feasible set Q_d is $\{u = [u_1, u_2, \dots, u_n]^T : \sum_{i=1}^n u_i^2 \leq 1\}$, which is selected to generalize the term Q_d in Equation 5.6 for n filters (instead of 2). The f_μ gradient for the filtering norm minimization is given by:

$$\nabla f_\mu = \mathbf{H}^H \begin{cases} \mu^{-1} \mathbf{H}x, & \text{if } \|\mathbf{H}x\|_2 < \mu \\ \|\mathbf{H}x\|_2^{-1} \mathbf{H}x, & \text{otherwise.} \end{cases} \quad (5.14)$$

The isotropic filtering norm was implemented based on the NESTA algorithm. After

building the matrix \mathbf{H} from the filters, the problem is solved using Algorithm 1, but substituting ∇f_μ in line 2 by the term given by Equation 5.14.

The problem in Equation 5.10 refers to the synthesis formulation of the isotropic filtering norm. However, the analysis formulation can also be solved with similar approach. By substituting x by $\mathbf{W}x$ on Equation 5.14, the gradient of the cost function for the analysis isotropic filtering norm problem can be obtained by:

$$\begin{aligned} & \operatorname{argmin}_x \text{iFN}(\mathbf{W}x, h) \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon. \end{aligned} \quad (5.15)$$

5.2.2 NESTA Anisotropic Filtering Norm

Anisotropic TV models can be used in several image processing applications [65, 67]. The anisotropic TV regularization function penalizes noise more than the isotropic TV [68]. Therefore, it is important to develop an anisotropic formulation for the filtering norm. As \mathbf{H} is not invertible, the anisotropic filtering norm minimization in Equation 5.11 must be solved as an analysis problem, as given by:

$$\begin{aligned} & \operatorname{argmin}_x \|\mathbf{H}x\|_1 \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon. \end{aligned} \quad (5.16)$$

This problem is solved by using the analysis formulation of NESTA ℓ_1 . A smooth and convex approximation of $\text{aFN}(x, h)$ can be obtained by substituting x by $\mathbf{H}x$ in Equation 5.1 and Equation 5.2, resulting in the following approximation:

$$\begin{aligned} \text{aFN}(x, h) & \approx f_\mu(x) = \max \langle u, \mathbf{H}x \rangle - \mu \frac{1}{2} \|u\|_2^2 \\ u & \in Q_d = \{u : \|u\|_\infty \leq 1\}. \end{aligned} \quad (5.17)$$

This function is also Lipschitz with constant L_μ/μ . Its gradient is given by:

$$\nabla f_\mu(x[i]) = \begin{cases} \mu^{-1}(\mathbf{H}x)(i), & \text{if } |(\mathbf{H}x)(i)| < \mu \\ \|\mathbf{H}x\|_1^{-1}(\mathbf{H}x)(i), & \text{otherwise.} \end{cases} \quad (5.18)$$

For solving the anisotropic analysis problem, as given by:

$$\begin{aligned} & \operatorname{argmin}_x \text{aFN}(\mathbf{W}x, h) \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon, \end{aligned} \quad (5.19)$$

the implementation composes the calculated \mathbf{H} with the provided transformation \mathbf{W} , *i.e.* I use $\mathbf{W} \cdot \mathbf{H}$ instead of \mathbf{W} in Equation 5.18.

5.2.3 NESTA Isotropic + Anisotropic Filtering Norm

Several solutions to reconstruct signals from partial Fourier data rely on models that combine isotropic and anisotropic approaches. Some models use positive weights [71, 89, 90, 91, 67], while others use negative weights for the isotropic model [72, 92]. Recently, good results were obtained using a weight equals to -0.5 for the isotropic term and a weight equals to 1.0 for the anisotropic term [93, 94].

To further increase the range of applications of our method, I implemented a general linear combination of the isotropic and anisotropic filter norms. Since the gradient operation is a linear operation, these linear combinations are easily implemented using the following expression:

$$\begin{aligned} & \operatorname{argmin}_x \alpha_1 \text{aFN}(x, h) + \alpha_2 \text{iFN}(x, h) \\ & \text{subject to } \|b - \mathbf{M}x\|_2 \leq \epsilon, \end{aligned} \quad (5.20)$$

where α_1 and α_2 are real values. The function f_μ , corresponding to $\alpha_1 \text{aFN}(x, h) + \alpha_2 \text{iFN}(x, h)$, is calculated as a linear combination of Equation 5.13 and Equation 5.17. The gradient ∇f_μ , which is a linear combination of Equation 5.14 and Equation 5.18, is calculated in each NESTA iteration. The estimations of x_k are computed as described in Algorithm 1. I use the notation $\text{iaFN}(x, h)$ to refer to $\text{aFN}(x, h) + \text{iFN}(x, h)$.

5.2.4 The matrix \mathbf{H}

Note that the filtering norms are modifications of the original NESTA selecting specific gradient functions. The common point of these functions is that they depend on the matrix \mathbf{H} , specified in more details in this section. The matrix \mathbf{H} is formed by the vertical concatenation of the discrete convolution Toeplitz matrices [86], corresponding to each filter h_k . For the 1D case, the rows of the convolution matrix is formed by the coefficients of the corresponding filters. For the 2D case, the coefficients of the filters are padded with zeros to reach the image size. Then, the result is vectorized to form the first row of the matrix. The subsequent rows of the matrix are shifted versions of the first row. The final matrix \mathbf{H} is the vertical concatenation of the matrices for each filter h_k . \mathbf{H} is implemented as a sparse matrix, which is composed mostly of zeros. This speeds-up the matrix multiplication operation and saves storage.

As an illustrative example for the 1D case, for a signal x with 5 entries, and for the filters: $h^1 = [a_1 \ a_2]$, and $h^2 = [b_1 \ b_2]$, the application of the matrix \mathbf{H} to x is:

$$\mathbf{H}x = \begin{bmatrix} a_2 & a_1 & 0 & 0 & 0 \\ 0 & a_2 & a_1 & 0 & 0 \\ 0 & 0 & a_2 & a_1 & 0 \\ 0 & 0 & 0 & a_2 & a_1 \\ b_2 & b_1 & 0 & 0 & 0 \\ 0 & b_2 & b_1 & 0 & 0 \\ 0 & 0 & b_2 & b_1 & 0 \\ 0 & 0 & 0 & b_2 & b_1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} x_1a_2 + x_2a_1 \\ x_2a_2 + x_3a_1 \\ x_3a_2 + x_4a_1 \\ x_4a_2 + x_5a_1 \\ x_1b_2 + x_2b_1 \\ x_2b_2 + x_3b_1 \\ x_3b_2 + x_4b_1 \\ x_4b_2 + x_5b_1 \end{bmatrix} \quad (5.21)$$

For an example of the 2D case, for a image x have 3 rows and 4 columns, for the filter bank $h^1 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, $h^2 = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$ and $h^3 = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$, the application of the matrix \mathbf{H} to x is:

$$\mathbf{H}x = \begin{bmatrix} a_{22} & a_{21} & 0 & 0 & a_{12} & a_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{22} & a_{21} & 0 & 0 & a_{12} & a_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{22} & a_{21} & 0 & 0 & a_{12} & a_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{22} & a_{21} & 0 & 0 & a_{12} & a_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{22} & a_{21} & 0 & 0 & a_{12} & a_{11} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{22} & a_{21} & 0 & 0 & a_{12} & a_{11} & 0 \\ b_{22} & b_{21} & 0 & 0 & b_{12} & b_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_{22} & b_{21} & 0 & 0 & b_{12} & b_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{22} & b_{21} & 0 & 0 & b_{12} & b_{11} & 0 & 0 & b_{12} & b_{11} & 0 \\ 0 & 0 & 0 & 0 & b_{22} & b_{21} & 0 & 0 & b_{12} & b_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_{22} & b_{21} & 0 & 0 & b_{12} & b_{11} & 0 \\ c_{22} & c_{21} & 0 & 0 & c_{12} & c_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{22} & c_{21} & 0 & 0 & c_{12} & c_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_{22} & c_{21} & 0 & 0 & c_{12} & c_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{22} & c_{21} & 0 & 0 & c_{12} & c_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{22} & c_{21} & 0 & 0 & c_{12} & c_{11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{12} & c_{11} & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{31} \\ x_{32} \\ x_{33} \\ x_{34} \end{bmatrix} = \begin{bmatrix} a_{22}x_{11} + a_{21}x_{12} + a_{12}x_{21} + a_{11}x_{22} \\ a_{22}x_{12} + a_{21}x_{13} + a_{12}x_{22} + a_{11}x_{23} \\ a_{22}x_{13} + a_{21}x_{14} + a_{12}x_{23} + a_{11}x_{24} \\ a_{22}x_{21} + a_{21}x_{22} + a_{12}x_{31} + a_{11}x_{32} \\ a_{22}x_{22} + a_{21}x_{23} + a_{12}x_{32} + a_{11}x_{33} \\ a_{22}x_{23} + a_{21}x_{24} + a_{12}x_{33} + a_{11}x_{34} \\ b_{22}x_{11} + b_{21}x_{12} + b_{12}x_{21} + b_{11}x_{22} \\ b_{22}x_{12} + b_{21}x_{13} + b_{12}x_{22} + b_{11}x_{23} \\ b_{22}x_{13} + b_{21}x_{14} + b_{12}x_{23} + b_{11}x_{24} \\ b_{22}x_{21} + b_{21}x_{22} + b_{12}x_{31} + b_{11}x_{32} \\ b_{22}x_{22} + b_{21}x_{23} + b_{12}x_{32} + b_{11}x_{33} \\ b_{22}x_{23} + b_{21}x_{24} + b_{12}x_{33} + b_{11}x_{34} \\ c_{22}x_{11} + c_{21}x_{12} + c_{12}x_{21} + c_{11}x_{22} \\ c_{22}x_{12} + c_{21}x_{13} + c_{12}x_{22} + c_{11}x_{23} \\ c_{22}x_{13} + c_{21}x_{14} + c_{12}x_{23} + c_{11}x_{24} \\ c_{22}x_{21} + c_{21}x_{22} + c_{12}x_{31} + c_{11}x_{32} \\ c_{22}x_{22} + c_{21}x_{23} + c_{12}x_{32} + c_{11}x_{33} \\ c_{22}x_{23} + c_{21}x_{24} + c_{12}x_{33} + c_{11}x_{34} \end{bmatrix} \quad (5.22)$$

Notice that in 2D case, some rows are excluded. These rows calculate convolution of the image x considering elements of different rows/columns of the image. Therefore, the matrix \mathbf{H} for the 2D case is built in such way the rows/columns do not interfere in the norm calculation of the others rows/columns.

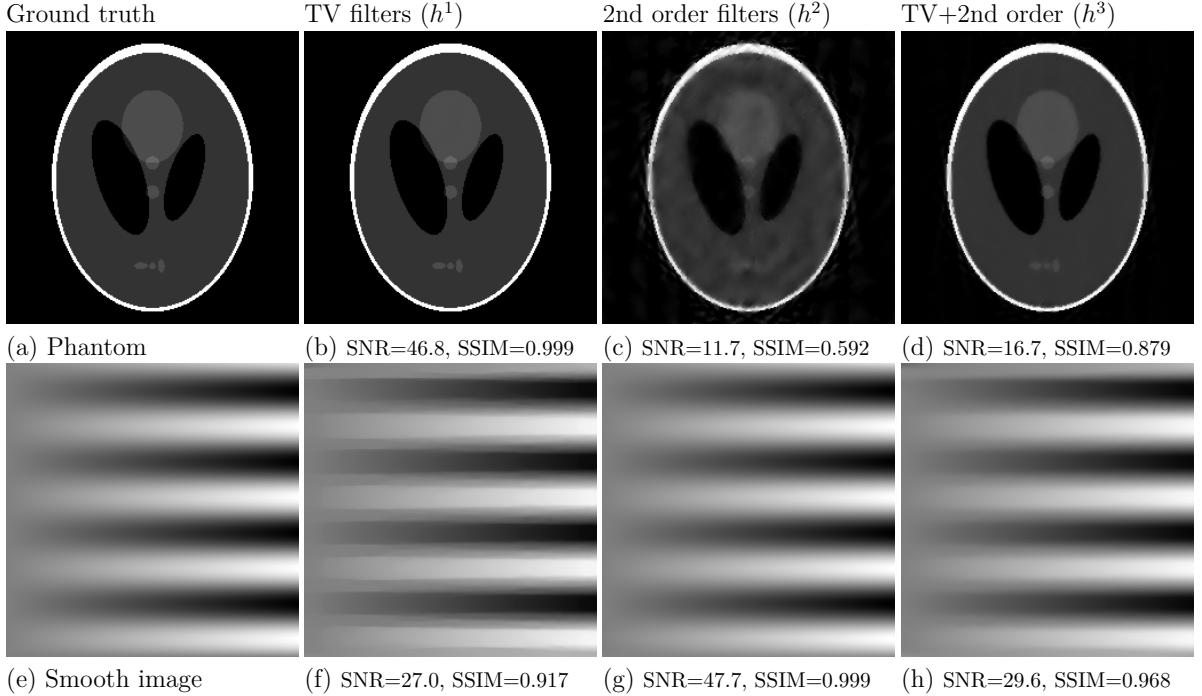


Figure 5.1: Isotropic filtering norm reconstructions using 20 radial projections.

5.3 Applications and Simulations

5.3.1 Tests Using Synthetic Images

In this section, I present a proof of concept for the proposed method using synthetic images. The simulation consists of reconstructing an image from incomplete 2D Fourier data. The magnetic resonance imaging (MRI) is our target application and, I use 20 radial projections on the 2D Fourier space approximated by a Cartesian grid. I chose 2 synthetic images and reconstruct them using the isotropic filtering norm with 3 sets of filters. One of the selected filters for this test are the horizontal and vertical first order finite differences (h^1): $h_1^1 = [1 \ -1]$ and $h_2^1 = [1 \ -1]^T$. I also test the horizontal and vertical second order finite differences filters (h^2): $h_1^2 = [1 \ -2 \ 1]$ and $h_2^2 = [1 \ -2 \ 1]^T$. While h^1 filters are more suitable to sparsify piece-wise constant images, h^2 filters have 2 vanishing moments and can sparsify smoother images. Finally, I also used a third set of filters (h^3), which corresponds to the union of the sets of filters h^1 and h^2 .

I compared the image qualities of the reconstructions using the SNR (dB) and SSIM metrics, detailed in chapter 2. The two synthetic images used in our simulations are presented in Figures 5.1(a) and (e). Figure 5.1(a) corresponds to the well known Shepp-Logan phantom [95], which is piece-wise constant that is better reconstructed by the TV minimization algorithm. The image in Figure 5.1(e) is a smooth image, whose amplitude changes linearly in the horizontal direction and according to a sinusoidal function in the

vertical direction. Figures 5.1 (b)-(d) and (f)-(h) show the reconstructions for these two images, respectively. Results show that each filter banks performs better for a specific type of image. While the TV minimization works very well for piece-wise constant images (Figure 5.1(b)), the filtering norm based on h^2 does not work as well (Figure 5.1(c)). Using both sets of filters corresponds to an intermediate result (Figure 5.1(d)). However, h^2 presents best results for the smoother sinusoidal image (Figure 5.1(g)), while the TV minimization has a lower performance (Figure 5.1(f)). Again, the combination of the two sets of filters shows an intermediate result (Figure 5.1(h)). Since synthetic images have very specific features, one can easily select the optimal set of filters to produce the best reconstruction quality. Natural images, on the other hand, have a variety of features, which may include piece-wise constant parts and smooth transitions. In the next section, I present test simulations with real MRI images.

5.3.2 Tests Using Magnetic Resonance Images

Experimental methodology

The simulation of MRI reconstruction processes are performed on the same dataset of previous chapters. The sampling pattern is also the same used in previous chapters. To reconstruct the images, three previously described cost functions models are used: isotropic (iFN), anisotropic (aFN), and a combination of isotropic and anisotropic (iaFN, with $\alpha_1 = 1$ and $\alpha_2 = 1$). A total of 8 different sets of filter banks were chosen to be used in the filtering norm reconstructions. As mentioned earlier, the filters have the goal of sparsifying the images to achieve reconstructed images with a better reconstruction quality. The filters can be classified in 3 types:

- First-order finite difference filters (FOFD): filters that sparsify piece-wise constant regions;
- Second-order finite differences filters (SOFD): filters that sparsify smooth regions;
- WIN filters: filters that are horizontal, vertical, and diagonal high-passes of order 2 (with cutoff frequencies ranging from 0.5 to 1.0), designed with the windowing method (Hamming window), as presented in chapter 3.

This specific WIN filter bank was chosen because it obtained one of the best reconstruction quality with pre-filtering. It is also a small filter bank, which corresponds to a very sparse matrix \mathbf{H} , which leads to a fast implementation.

Table 5.1 presents a summary of the parameters of these sets of filters. Note that the combination of h^1 with the cost function iFN corresponds to the isotropic TV minimization

Table 5.1: Description of the set of filter banks used on our experiment.

filter banks	Description	Type	Nof Filters	Nof coeffs.
h^1	The horizontal and vertical finite differences, used on TV minimization. It was described in section 5.1. Good for piece-wise regions.	FOFD	2	4
h^2	2D Haar high passes filter, horizontal ($h_1^2 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$), vertical ($h_2^2 = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$) and diagonal ($h_3^2 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$) first order finite differences [96].	FOFD	3	12
h^3	The horizontal and vertical second order finite differences, as described in section 5.1. Good for smooth regions.	SOFD	2	6
h^4	WIN filters from [19], <i>i.e.</i> high pass filters designed with windowing method.	WIN	3	27
h^5	Composed from filters from h^2 and h^3 .	FOFD, SOFD	5	18
h^6	Composed from filters from h^2 and h^4 .	FOFD, WIN	6	39
h^7	Composed from filters from h^3 and h^4 .	SOFD, WIN	5	33
h^8	Composed from filters from h^2 , h^3 and h^4 .	FOFD, SOFD, WIN	8	45

(NESTA TV), a very established method for image CS reconstruction. Therefore, by including it in the tests, the proposed method is compared to the state-of-the-art methods.

I simulate the measurement noise with four different levels of Gaussian noise. The levels of noise are the following: no noise, 60 dB, 40 dB, and 20 dB. The standard deviation σ of the Gaussian noise distributions is used to compute the tolerance of the search space (ϵ in 5.20). As suggested in the original NESTA paper, $\epsilon = \sqrt{m + 2\sqrt{2m}\sigma}$. Finally, to obtain an empirical trade-off between speed and reconstruction quality, I set $\mu = 10^{-4}$. I perform 26,400 image reconstructions, corresponding to testing combinations of 4 parameters for all 55 images of the dataset: 8 filter banks, 3 cost functions, 4 noise levels, and 5 total radial lines projections. I run the experiment on the same computational cluster of the experiments of previous chapters. The implementation is performed in MATLAB, based on the original NESTA code [97], modified to accommodate the methods proposed in this work. To analyze the several parameter combinations, I use an n-way analysis of variance (ANOVA) [98] to test which pairs of parameters lead to a statistically significant difference in terms of image quality. I consider a p-value of 0.05 and use both SNR and SSIM, described in the previous section, as performance quality metrics.

Results

Table 5.2 shows the results of the ANOVA test for the SNR/SSIM mean values, separated by the filter banks (left table) and the cost functions (right table). Among the filter banks,

h^8 is the filter with the best quality performance, in terms of both SNR and SSIM. The second best filter bank is h^6 , followed by h^7 and h^5 . Filters h^3 and h^4 present the worst results, while h^2 is slightly better than h^1 (the diagonal finite difference filter improves the results of the TV). Notice that filters h^1 to h^4 are single filters, while h^5 to h^8 are combinations of different types of filters. Therefore, in this experiment, combinations of different types of filters lead to better image quality results. It is worth mentioning that the combinations of WIN and FOFD filters outperform other combinations of filters. Table 5.2 (right) also shows the ANOVA tests separated by cost functions. Notice that iaFN results in significant higher average SNR and SSIM values than iFN and aFN. When comparing aFN and iFN functions, aFN presents a better SNR value than iFN, while iFN presents a better SSIM value.

Table 5.3 shows the ANOVA test for several combinations of filter banks and functions. We observe in this table that the combination of h^8 with iaFN has significant higher average SNR values than any other combination, while having the same average SSIM values than iFN(x, h^6), iFN(x, h^8), and iaFN(x, h^8). At same time, iFN with h^1 , the NESTA TV, presents significant lower average SNR and SSIM values than the filter banks with combinations of the different filter types (except for the SSIM value, when compared with aFN(x, h^5)). Additionally, the ANOVA tests for radial lines show that the number of radial projections lead to statistically significant differences in quality, *i.e.* an increase in the number of radial lines (rl) projections increases both SNR and SSIM average values. The ANOVA test for levels of noise shows a not significant difference of SNR/SSIM values for no-noise and 60 dB noise. Meanwhile, the levels of noise of 40 dB and 20 dB present a significant smaller SNR/SSIM values than no-noise and 60 dB.

Table 5.2: ANOVA test between SNR/SSIM values for the image reconstructions, isolated by filter banks (left table) and cost functions (right table). Each cell has a pair of ‘1’, ‘0’, ‘-1’, with the first value corresponding to the ANOVA test results for SNR and the second for SSIM. The ‘1’ value means that the mean value of the corresponding row parameter is statistically higher ($p < 0.1$) than the corresponding column parameter. The ‘-1’ value means that the value to the corresponding row parameter is statistically lower than the one corresponding to the column parameter. Finally, the ‘0’ value means that there is no statistical difference between the two values. Green cells corresponds to ‘1’ to both SNR and SSIM, red cells to both ‘0’ or ‘-1’, and blue cells to one of the metrics corresponding to ‘1’.

	h^1	h^2	h^3	h^4	h^5	h^6	h^7	h^8
h^1	0/0	-1/0	1/1	1/1	-1/-1	-1/-1	-1/-1	-1/-1
h^2	1/0	0/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1
h^3	-1/-1	-1/-1	0/0	0/0	-1/-1	-1/-1	-1/-1	-1/-1
h^4	-1/-1	-1/-1	0/0	0/0	-1/-1	-1/-1	-1/-1	-1/-1
h^5	1/1	1/0	1/1	1/1	0/0	-1/-1	0/-1	-1/-1
h^6	1/1	1/1	1/1	1/1	1/1	0/0	1/1	-1/0
h^7	1/1	1/1	1/1	1/1	0/1	-1/-1	0/0	-1/-1
h^8	1/1	1/1	1/1	1/1	1/1	1/0	1/1	0/0

	iFN	aFN	iaFN
iFN	0/0	-1/1	-1/-1
aFN	1/-1	0/0	-1/-1
iaFN	1/1	1/1	0/0

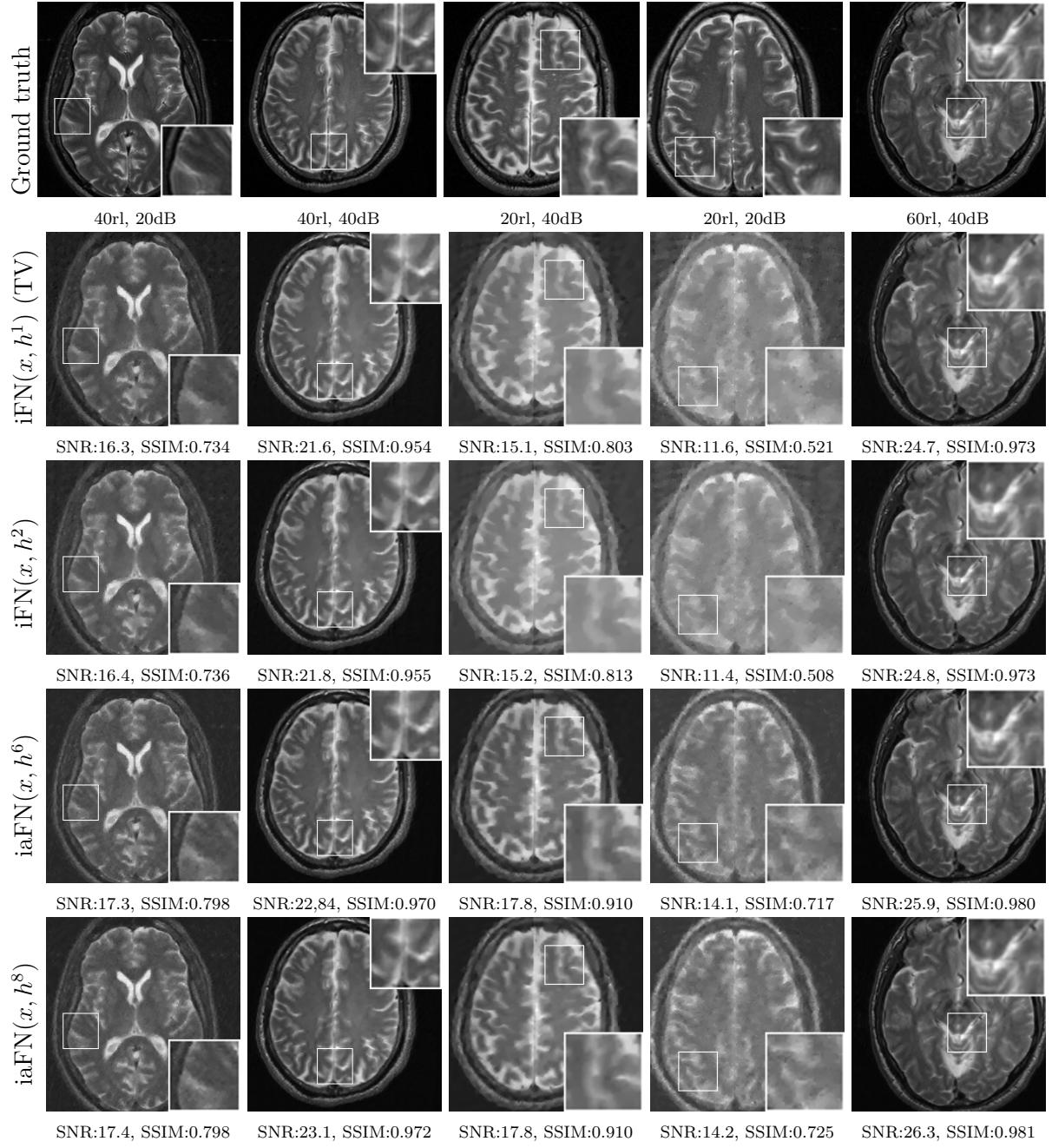


Figure 5.2: MRI reconstructions. 1st row: ground truth images. 2nd to 5-th rows: TV minimization; $iFN(x, h^2)$; $iaFN(x, h^6)$; and $iaFN(x, h^8)$. 1st to 5th columns: 40 radial lines and 20dB of added noise; 40 radial lines and 40 dB of noise; 20 radial lines and 40 dB of noise; 20 radial lines and 20 dB of noise; and 60 radial lines and 40 dB of noise.

Figure 5.2 shows qualitative results for 4 different reconstruction scenarios, with the following combinations of functions and filter banks: $iFN(x, h^1)$ (NESTA TV) as a comparison point; $iFN(x, h^2)$ to show the difference of adding the diagonal FOFD filter to NESTA TV; $iaFN(x, h^6)$ and $iaFN(x, h^8)$ that are the two best performing methods. I included a crop and a zoom of all images for better visualization. We can observe that images reconstructed using $iaFN(x, h^6)$ and $iaFN(x, h^8)$ present more details and more contrast than images reconstructed with $iFN(x, h^1)$ and $iFN(x, h^2)$. This is easier to visualize for the images in the 3rd and 4th columns of Figure 5.2, which correspond to the scenarios with the lowest number of radial lines. For these cases, when compared to the isotropic TV minimization, the model $iaFN(x, h^8)$ improved up to 2.8 dB in terms of SNR and 0.204 in terms of SSIM. For a higher number of radial lines and a small amount of noise, the differences in performance are smaller. Nevertheless, $iaFN(x, h^8)$ always presents the best results in this experiment, in terms of qualitative visual quality, SNR, and SSIM.

Figure 5.3 shows the average SNR and SSIM for a selection of 4 filter banks of Figure 5.2, considering all functions and the highest (20 Db) and lowest (no noise) levels of noise. We can observe that filters h^6 and h^8 outperform (in terms of SNR and SSIM) h^1 and h^2 in most scenarios, specially for fewer number of measurements with noise. Also, using the aFN and iaFN favored h^6 and h^8 the most, specially for more measurements and less noise. Figure 5.4 compares the NESTA TV minimization $iFN(x, h^1)$ and the best performing model, $iaFN(x, h^8)$, for every tested scenario (all levels of added noise and number of radial lines). In terms of SNR, $iaFN(x, h^8)$ significantly outperforms (with 95% level of confidence) the state-of-the-art method for every scenario. In terms of SSIM, it outperforms the state-of-the-art method for almost all scenarios.

Computational Time Analysis

Finally, I measured and compared the computational running time. I believe this metric is worth mentioning, given that the matrix \mathbf{H} can heavily impact the computational time of $\nabla f_\mu(x_k)$. More specifically, \mathbf{H} is an overcomplete transformation matrix that gets bigger with the number of filters used by the method. Fortunately, the implementation using sparse matrix representation mitigates the effect of the computational time, as long the matrix is sufficiently sparse. However, every (non border) column of the matrix is N -sparse, where N is the total number of coefficients of the filter bank. Figure 5.5 shows the reconstruction times versus the total number of coefficients of the corresponding filter banks, considering all reconstructions. The figure shows a boxplot which the central bars

Table 5.3: ANOVA test between SNR/SSIM values for the image reconstructions of combinations of filter banks and cost functions. Each cell has a pair of '1', '0', '-1', with the first value corresponding to the ANOVA test results for SNR and the second for SSIM. The '1' value means that the mean value of the corresponding row parameter is statistically higher ($p < 0.1$) than the corresponding column parameter. The '-1' value means that the value to the corresponding row parameter is statistically lower than the one corresponding to the column parameter. Finally, the '0' value means that there is no statistical difference between the two values. Green cells corresponds to '1' to both SNR and SSIM, red cells to both '0' or '-1', and blue cells to one of the metrics corresponding to '1'.

		iFN								aFN								iaFN							
		h^1	h^2	h^3	h^4	h^5	h^6	h^7	h^8	h^1	h^2	h^3	h^4	h^5	h^6	h^7	h^8	h^1	h^2	h^3	h^4	h^5	h^6	h^7	h^8
iFN	h^1	0/0	-1/0	1/1	1/1	-1/-1	-1/-1	-1/-1	-1/-1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^2	1/0	0/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^3	-1/-1	0/0	1/-1	0/0	0/0	-1/-1	-1/-1	-1/-1	-1/0	1/-1	1/-1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
	h^4	-1/-1	0/0	1/-1	0/0	0/0	-1/-1	-1/-1	-1/-1	-1/0	1/-1	1/-1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
	h^5	1/1	1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
	h^6	1/1	1/1	1/1	1/1	0/0	-1/0	-1/0	-1/0	-1/0	0/1	1/1	-1/1	-1/1	-1/1	-1/1	-1/0	0/1	0/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^7	1/1	1/1	1/1	1/1	0/1	-1/-1	-1/-1	-1/-1	-1/0	0/1	1/1	-1/1	-1/1	-1/1	-1/1	-1/1	0/1	0/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^8	1/1	1/1	1/1	1/1	1/0	-1/0	-1/0	-1/0	-1/0	1/1	1/1	0/1	0/1	0/1	0/1	0/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
aFN	h^1	1/1	0/-1	1/1	1/1	0/-1	-1/1	-1/1	-1/1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^2	1/0	1/0	1/1	1/1	0/-1	-1/1	-1/1	-1/1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
	h^3	1/1	1/1	1/1	1/1	0/-1	-1/1	-1/1	-1/1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^4	1/0	1/0	1/1	1/1	0/-1	-1/1	-1/1	-1/1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^5	1/1	1/1	1/1	1/1	0/-1	-1/1	-1/1	-1/1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^6	1/1	1/1	1/1	1/1	0/-1	-1/1	-1/1	-1/1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
	h^7	1/1	1/1	1/1	1/1	0/-1	-1/1	-1/1	-1/1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
	h^8	1/1	1/1	1/1	1/1	1/0	-1/0	-1/0	-1/0	-1/0	1/1	1/1	-1/1	-1/1	-1/1	-1/1	-1/0	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
iaFN	h^1	1/0	1/0	1/1	1/1	0/0	0/-1	-1/-1	-1/-1	-1/0	1/1	1/1	-1/0	-1/-1	-1/-1	-1/-1	-1/-1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
	h^2	1/1	1/0	1/1	1/1	0/0	0/-1	-1/-1	-1/-1	-1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/-1	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
	h^3	-1/-1	1/0	1/0	1/-1	0/0	-1/-1	-1/-1	-1/-1	-1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/-1	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
	h^4	-1/-1	1/0	1/0	1/-1	0/0	-1/-1	-1/-1	-1/-1	-1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/-1	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
	h^5	1/1	1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/-1	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
	h^6	1/1	1/1	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/-1	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
	h^7	1/1	1/1	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/-1	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0
	h^8	1/1	1/1	1/1	1/1	1/0	-1/0	-1/0	-1/0	-1/0	1/1	1/1	0/0	-1/-1	-1/-1	-1/-1	-1/-1	0/0	0/0	0/0	0/0	0/0	0/0	0/0	0/0

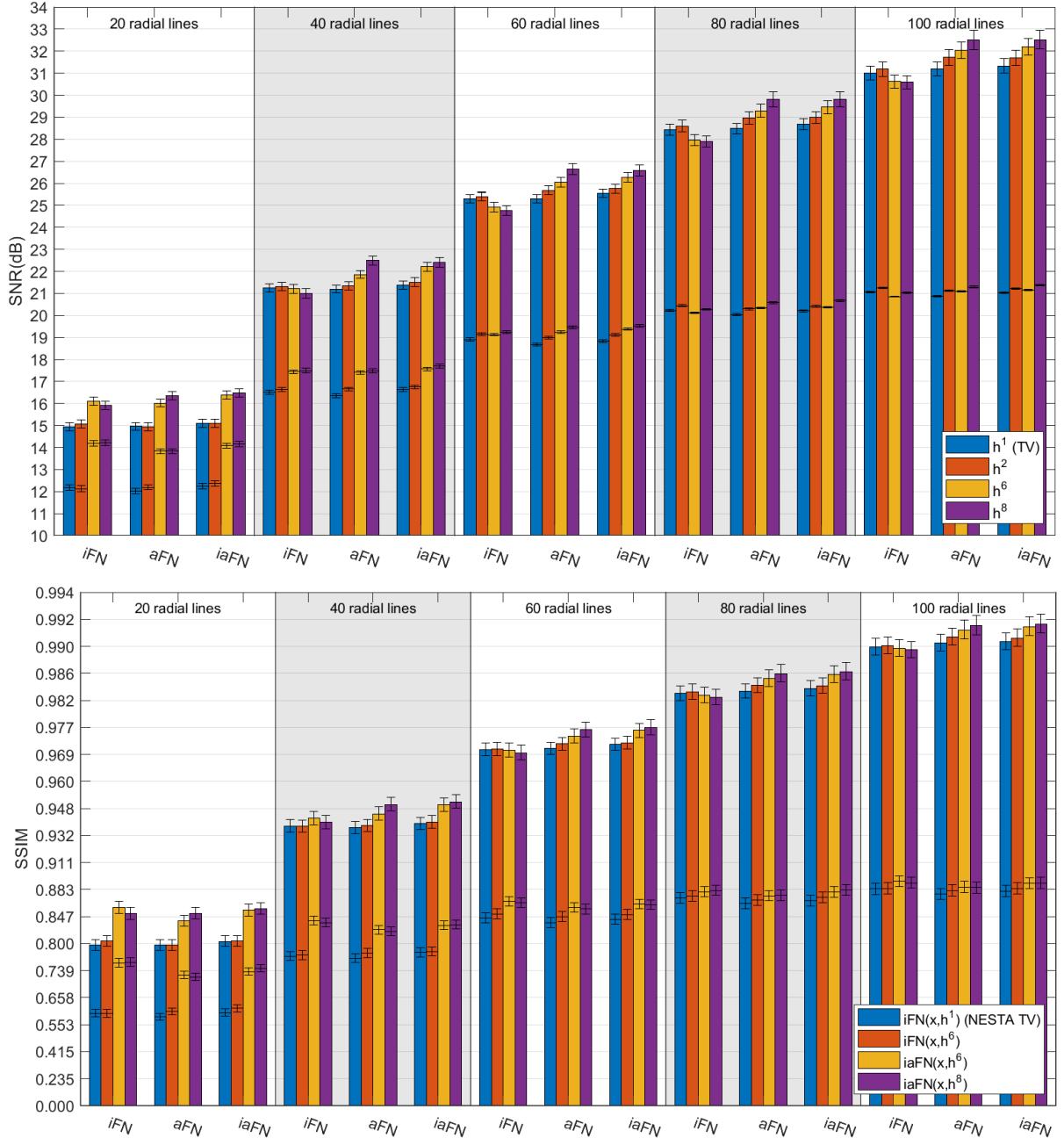


Figure 5.3: Average SNR (top) and SSIM (bottom) values, with the corresponding confidence intervals (95%), computed between the ground truth and the images reconstructed using the filtering norm with filter banks h^1, h^2, h^6 , and h^8 for the iFN , aFN and $iaFN$ models. The plot illustrates the values for all tested radial lines values, considering the no-noise (top of the bars) and the 20dB noise (middle of the bars) cases.

correspond to the median values, and the boxes correspond to the interquartile ranges¹. Notice that the median computational time, as expected, increases with the number of coefficients. The median reconstruction time using h^8 (45 coefficients) is 3.6 times slower

¹The interquartile boxplot was more appropriated to data visualization because some outliers skew the average value significantly

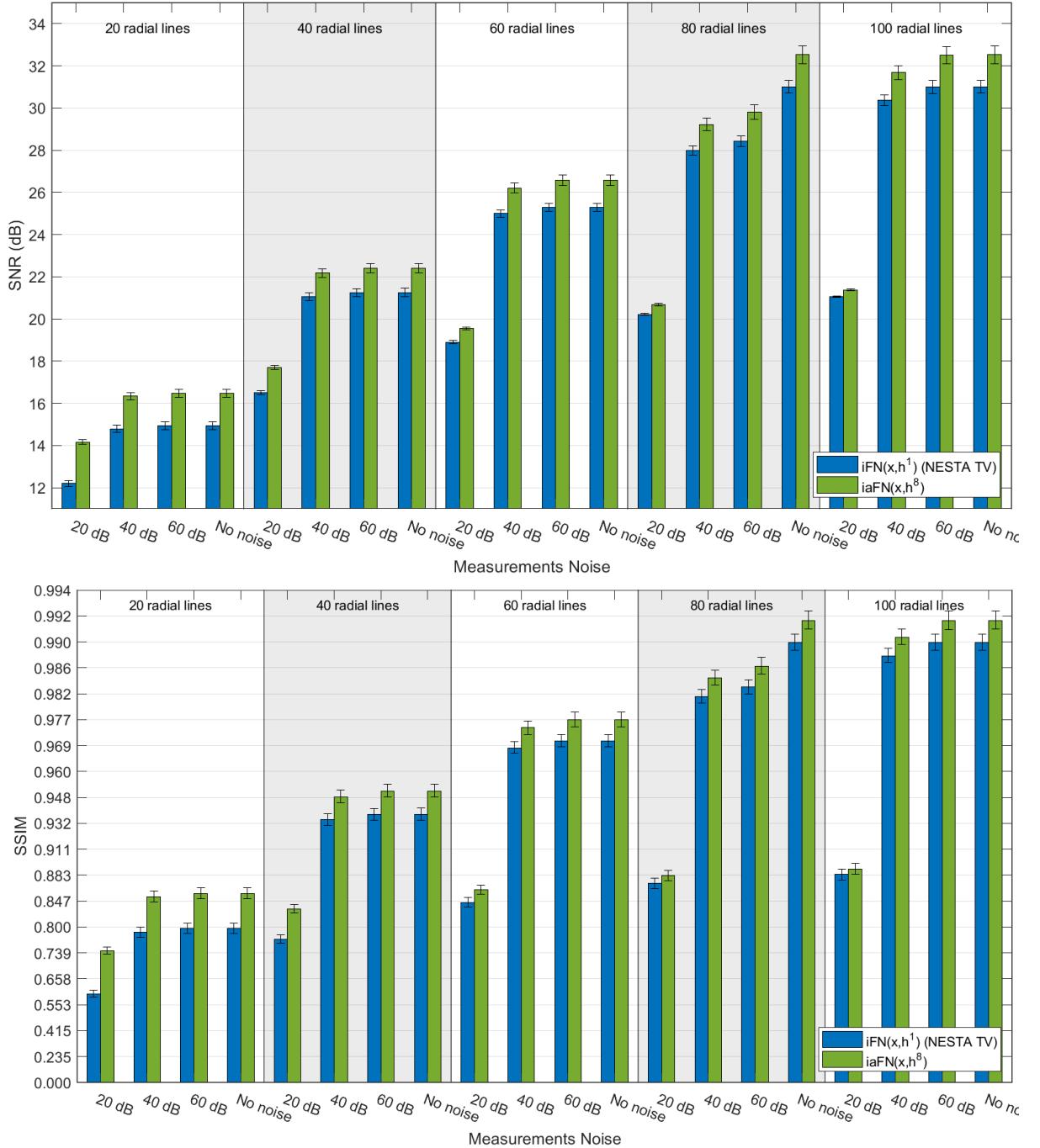


Figure 5.4: Average SNR (top) and SSIM (bottom) values, with the corresponding confidence intervals (95%), computed between the ground truth and the images reconstructed using the NESTA TV algorithm ($iFN(x, h^1)$) and the isotropic and anisotropic filtering with h^8 ($iaFN(x, h^8)$, best configuration) for the tested samplings and levels of noise.

than using the TV (4 coefficients). The filter banks with a higher number of coefficients had also a higher variance.

I also analyzed the median reconstruction time for different cost functions. In this case, the computation of $\nabla f_\mu(x_k)$ for iaFN is the sum of the $\nabla f_\mu(x_k)$ corresponding to iFN

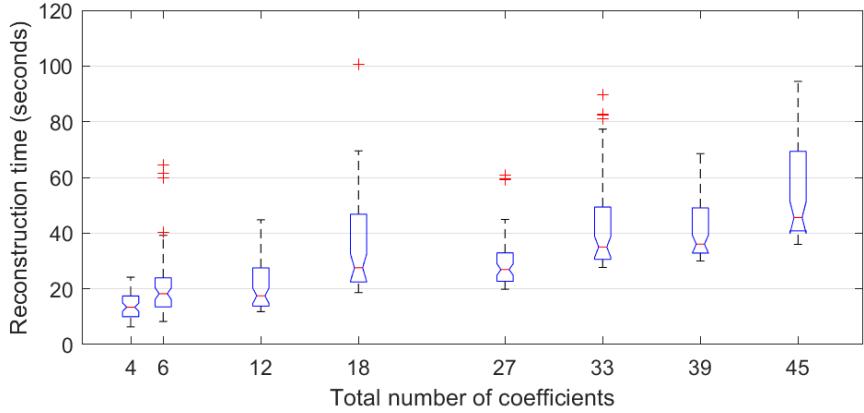


Figure 5.5: Reconstruction times(s), grouped by the total number of coefficients of the filter banks. Outliers over 120s are not shown because the interval was chosen for the best visualization.

Table 5.4: Median reconstruction time(s).

filter bank	h^1	h^2	h^3	h^4	h^5	h^6	h^7	h^8	Overall median
Total n. coeffs.	4	12	6	27	18	39	33	45	
iFN	13.2	17.7	18.5	28.5	29.2	42.8	39.3	51.4	30.8
aFN	15.3	20.2	18.5	32.8	28.8	41.6	39.0	53.5	30.6
iaFN	18.4	35.2	32.2	35.9	60.1	59.2	63.3	91.6	50.9

and the $\nabla f_\mu(x_k)$ corresponding to aFN, leading to a higher reconstruction time. Table 5.4 shows the median reconstruction time for combinations of filter banks and functions. The table also shows the overall median reconstruction time for all filter banks. If the reconstruction time is the most important factor for the application, the best cost function is iFN(x, h^1). But, if image quality is the most important factor, iaFN(x, h^8) is the best cost function. On the other hand, iaFN(x, h^6) is a good compromise because it provides a reconstruction quality that is very close to the best performing model (iaFN(x, h^8)), with around 2/3 of the reconstruction time.

5.4 Conclusions

I proposed a filtering norm minimization algorithm for compressive sensing and image reconstruction. The method is a generalization of the CS TV minimization algorithm that uses different filter banks, besides the finite differences filters used by the original algorithm. I also expanded the NESTA algorithm, which is commonly used in CS, to include isotropic and anisotropic formulations. For the reconstruction of piece-wise constant and smooth images, I obtained a good balance (in terms of image quality) when I combined first and second order finite differences filters.

In our MRI reconstruction simulations, the best results corresponded to the filter banks h^8 , which combines 8 different filters including first and second order finite differences filters, as well as filters designed with the windowing method. Results showed that the best cost function is the linear combination of isotropic and anisotropic filter norms (iaFN). When compared to the original NESTA TV algorithm ($iFN(x, h^1)$ model), the combination of the best filter banks and functions led to significant improvements in image quality, but required larger reconstruction times. Also, a higher improvement in image quality was achieved for lower sampling settings.

Chapter 6

Towards a Pre-filtering with decimation

Despite some good results for some filter banks, the pre-filtering method has some limitations. In chapter 3, the plot of sparsity versus reconstruction SNR (Figure 3.10) does not show a positive correlation between the sparsity and the image quality. When investigating this issue, I observed that filters that sparsify the signals in the space domain usually also sparsify them in the frequency domain. A possible explanation for this behavior is that very selective band-pass filters do not reconstruct the images as well as some filters with smooth frequency response. I believe that previous results can be improved if I manage to keep the sparsity on the space domain, but do not sparsify the frequency domain. To do that, one can make use of the frequency spectrum spreading, often caused by the decimation operation.

Let us consider the signal $x[n]$ to be decimated by a factor of M . The decimation operation consists of taking only the samples in $x[n]$ that are multiples of M , i.e. the signal decimated by a factor M is given by $x[Mn]$. Let the DTFT discrete time Fourier transform (DTFT) of $x[n]$ be $X_{2/\pi}(\omega)$. The decimation property states that:

$$\text{DTFT}\{x[Mn]\} = \frac{1}{M} \sum_{k=1}^{M-1} X_{2\pi} \left(\frac{\omega - 2\pi k}{M} \right). \quad (6.1)$$

As consequence, the values of $X_{2\pi}$ corresponding to frequencies below $2\pi/M$ are stretched over all the spectrum. Also, the frequencies in the intervals given by:

$$[2\pi/M, 4\pi/M], [4\pi/M, 6\pi/M], \dots, [2(M-1)\pi/M, 2M\pi/M] \quad (6.2)$$

are overlapped in the interval $[0, 2\pi/M]$, then they are stretched by a factor of M and fit the interval $[0, 2\pi]$. Therefore, a signal that is sparse in the frequency domain can

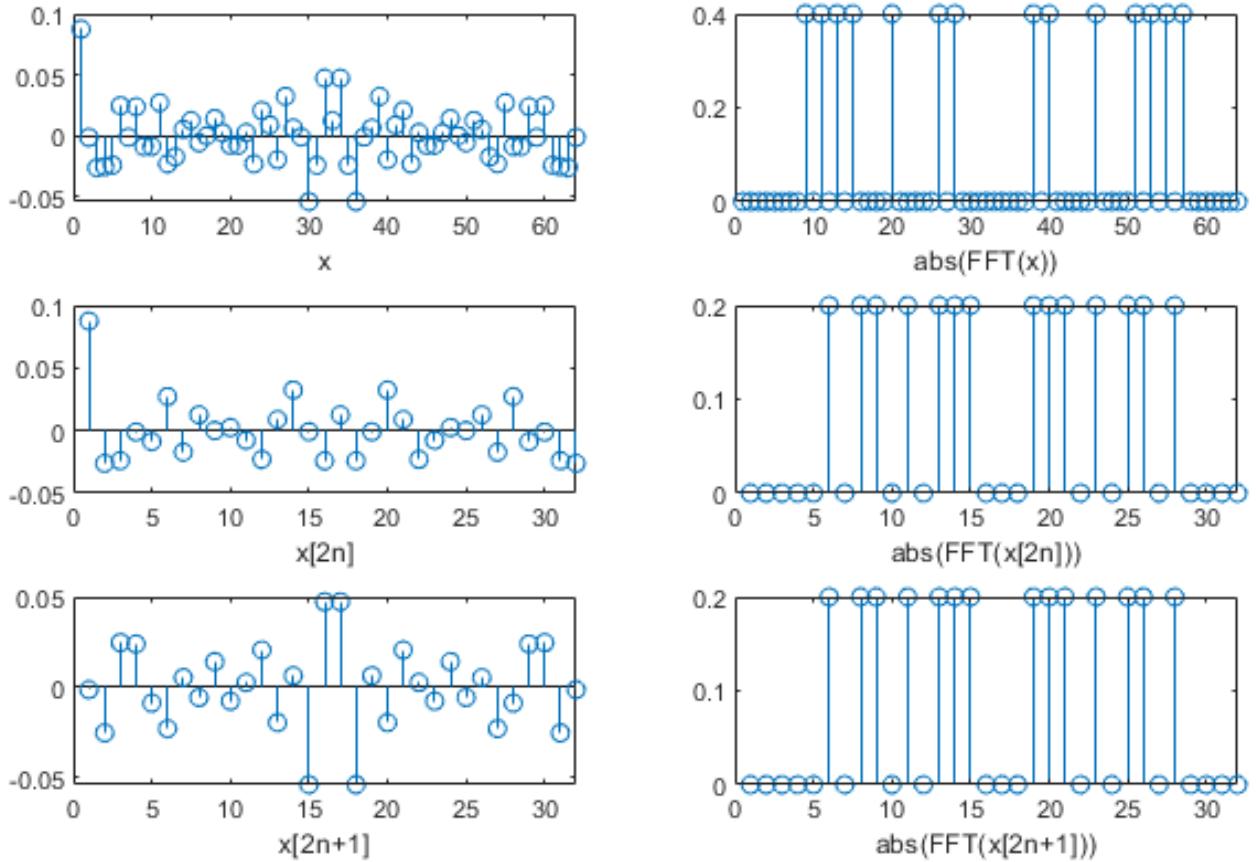


Figure 6.1: A signal x (top left), its decimation $x[2n]$ (middle left) and decimated and shifted $x[2n + 1]$ (bottom left). On the right sizes, the absolute value of the DFT of the signals in the right.

be transformed to a dense spectrum signal by applying a decimation operation. At the same time, although this operation makes the signal less sparse in frequency domain, it does not affect considerably the sparsity in the space domain. The sparsity level on space remains proportional to the signal length.

Figure 6.1 shows an example of a decimation by 2 for a signal x of size 64. In this example, the DFT of x has 14 spikes and 40 zeros (21.88% of non-zeros), while the DFT of $x[2n]$ has 14 spikes and 18 zeros (43.75% of non-zeros). The same happens to the decimated and shifted version of x , $x[2n + 1]$. Both the DFT of $x[2n]$ and $x[2n + 1]$ have half the sparsity of the DFT of x .

Therefore, the decimation property is important to test our hypothesis that the pre-filtering does not reconstruct very well when the filters sparsify the frequency domain. In this chapter, I propose to incorporate the decimation in the pre-filtering model. The goal is to pre-process the measurements in such way they correspond to a signal with a wide spectral bandwidth, while still being sparse in the space domain. However, the

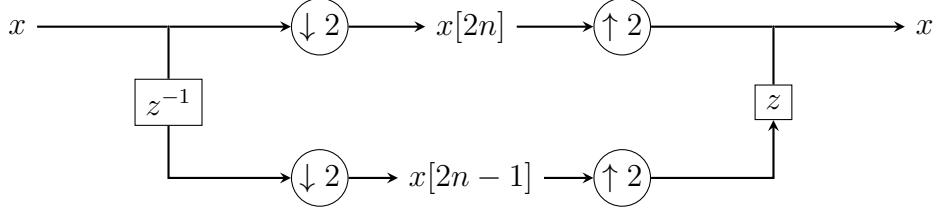


Figure 6.2: The lazy wavelet transform decomposition and reconstruction for a decimation factor of 2.

method still has limitations. For example, each measurement in the decimated domain requires at least 2 original measurements to form a valid measurement for reconstructing the decimated signal. These two measurements may not necessarily be in the original sampling. It is worth pointing out that this chapter describes a work in progress. There are no reconstruction results in this chapter.

6.1 Pre-filtering with decimation

To incorporate the decimation in a reconstruction scheme, such as the pre-filtering, we first need to reconstruct a decimated signal back. For this, I included a polyphase representation that is generally used in filter bank design, which is known as the *Lazy wavelet transform* [81, 99]. The lazy wavelet transform decomposition and reconstruction is shown in Figure 6.2, for a decimation factor of two. The decomposition stage is formed by decimating (with factor of M) every shifted version of a signal, until $M - 1$. The reconstruction is perfect and is formed by the time expansion by a factor of M with the opposite shift.

The lazy wavelet decomposition is applied before the CS reconstruction, while the lazy wavelet reconstruction is performed after the CS reconstruction. However, the inputs of CS reconstruction are undersampled Fourier coefficients. Therefore, the lazy wavelet decomposition is performed in the frequency domain.

Based on the pre-filtering model shown in Figure 4.1 and the lazy wavelet transform, I propose a modified model that is shown in Figure 6.3. Notice that I also included the transformation \mathbf{S}_d before the reconstruction. This transformation samples the measurements in the decimated form. Also, the transformation \mathbf{S}_d is related to the sampling pattern of the original measurements \mathbf{S} . In next section, I present the mathematical formulation that allows to understand the relation between \mathbf{S}_d and \mathbf{S} . This relation will set the limitations of the method in the original sampling frequency positions. Also, the mathematical formulation describes the decimation transformation in frequency domain, $\hat{\mathbf{D}}_0$ and $\hat{\mathbf{D}}_1$.

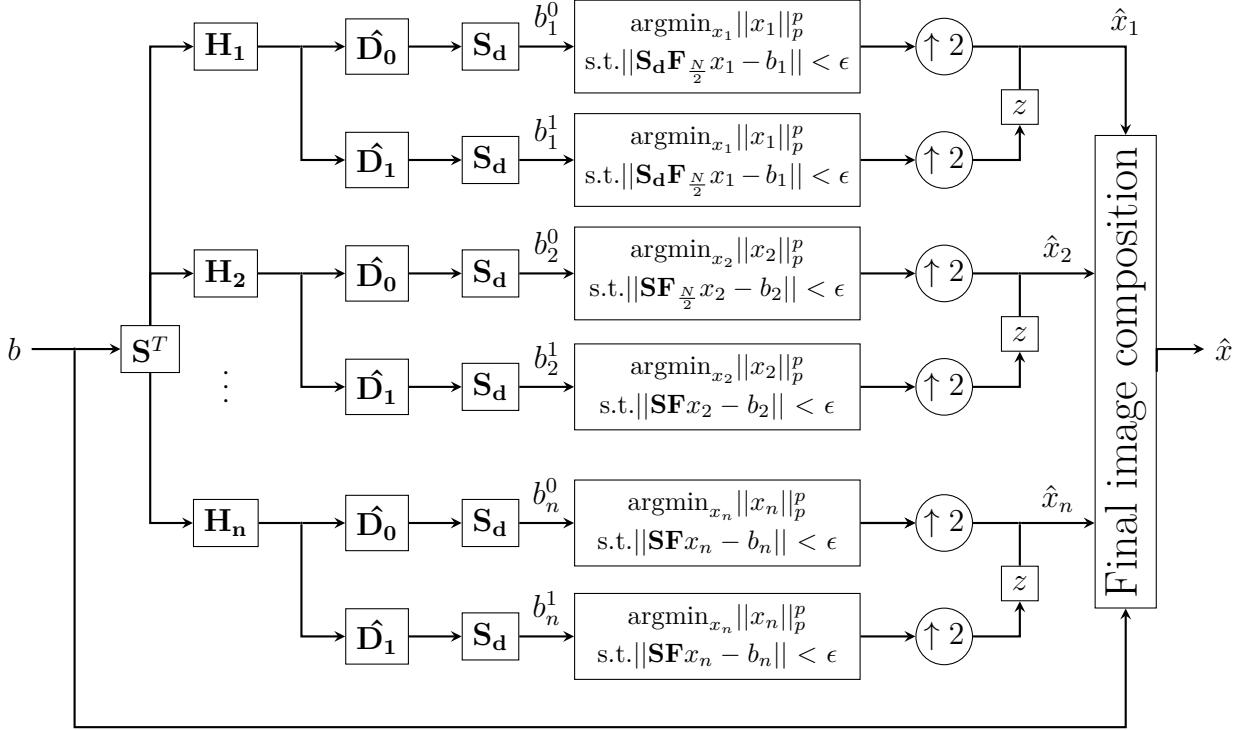


Figure 6.3: Pre-filtering with decimation model.

6.2 Mathematical formulation of the pre-filtering with decimation

The filtering step

I use similar matrix notation as the one in Section 4.1. Let x be a signal in time domain, with length N , and \mathbf{F}_N the N point discrete Fourier transform (DFT) transformation. X is the DFT of x , *i.e.* $X = \mathbf{F}_N x$. Let the vector \mathbf{s} represents the selection of the sampling pattern. The matrix \mathbf{S} selects the \mathbf{s} elements of the input, *i.e.* $\mathbf{S} = \mathbf{I}_{\mathbf{s},*}$. The measurements composed by a selection \mathbf{s} of the Fourier coefficients, given by:

$$b = \mathbf{S} \mathbf{F}_N x, \quad (6.3)$$

therefore, $\mathbf{S} \mathbf{F}_N$ is the sampling matrix.

The first step of the method is to calculate the filtered versions of the measurements b_k . This is performed by multiplying b by a filter in the frequency domain. The matrix $\mathbf{H}_k = \text{diag}(\mathbf{F}_N h_k)$ represents the transformation that multiplies elementwise the spectrum of a N -length signal by the spectrum of the filter h_k . As seen in Section 4.1, the filtered measurements are given by:

$$b_k = \mathbf{S} \mathbf{H}_k \mathbf{S}^T b. \quad (6.4)$$

Time shift and decimation

For a decimation factor of 2, let \mathbf{D}_0 be the matrix that selects the even entries of a signal, and \mathbf{D}_1 the matrix that selects the odd entries. \mathbf{D}_0 can be represented as a matrix formed by the odd rows of a identity matrix and \mathbf{D}_1 a matrix formed by the even rows¹, as given by:

$$\begin{aligned}\mathbf{D}_0 x &= x[2n] \\ \mathbf{D}_1 x &= x[2n + 1],\end{aligned}\tag{6.5}$$

where n ranges from 1 to $N/2$ and the subscripts 0 and 1 refer to the shift applied in the decimation process (z^{-1} block in Figure 6.2).

The signal x is recovered from $x[2n]$ and $x[2n + 1]$ by interlacing $x[2n]$ and $x[2n + 1]$. This is equivalent to zero filling the odd entries of $x[2n]$ and the even entries of $x[2n + 1]$ and summing both. In the matrix formulation, this operation can be represented by:

$$x = \mathbf{D}_0^T x[2n] + \mathbf{D}_1^T x[2n + 1].\tag{6.6}$$

As our measurements are taken in the frequency domain, I want to represent the equivalent of the decimation process (Equation 6.5) in the frequency domain. In other words, I want to find the matrix that, when applied to the spectrum of a signal, corresponds to the operation of decimation in time. Let us denote this matrix as $\hat{\mathbf{D}}_0$, which is given by the relation:

$$\hat{\mathbf{D}}_0 X = \mathbf{F}_{\frac{N}{2}} x[2n] = \mathbf{F}_{\frac{N}{2}} \mathbf{D}_0 x.\tag{6.7}$$

As the signal $x[2n]$ has length $N/2$, I apply the DFT of $N/2$ points to it. As $X = \mathbf{F}_N x$, by excluding x that appears in both sides of the equations, multiplying the result by $\mathbf{F}_N^{-1} = \mathbf{F}_N^T$ on both sides, and isolating $\hat{\mathbf{D}}_0$, we obtain:

$$\hat{\mathbf{D}}_0 = \mathbf{F}_{\frac{N}{2}} \mathbf{D}_0 \mathbf{F}_N^T.\tag{6.8}$$

An example of such a matrix, for $N=4$, is given by:

$$\hat{\mathbf{D}}_0 = 1/2 \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.\tag{6.9}$$

¹Vector signals are denoted to start with index zero, while where matrix rows starts at 1. Thus, the even and odd entries are inverted for signals and matrices

When applied to a spectrum of a signal, this transformation averages the coefficients in the positions i and $i + N/2$, resulting spectrum with $N/2$ points .

Similarly, we can obtain the equivalent in frequency of the shifted version of the decimation operator $\hat{\mathbf{D}}_1$:

$$\hat{\mathbf{D}}_1 = \mathbf{F}_{\frac{N}{2}} \mathbf{D}_1 \mathbf{F}_N^T. \quad (6.10)$$

$\hat{\mathbf{D}}_1$ also averages the coefficients in the positions i and $i + N/2$, but with different weights.

In this formulation, the time shifts and decimation operations are embedded to a single matrix operation ($\hat{\mathbf{D}}_1$), for both time frequency domains. The matrices from Equations 6.8 and 6.10 can be generalized for any decimation factor and shifts in time. The shape of this matrix for a general factor d of decimation is a horizontal concatenation of d diagonal matrices of size N/d each. The different values of shift changes the values of the diagonals. For simplicity, I use only a decimation factor of 2. However, the formulation can be generalized for any natural factor d that divides N .

Applying the time shift and decimation to b_k

The next step consists of applying the decimation and time shift (when applicable) to the filtered version of b_k . The approach is to zero fill b_k at non-sampled positions (\mathbf{S}^T) to get a full frequency spectrum and, then, apply the decimation $\hat{\mathbf{D}}_s$. After this step, we will have a N/d length spectrum, with many non-sampled frequencies. The last step is to get rid of these non-sampled, using a new matrix \mathbf{S}_d , the decimation sampling matrix. Therefore, the decimated and shifted (in time by respectively 0 and 1) versions of b_k are:

$$\begin{aligned} b_k^0 &= \mathbf{S}_d \hat{\mathbf{D}}_0 \mathbf{S}^T b_k \\ b_k^1 &= \mathbf{S}_d \hat{\mathbf{D}}_1 \mathbf{S}^T b_k. \end{aligned} \quad (6.11)$$

In this work, I only use $s = 0$ and $s = 1$ because I am using a decimation factor $d=2$. But, this can be generalized for any shift values between 0 and $d-1$.

With Equations 6.4 and 6.11, we obtain the entire linear transformation applied to the original measurements:

$$b_k^s = \mathbf{S}_d \hat{\mathbf{D}}_s \mathbf{S}^T \mathbf{S} \mathbf{H}_k \mathbf{S}^T b_k. \quad (6.12)$$

The matrices \mathbf{S}_d and \mathbf{S} are very correlated. When we apply $\hat{\mathbf{D}}_s$ (Equation 6.8) to a signal spectrum, we are considering that all points are sampled. But we do not have all the measurements. Notice that \mathbf{S}^T appears just in the right side of $\hat{\mathbf{D}}_s$ in Equation 6.12, so the input of $\hat{\mathbf{D}}_s$ has some zeros corresponding to non-sampled positions. When we apply $\hat{\mathbf{D}}_s$ (matrix in Equation 6.9) we average coefficients that are sampled with coefficients that are not. Therefore, the accuracy of these measurements is compromised. The decimation sampling matrix \mathbf{S}_d has to be built to avoid it.

However, it is possible to select \mathbf{S} in such way that the matrix \mathbf{S}_d does not discard any information. To accomplish that, $\mathbf{S}_d \hat{\mathbf{D}}_s \mathbf{S}^T \mathbf{S}$ must be equal to $\mathbf{S}_d \hat{\mathbf{D}}_s$. To guarantee this equality, the sampling process by \mathbf{S} does not discard the samples kept by \mathbf{S}_d . To avoid discarding samples, the relation between \mathbf{S} and \mathbf{S}_d is as follows:

$$\mathbf{S}^T \mathbf{S} = \begin{bmatrix} \mathbf{S}_d^T \mathbf{S}_d & 0 \\ 0 & \mathbf{S}_d^T \mathbf{S}_d \end{bmatrix}. \quad (6.13)$$

Notice that $\mathbf{S}^T \mathbf{S}$ is a $N \times N$ diagonal matrix, in which $\mathbf{S}^T \mathbf{S}_{(i,i)} = 1$ if i is a sampled position, and $\mathbf{S}^T \mathbf{S}_{(i,i)} = 0$ otherwise. In other words, the sampling pattern must be repeated in the first and second half of the spectrum. The result from Equation 6.13 comes directly from the following theorem.

Theorem 1. *Let*

$$\mathbf{S}^T \mathbf{S} = \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2 \end{bmatrix}, \quad (6.14)$$

where \mathbf{S}_1 and \mathbf{S}_2 are diagonal matrices. Then, if $\mathbf{S}_1 = \mathbf{S}_2 = \mathbf{S}_d^T \mathbf{S}_d$ then $\mathbf{S}_d \hat{\mathbf{D}}_s \mathbf{S}^T \mathbf{S} = \mathbf{S}_d \hat{\mathbf{D}}_s$.

Proof of Theorem 1. Let us rewrite $\hat{\mathbf{D}}_s$ in the form $[\mathbf{D}_L | \mathbf{D}_R]$, where \mathbf{D}_L and \mathbf{D}_R are diagonal matrices, arriving in the following expression:

$$\mathbf{S}_d \hat{\mathbf{D}}_s \mathbf{S}^T \mathbf{S} = \mathbf{S}_d [\mathbf{D}_L | \mathbf{D}_R] \begin{bmatrix} \mathbf{S}_1 & 0 \\ 0 & \mathbf{S}_2 \end{bmatrix} = [\mathbf{S}_d \mathbf{D}_L \mathbf{S}_1 | \mathbf{S}_d \mathbf{D}_R \mathbf{S}_2].$$

Since \mathbf{D}_L , \mathbf{D}_R and \mathbf{S}_1 are all diagonal matrices, they can be commuted. Therefore,

$$[\mathbf{S}_d \mathbf{D}_L \mathbf{S}_1 | \mathbf{S}_d \mathbf{D}_R \mathbf{S}_2] = [\mathbf{S}_d \mathbf{S}_1 \mathbf{D}_L | \mathbf{S}_d \mathbf{S}_2 \mathbf{D}_R].$$

Assuming that $\mathbf{S}_1 = \mathbf{S}_2$, then

$$[\mathbf{S}_d \mathbf{S}_1 \mathbf{D}_L | \mathbf{S}_d \mathbf{S}_2 \mathbf{D}_R] = \mathbf{S}_d \mathbf{S}_1 [\mathbf{D}_L | \mathbf{D}_R] = \mathbf{S}_d \mathbf{S}_1 \hat{\mathbf{D}}_s.$$

Supposing $\mathbf{S}_1 = \mathbf{S}_d^T \mathbf{S}_d$ and using that $\mathbf{S}_d \mathbf{S}_d^T = \mathbf{I}$, arriving in:

$$\mathbf{S}_d \mathbf{S}_1 \hat{\mathbf{D}}_s = \mathbf{S}_d \mathbf{S}_d^T \mathbf{S}_d \hat{\mathbf{D}}_s = \mathbf{S}_d \hat{\mathbf{D}}_s.$$

□

For a general decimation factor d , the relation between \mathbf{S} and \mathbf{S}_d is such that $\mathbf{S}^T \mathbf{S}$ is equal to a diagonal matrix formed by d repetitions of $\mathbf{S}_d^T \mathbf{S}_d$ in the diagonal. Finally,

selecting a sampling matrix according to Equation 6.13, the pre-processed measurements can be obtained by:

$$b_k^s = \mathbf{S}_d \hat{\mathbf{D}}_s \mathbf{H}_k \mathbf{S}^T b. \quad (6.15)$$

Reconstruction

I use compressed sensing to reconstruct each of the pre-processed measurements b_k^s . I want to obtain the signal x_k^s which corresponds to the signal x filtered by the filter h_k , time shifted by s , and decimated by a factor of 2. Therefore, b_k^s corresponds to the undersampled (by $\hat{\mathbf{S}}$) Fourier spectrum of x_k^s . To accomplish that, the CS reconstruction model has to be changed accordingly to the following problem:

$$\begin{aligned} & \operatorname{argmin}_{\hat{x}} \|\hat{x}\|_1 \\ & \text{subject to } \|\hat{\mathbf{S}} \mathbf{F}_{\frac{N}{2}} \hat{x}\|_2 < \epsilon \end{aligned} \quad (6.16)$$

As seen in the model in Figure 6.3, this process is repeated two times for every filter h_k ; one corresponding to decimation with no temporal shift (the path through $\hat{\mathbf{D}}_0$) and the other corresponding to a decimation with a temporal shift ($\hat{\mathbf{D}}_0$).

After these two CS reconstructions, the reconstruction step of the lazy wavelet is performed to recover the filtered versions x_k . Finally, I combine the reconstructed signals x_k and the original measurements in the final composition (Algorithm 1) to obtain the reconstruction of signal x .

6.3 Practical implications

For 2D signals

Figure 6.3 shows the pre-filtering with decimation model for an unidimensional signal x . For 2D signals, which is a case where MRI is applied to, some changes have to be made. The 2D signal $x[n, m]$ is decomposed and perfectly reconstructed using the diagram depicted in Figure 6.4. In this Figure, $\downarrow 2_{1,2}$ represents the decimation operation in both vertical and horizontal dimensions, $[z_1]$ represents the spatial delay in the horizontal dimension, and $[z_2]$ in the vertical dimension.

Sampling pattern limitations

Restrictions for building sampling patterns pose a major limitation in the proposed method. When the transformation $\hat{\mathbf{D}}_s$ is applied to a signal spectrum, it averages (with exponential weights corresponding to the temporal shifts) the values of corresponding

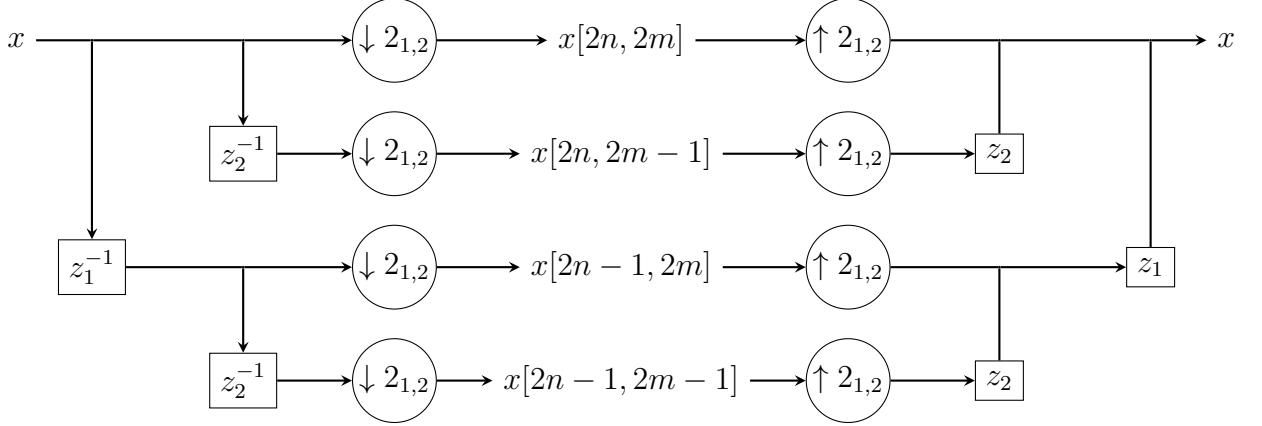


Figure 6.4: The 2D lazy wavelet transform decomposition and reconstruction for a vertical and horizontal decimation factor of 2.

aliased positions i and $i + N/2$. If i is a sampled position and $i + N/2$ is not (or vice versa), this generates an invalid measurement. A reconstruction using invalid measurements does not correspond to the solution space of the original CS problem, and, therefore, this solution is not correct. Therefore, the two possible solutions for this problem are: 1) discard the invalid measurements; or to 2) select a sampling pattern that allows only valid measurements in the acquisition. However, the second option is not general, it requires restrictions to the sampling pattern. To generalize the method for any pattern, measurements need to be discarded.

For example, if a 1D signal of length N is sampled taking m measurements in random positions (*i.e.* a sampling rate of m/N), the expectation value of the valid sampling rate is $(m/N)^2$. For the 2D case, 1 valid measurement is formed by exactly 4 measurements in specific positions. The valid sampling rate of pre-filtering with decimation for a random sampled 2D signal is even smaller than in the 1D case. Figure 6.5 compares the valid sampling rates for a random sampling. Notice that an original sampling rate of 50% corresponds to a valid sampling rate of 25% for the 1D case, and 6.25% for the 2D case. Considering original random sampling rates at the same number of measurements of 20, 40, 60, 80 and 100 radial lines, would result in valid sampling rates of 0.0001, 0.0019, 0.0086, 0.0244 and, 0.0530. Therefore, relying on the valid sampling values for 2D pre-filtering with decimation is impractical.

Increasing the valid sampling rate

However, some strategies to help to increase the valid sampling rate can be used. One of the approaches is to consider we are handling with signals in the domain of real numbers. Let x be a real signal, its DFT X is even and symmetric, which can be mathematically

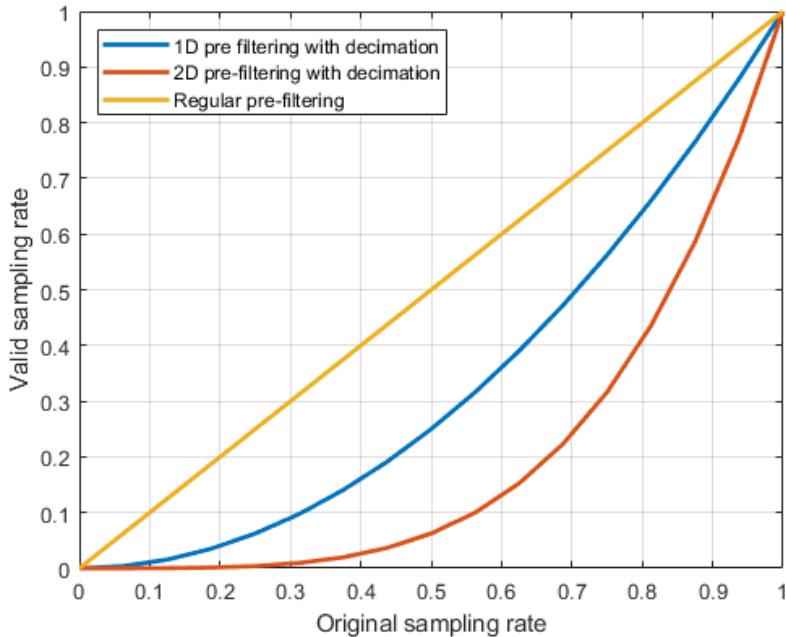


Figure 6.5: Comparison of valid sampling rates compared to the original sampling rate of b .

expressed by [57]:

$$X_{N-k} = X_k^H. \quad (6.17)$$

This means that we can obtain the value of the symmetric position of a measurement using the complex conjugate value of the known measurements. This step can be performed before the pre-processing of the measurements b . We zero-fill b by $\mathbf{S}^T b$, add the values of the symmetric positions, update \mathbf{S} with the new entries, and obtain the updated b by applying the updated \mathbf{S} . However, when using the radial measurements in the 2D k-space, the symmetric spectrum positions are all sampled. Therefore, there is no advantage in using this approach for the radial sampling pattern (the same can not be said for other trajectories).

Other possible strategy relies on the spectrum of the filters h_k , using the ideas presented in Chapter 4. Zero-valued measurements can be included in the positions of the filter spectrum which magnitude is below a given threshold. This increases the number of measurements, and as consequence, increases the valid measurements in the decimation process.

6.4 Conclusions

In this chapter, I proposed including a decimation step to the pre-filtering method. I mathematically formulated the operations to perform the decimation in the

measurements, *i.e.* the undersampling operation in the frequency domain. The formulation still has limitations. Either a specific sampling pattern is required or either most of the measurements cannot be used, which can significantly compromise the reconstruction. This issue is more problematic for the 2D case, in which even less measurements can be used. Solutions to decrease this sampling pattern issue has been proposed, however, this problem has not yet been solved.

Chapter 7

Conclusions

In this thesis, I addressed the undersampled MRI reconstruction problem. By solving this problem, the acquisition time of MRI exams can be reduced, which makes it more affordable, spreads its usage for other types of exams, increases the patient comfort, and improves the image quality.

The approach to solve this problem was based on filtering strategies using CS. Filtering strategies in CS are used in the pre-filtering method and in the reconstruction using the TV minimization algorithms. However, only finite difference filters were tested in these approaches.

I performed a systematic experiment that simulates the undersampled MRI reconstruction using the pre-filtering method implementation with the IRLS algorithm. In this experiment, I tested a large set of filter banks divided in two classes, one designed with a windowing method and other based on wavelet decomposition filters. Filter banks designed with windowing method with a low order provided a significantly higher image quality than the other filter banks. The filter WIN(2,2) provided the best compromise between reconstruction time and reconstruction quality. However, the filters that provided the best results did not present the best ℓ_1 , used as sparsity measure. I could not establish a correlation between the sparsity provided by the filters and reconstruction quality.

I also tested the NESTA ℓ_1 minimization as the reconstruction algorithm for pre-filtering, using the same simulations settings of experiments with IRLS. Using NESTA, some filter banks had a much faster reconstruction time. The reconstruction quality for WAV filters was similar for NESTA and IRLS. However, for WIN filters, NESTA presented worst reconstruction quality than IRLS.

Other proposed method was the inclusion of zero-valued measurements in the reconstruction stage of the pre-filtering method. The method includes a measurement for each spectrum position where the value of the magnitude of the filter is below a

threshold. The threshold of 5% of the maximum amplitude of the filters led to the highest improvements in the reconstruction quality. However, the threshold corresponding to the best results seemed to be content and filter dependent, the best values might be chosen empirically for each content and application.

I proposed a modification of the NESTA algorithm, in which I changed the minimization cost function, called filtering norms. The proposed cost function is a norm of a filter operation that can sparsify the iterated signals, leading to better reconstructions. Different images are better sparsified by different filters, and the proposed method allows the user to choose the filters. The isotropic and anisotropic filtering norms were also implemented. I showed an example of a smooth image that is better reconstructed with a second order filter instead of the finite difference (NESTA TV). For the MRI reconstruction experiments, the highest image quality was obtained by a combination of 8 filters, containing first and second order filters and WIN filters. The results using these filters were statistically significant better than the results obtained TV minimization, specially for the isotropic and anisotropic combinations of the filtering norm.

I developed the mathematical formulation of the inclusion of a decimation in the pre-filtering method. The decimation process decreases the sparsity on the frequency domain while not changing significantly the sparsity of the signal in the space domain. I believe that this could increase the quality of the reconstruction in the reconstruction stages of the pre-filtering. However, the method presented limitations in the measurement process. I showed that a specific sampling pattern is required for obtaining valid measurements for using the method.

Table 7.1 shows the configurations with the best reconstruction quality (and the 95% confidence interval) for every method discussed in this work. Overall, the best reconstruction quality for a lower sampling rate (20 and 40 radial lines) was achieved with IRLS pre-filtering for low order filters WIN with 2 band divisions. In average, the SNR/SSIM was 2.9dB/0.103 higher than NESTA TV minimization for 20 radial lines. The improvement was also noticeable for the different sampling rates, reaching an improvement of 1.2dB/0.003 SNR/SSIM for 100 radial lines.

The inclusion of zero-valued measurements improved the reconstruction quality until 0.4 dB SNR in average for the tested scenarios, and very rarely decreases the quality. With an optimum value of threshold it is possible to further improve the pre-filtering results.

The highest quality for the highest sampling rates was obtained for the isotropic + anisotropic filtering norms with the filters h^8 . The best results exceeded significantly the results of NESTA TV, ranging from values 1.1 dB to 1.5 dB higher SNR than what

Table 7.1: SNR(dB) and SSIM corresponding to the best results of the presented methods.

Method	TV NESTA	pre-filtering IRLS	pre-filtering NESTA 0% t_k	pre-filtering NESTA 5% t_k	filtering norm iaFN
filter bank		WIN(2,3)	WAV(coif,2)	WAV(coif,2)	h^8
SNR(dB), 20 rl	14.9±0.2	17.8±0.2	14.1±0.1	14.5±0.1	16.5±0.2
SNR(dB), 40 rl	21.3±0.2	22.8±0.2	20.7±0.2	21.1±0.1	22.4±0.2
SNR(dB), 60 rl	25.3±0.2	26.3±0.2	25.2±0.2	25.6±0.2	26.6±0.3
SNR(dB), 80 rl	28.4±0.2	29.3±0.3	28.7±0.3	28.9±0.3	29.8±0.3
SNR(dB), 100 rl	31.0±0.3	32.2±0.3	31.5±0.4	31.7±0.4	32.5±0.4
SSIM, 20 rl	0.797±0.011	0.900±0.006	0.742±0.013	0.766±0.011	0.859±0.008
SSIM, 40 rl	0.938±0.004	0.957±0.003	0.924±0.005	0.933±0.004	0.951±0.003
SSIM, 60 rl	0.971±0.002	0.976±0.000	0.968±0.002	0.971±0.002	0.977±0.002
SSIM, 80 rl	0.983±0.001	0.986±0.001	0.983±0.001	0.984±0.001	0.987±0.001
SSIM, 100 rl	0.989±0.000	0.992±0.000	0.990±0.000	0.990±0.000	0.992±0.000

was obtained with NESTA TV (on average). Isotropic+anisotropic filtering norms also presented the best overall results for 60, 80 and 100 radial lines.

The thesis presented filtering methods that improve the reconstruction quality of MRI images when compared to finite difference filters based methods. I presented the filter banks to use in each method which most improved the image quality. I obtained a significative gain in the reconstruction quality, for both low and high sampling rates. The proposed techniques and filter banks can be used in other MRI reconstruction techniques and using real MRI machines data. It can further improve the image quality obtained by these MRI reconstruction techniques, or even decrease the acquisition time of MRI exams. However, some unanswered question still remain, letting opportunities for future works.

7.1 Future works

One of the main unsolved issues in this thesis is the relation between sparsity and reconstruction quality. I expected a strong correlation, but this was not observed in the sparsity and quality metrics results obtained. Further studies are needed in order to determine if there is a relationship between this two variables.

The IRLS algorithm presented the best overall results in the pre-filtering. Also, the inclusion of zero-valued measurements almost always improved the reconstruction quality, despite only being implemented for the NESTA algorithm. A future work is to implement the zero-valued measurements inclusion using the IRLS algorithm.

Another future work is the implementation of the pre-filtering with decimation, started in this work. Also, the problem of the restrictions on the sampling pattern is still open.

Perhaps hybrid approaches can be performed for obtaining the spectrum positions that are required for the restrictions of the sampling pattern before the reconstruction.

Other methods inspired by pre-filtering are also possible. One of them is based on the wavelet decomposition and recombination. The measurements are decomposed in its wavelet components. The final recombination could be performed by a sum of the reconstruction of components. The advantage could be that we are not necessarily pursuing sparsity, but the best wavelet decompositions.

As the test for filters were empirical, different filters could lead to even better results with the proposed methods. Designing these filters by optimization processes, maximizing the reconstruction quality (of the factors that lead to the best quality) would be a valuable work.

The internal reconstructions of the pre-filtering method were all performed with a ℓ_p solver, and usually with $p=1$. Different approaches could be tested. Using TV minimization or even the filtering norms are possible approaches. Specially, when using the filtering norm, 2 filters banks will be used in the problem, the one used by filtering the measurements, and other used as reconstruction cost function. Testing these combination of filters and their features that lead to better reconstruction quality would be an interesting study.

Bibliography

- [1] Philip Palin Dendy and Brian Heaton. *Physics for diagnostic radiology*. CRC press, 2011. iv, 1
- [2] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Inf. Theory*, 52(2):489–509, 2006. iv, 2, 15, 16, 27
- [3] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008. iv, 2
- [4] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992. iv, 2, 16
- [5] Kai Tobias Block, Martin Uecker, and Jens Frahm. Undersampled radial MRI with multiple coils. iterative image reconstruction using a total variation constraint. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 57(6):1086–1098, 2007. iv, 2, 20
- [6] Matthias J Ehrhardt and Marta M Betcke. Multicontrast MRI reconstruction with structure-guided total variation. *SIAM Journal on Imaging Sciences*, 9(3):1084–1106, 2016. iv, 2
- [7] Junzhou Huang, Chen Chen, and Leon Axel. Fast multi-contrast MRI reconstruction. *Magnetic resonance imaging*, 32(10):1344–1352, 2014. iv, 2
- [8] Chen Chen, Yeqing Li, Leon Axel, and Junzhou Huang. Real time dynamic MRI with dynamic total variation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 138–145. Springer, 2014. iv, 2
- [9] Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing mri. *IEEE signal processing magazine*, 25(2):72, 2008. iv, 2, 16
- [10] Li Feng, Robert Grimm, Kai Tobias Block, Hersh Chandarana, Sungheon Kim, Jian Xu, Leon Axel, Daniel K Sodickson, and Ricardo Otazo. Golden-angle radial sparse parallel mri: combination of compressed sensing, parallel imaging, and golden-angle radial sampling for fast and flexible dynamic volumetric mri. *Magnetic resonance in medicine*, 72(3):707–717, 2014. iv, 2

- [11] Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011. iv, 2, 15, 16, 21, 22, 60
- [12] C Miosso, R Von Borries, and J Pierluissi. Compressive sensing method for improved reconstruction of gradient-sparse magnetic resonance images. In *IEEE Forty-Third Asilomar Conference on Signals, Systems and Computers*, pages 799–806, 2009. v, 3, 5, 8, 17, 18, 19, 31, 33
- [13] Yong Lim and Sydney Parker. Fir filter design over a discrete powers-of-two coefficient space. *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, 31(3):583–591, 1983. v, 26
- [14] Rick Chartrand and Wotao Yin. Iteratively reweighted algorithms for compressive sensing. In *Acoustics, speech and signal processing, 2008. ICASSP 2008. IEEE international conference on*, pages 3869–3872. IEEE, 2008. vi, 14
- [15] J Lima, C Miosso, and M Farias. Per-pixel mirror-based acquisition method for video compressive sensing. In *European Signal Processing Conference (EUSIPCO)*, Lisbon, Portugal, 2014. xi
- [16] Paula Lima, Jonathan Lima, and Priscila Solis. Rea-wsn: Intercluster routing algorithm for energy optimization in wireless sensor networks. In *2015 7th IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–7, 2015. xi
- [17] J Lima, C Miosso, and M Farias. Per-pixel mirror-based method for high-speed video acquisition. *J. of Visual Communication and Image Representation*, 47:23–35, 2017. xi
- [18] Jonathan A. Lima, Cristiano J. Miosso, and Mylène C.Q. Farias. Avaliação de filtros de decomposição wavelet para reconstrução de imagens de ressonância magnética com base em compressive sensing com pré-filtragem. In *V Congresso Brasileiro de Eletromiografia e Cinesiologia e X Simpósio de Engenharia Biomédica (COBEC-SEB 2017)*, 2017. xi
- [19] Jonathan Lima, Cristiano Miosso, Mylene Farias, and Ricardo von Borries. Evaluation of different types of filters in magnetic resonance imaging using compressive sensing with pre-filtering. In *Engineering in Medicine and Biology Society (EMBC), 2018 40th Annual International Conference of the IEEE*. IEEE, 2018. xi, 25, 66
- [20] MRI: A guided tour. <https://nationalmaglab.org/education/magnet-academy/learn-the-basics/stories/mri-a-guided-tour>. Accessed: 2016-02-15. xiv, 9
- [21] Rebecca Smith-Bindman, Jafi Lipson, Ralph Marcus, Kwang-Pyo Kim, Mahadevappa Mahesh, Robert Gould, Amy Berrington De González, and Diana L Miglioretti. Radiation dose associated with common computed tomography examinations and the associated lifetime attributable risk of cancer. *Archives of internal medicine*, 169(22):2078–2086, 2009. 1

- [22] L Zhi-Pei and P Lauterbur. *Principles of magnetic resonance imaging: a signal processing perspective*. “The” Institute of Electrical and Electronics Engineers Press, 2000. 1
- [23] Chelsea S Kidwell, Julio A Chalela, Jeffrey L Saver, Sidney Starkman, Michael D Hill, Andrew M Demchuk, John A Butman, Nicholas Patronas, Jeffry R Alger, Lawrence L Latour, et al. Comparison of MRI and ct for detection of acute intracerebral hemorrhage. *Jama*, 292(15):1823–1830, 2004. 1
- [24] J Kalita and UK Misra. Comparison of ct scan and MRI findings in the diagnosis of japanese encephalitis. *Journal of the neurological sciences*, 174(1):3–8, 2000. 1
- [25] O Lubovsky, M Liebergall, Y Mattan, Y Weil, and R Mosheiff. Early diagnosis of occult hip fractures: MRI versus CT scan. *Injury*, 36(6):788–792, 2005. 1
- [26] Martin Köhrmann, Eric Jüttler, Jochen B Fiebach, Hagen B Huttner, Stefan Siebert, Christian Schwark, Peter A Ringleb, Peter D Schellingen, and Werner Hacke. Mri versus ct-based thrombolysis treatment within and beyond the 3 h time window after stroke onset: a cohort study. *The Lancet Neurology*, 5(8):661–667, 2006. 1
- [27] Mark A Horsfield and Derek K Jones. Applications of diffusion-weighted and diffusion tensor MRI to white matter diseases—a review. *NMR in Biomedicine: An International Journal Devoted to the Development and Application of Magnetic Resonance In Vivo*, 15(7-8):570–577, 2002. 1
- [28] M. Greicius, G. Srivastava, A. Reiss, and V. Menon. Default-mode network activity distinguishes alzheimer’s disease from healthy aging: evidence from functional MRI. *Proc. of the Nat. Acad. of Sci. of the USA*, 101(13):4637–4642, 2004. 1
- [29] S. Gregory, R Blair, et al. Punishment and psychopathy: A case-control functional MRI investigation of reinforcement learning in violent antisocial personality disordered men. *The Lancet Psychiatry*, 2(2):153–160, 2015. 1
- [30] S Spiteri, T Hassa, D Claros-Salinas, et al. Functional MRI changes illustrating cognitive fatigue in patients with multiple sclerosis. In *Rehabilitationswissenschaftliches Kolloquium*, 2016. 1
- [31] Gabriel S Dichter, Devin Gibbs, and Moria J Smoski. A systematic review of relations between resting-state functional-MRI and treatment response in major depressive disorder. *Journal of affective disorders*, 172:8–17, 2015. 1
- [32] Sam J Marzo and John P Leonetti. The importance of magnetic resonance imaging in the evaluation of vertigo and imbalance. *Skull base surgery*, 10(04):0171–0172, 2000. 1
- [33] Jörg Wellmer, Joachim Von Oertzen, Carlo Schaller, Horst Urbach, Roy König, Guido Widman, Dirk Van Roost, and Christian E Elger. Digital photography and 3D MRI-based multimodal imaging for individualized planning of resective neocortical epilepsy surgery. *Epilepsia*, 43(12):1543–1550, 2002. 1

- [34] Marie T Krüger, Volker A Coenen, Carolin Jenkner, Horst Urbach, Karl Egger, and Peter C Reinacher. Combination of CT angiography and MRI in surgical planning of deep brain stimulation. *Neuroradiology*, 60(11):1151–1158, 2018. 2
- [35] Rushna Ali, Muhib Khan, Victor Chang, Jayant Narang, Rajan Jain, Horia Marin, Jack Rock, and Max Kole. MRI pre and post-embolization enhancement patterns predict surgical outcomes in intracranial meningiomas. *Journal of Neuroimaging*, 26(1):130–135, 2016. 2
- [36] T Krings, MHT Reinges, S Erberich, S Kemeny, V Rohde, U Spetzger, M Korinth, K Willmes, JM Gilsbach, and A Thron. Functional MRI for presurgical planning: problems, artefacts, and solution strategies. *Journal of Neurology, Neurosurgery & Psychiatry*, 70(6):749–760, 2001. 2
- [37] Melissa A Mallory, Yasuaki Sagara, Fatih Aydogan, Stephen DeSantis, Jagadeesan Jayender, Diana Caragacianu, Eva Gombos, Kirby G Vosburgh, Ferenc A Jolesz, and Mehra Golshan. Feasibility of intraoperative breast MRI and the role of prone versus supine positioning in surgical planning for breast-conserving surgery. *The breast journal*, 23(6):713–717, 2017. 2
- [38] Muhammad Asad Parvaiz, Peiming Yang, Eisha Razia, Margaret Mascarenhas, Caroline Deacon, Pilar Matey, Brian Isgar, and Tapan Sircar. Breast MRI in invasive lobular carcinoma: a useful investigation in surgical planning? *The breast journal*, 22(2):143–150, 2016. 2
- [39] Nicole Wake, Temitope Rude, Stella K Kang, Michael D Stifelman, James F Borin, Daniel K Sodickson, William C Huang, and Hersh Chandarana. 3D printed renal cancer models derived from MRI data: application in pre-surgical planning. *Abdominal Radiology*, 42(5):1501–1509, 2017. 2
- [40] David Atkinson, Derek LG Hill, Peter NR Stoyle, Paul E Summers, Stuart Clare, Richard Bowtell, and Stephen F Keevil. Automatic compensation of motion artifacts in mri. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 41(1):163–170, 1999. 2
- [41] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. on pure and applied mathematics*, 59(8):1207–1223, 2006. 2, 13
- [42] Richard G Baraniuk. Compressive sensing [lecture notes]. *Signal Processing Magazine, IEEE*, 24(4):118–121, 2007. 2, 13, 15
- [43] Emmanuel Candes and Justin Romberg. Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3):969, 2007. 2, 15
- [44] B. Dale, M. Brown, and R. Semelka. *MRI: basic principles and applications*. John Wiley & Sons, 2015. 2
- [45] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In *Doklady AN USSR*, volume 269, pages 543–547, 1983. 3, 21

- [46] Fábio Jeronimo Costa. Reconstrução de imagens de ressonância magnética com base em compressive sensing com pré-filtragem no domínio de medidas usando computação paralela. dissertação de mestrado em engenharia biomédica. Master's thesis, Faculdade do Gama da Universidade de Brasília, 2016. 3, 20
- [47] Daniel Almeida. Reconstrução de imagens de ressonância magnética com base em compressive sensing usando informação a priori estrutural em abordagem bayesiana. Master's thesis, Faculdade do Gama da Universidade de Brasília, 2017. 3
- [48] LC Balling and FM Pipkin. Gyromagnetic ratios of hydrogen, tritium, free electrons, and rb 85. *Physical Review*, 139(1A):A19, 1965. 9
- [49] Roald K Wangsness and Felix Bloch. The dynamical theory of nuclear induction. *Physical Review*, 89(4):728, 1953. 10
- [50] Zhi-Pei Liang and Paul C Lauterbur. *Principles of magnetic resonance imaging: a signal processing perspective*. SPIE Optical Engineering Press, 2000. 13
- [51] Martin Uecker, Shuo Zhang, Dirk Voit, Alexander Karaus, Klaus-Dietmar Merboldt, and Jens Frahm. Real-time MRI at a resolution of 20 ms. *NMR in Biomedicine*, 23(8):986–994, 2010. 13
- [52] Aaron Niebergall, Shuo Zhang, Esther Kunay, Götz Keydana, Michael Job, Martin Uecker, and Jens Frahm. Real-time MRI of speaking at a resolution of 33 ms: Undersampled radial flash with nonlinear inverse reconstruction. *Magnetic Resonance in Medicine*, 69(2):477–485, 2013. 13
- [53] Volker Rasche, Ruud W De Boer, Dietrich Holz, and Roland Proksa. Continuous radial data acquisition for dynamic mri. *Magnetic resonance in medicine*, 34(5):754–761, 1995. 13
- [54] Markus Untenberger, Zhengguo Tan, Dirk Voit, Arun A Joseph, Volkert Roeloffs, K Dietmar Merboldt, Sebastian Schätz, and Jens Frahm. Advances in real-time phase-contrast flow MRI using asymmetric radial gradient echoes. *Magnetic resonance in medicine*, 75(5):1901–1908, 2016. 13
- [55] Li Feng, Thomas Benkert, Kai Tobias Block, Daniel K Sodickson, Ricardo Otazo, and Hersh Chandarana. Compressed sensing for body mri. *Journal of Magnetic Resonance Imaging*, 45(4):966–987, 2017. 13
- [56] Siemens Healthineers. *Compressed Sensing GRASP-VIBE*, 2017 (accessed June 14, 2019). <http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm>. 13
- [57] Sanjit Kumar Mitra and Yonghong Kuo. *Digital signal processing: a computer-based approach*, volume 2. McGraw-Hill New York, 2006. 13, 27, 28, 84
- [58] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006. 13, 15

- [59] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. on Inf. Theory*, 52(12):5406–5425, 2006. 13
- [60] David L Donoho. Compressed sensing. *IEEE Trans. on Inf. Theory*, 52(4):1289–1306, 2006. 13, 14, 15
- [61] Biswa Nath Datta. *Numerical linear algebra and applications*. SIAM, 2010. 14
- [62] Richard G Baraniuk. Compressive sensing. *IEEE signal processing magazine*, 24(4), 2007. 14
- [63] C Miosso, R von Borries, M Argaez, L Velázquez, C Quintero, and C Potes. Compressive sensing reconstruction with prior information by iteratively reweighted least-squares. *IEEE Trans. on Signal Proc.*, 57(6):2424–2431, 2009. 14
- [64] Michael Elad, Peyman Milanfar, and Ron Rubinstein. Analysis versus synthesis in signal priors. *Inverse problems*, 23(3):947, 2007. 16
- [65] Rustum Choksi, Yves van Gennip, and Adam Oberman. Anisotropic total variation regularized l_1 -approximation and denoising/deblurring of 2D bar codes. *arXiv preprint arXiv:1007.1035*, 2010. 16, 61
- [66] Selim Esedoḡlu and Stanley J Osher. Decomposition of images by the anisotropic rudin-oshер-fatemi model. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(12):1609–1626, 2004. 16
- [67] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009. 16, 61, 62
- [68] Laurent Condat. Discrete total variation: New definition and minimization. *SIAM Journal on Imaging Sciences*, 10(3):1258–1290, 2017. 16, 61
- [69] Emmanuel Candès. *l_1 -MAGIC*, 2005 (accessed June 14, 2019). <https://statweb.stanford.edu/~candes/l1magic/>. 19
- [70] J Fessler and B Sutton. Nonuniform fast fourier transforms using min-max interpolation. *IEEE Trans. on Signal Processing*, 51(2):560–574, 2003. 20
- [71] Junfeng Yang, Yin Zhang, and Wotao Yin. A fast alternating direction method for $tv_l_1-l_2$ signal reconstruction from partial fourier data. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):288–297, 2010. 20, 62
- [72] Penghang Yin, Yifei Lou, Qi He, and Jack Xin. Minimization of l_1-l_2 for compressed sensing. *SIAM Journal on Scientific Computing*, 37(1):A536–A563, 2015. 20, 62
- [73] Bo Liu, Yi Ming Zou, and Leslie Ying. Sparsesense: application of compressed sensing in parallel mri. In *2008 International Conference on Information Technology and Applications in Biomedicine*, pages 127–130. IEEE, 2008. 20

- [74] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008. 20
- [75] P Tseng. On accelerated proximal gradient methods for convex-concave optimization. *SIAM Journal on Control and Optimization*, 2008. 21, 23
- [76] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005. 21, 22, 23
- [77] Taeyoung Ha, Wookeen Chung, and Changsoo Shin. Waveform inversion using a back-propagation algorithm and a huber function norm. *Geophysics*, 74(3):R15–R24, 2009. 22
- [78] Jonathan A. Lima, Cristiano J. Miosso, and Mylène C.Q. Farias. Avaliação de filtros de decomposição wavelet para reconstrução de imagens de ressonância magnética com base em compressive sensing com pré-filtragem. In *V Congresso Brasileiro de Eletromiografia e Cinesiologia e X Simpósio de Engenharia Biomédica (COBEC-SEB 2017)*, 2017. 25
- [79] M Wickerhauser. *Adapted wavelet analysis from theory to software*. IEEE press, 1994. 27
- [80] J Grethe, C Baru, A Gupta, et al. Biomedical informatics research network: building a national collaboratory to hasten the derivation of new understanding and treatment of disease. *Studies in health technology and informatics*, 112:100–110, 2005. 28
- [81] A Primer, C Sidney Burrus, and Ramesh A Gopinath. Introduction to wavelets and wavelet transforms, 1998. 29, 77
- [82] Z Wang, A Bovik, H Sheikh, and E Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE trans. on Image Proc.*, 13(4):600–612, 2004. 31, 32
- [83] Z Wang and A Bovik. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009. 32
- [84] Mylène CQ Farias. *Video quality metrics*. INTECH Open Access Publisher, 2010. 32
- [85] X Xu, S Xu, L Jin, and E Song. Characteristic analysis of Otsu threshold and its applications. *Pattern recognition letters*, 32(7):956–961, 2011. 32
- [86] Robert M Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006. 59, 62
- [87] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 60

- [88] Emmanuel Candes and Justin Romberg. l1-magic: Recovery of sparse signals via convex programming. *URL: www. acm. caltech. edu/l1magic/downloads/l1magic.pdf*, 4:14, 2005. 60
- [89] João FC Mota, Nikos Deligiannis, and Miguel RD Rodrigues. Compressed sensing with prior information: Strategies, geometry, and bounds. *IEEE Transactions on Information Theory*, 63(7):4472–4496, 2017. 62
- [90] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. 62
- [91] Harald Birkholz. A unifying approach to isotropic and anisotropic total variation denoising models. *Journal of computational and applied mathematics*, 235(8):2502–2514, 2011. 62
- [92] Yifei Lou, Penghang Yin, Qi He, and Jack Xin. Computing sparse representation in a highly coherent dictionary based on difference of l1 and l2. *Journal of Scientific Computing*, 64(1):178–196, 2015. 62
- [93] Yifei Lou, Tieyong Zeng, Stanley Osher, and Jack Xin. A weighted difference of anisotropic and isotropic total variation model for image processing. *SIAM Journal on Imaging Sciences*, 8(3):1798–1823, 2015. 62
- [94] Yifei Lou and Ming Yan. Fast l1–l2 minimization via a proximal operator. *Journal of Scientific Computing*, 74(2):767–785, 2018. 62
- [95] Lawrence A Shepp and Benjamin F Logan. The fourier reconstruction of a head section. *IEEE Transactions on nuclear science*, 21(3):21–43, 1974. 64
- [96] Gilbert G Walter and Xiaoping Shen. *Wavelets and other orthogonal systems*. CRC press, 2000. 66
- [97] Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. NESTA a fast and accurate first-order method for sparse recovery. <https://statweb.stanford.edu/~candes/nesta/>. Acessed: 2018-06-01. 66
- [98] Donald F Morrison. Multivariate analysis of variance. *Encyclopedia of biostatistics*, 5, 2005. 66
- [99] Wim Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and computational harmonic analysis*, 3(2):186–200, 1996. 77