

UNICAMP

Estudo e implementação da reconstrução de imagens de MR adquiridas com múltiplas bobinas

Relatório Final – Iniciação Científica

Aluno: Rafael Ferreira da Costa Vescovi



Orientadora: Gabriela Castellano



1. Introdução

Neste relatório descrevem-se as atividades realizadas e resultados obtidos durante todo o projeto. Para uma melhor linearidade das idéias e conteúdos abordados, a Seção 3 irá incluir na íntegra todos os conteúdos abordados no relatório parcial. A Seção 2 apresenta uma descrição da proposta inicial, a Seção 3 descreve as tarefas realizadas, e na Seção 4 são apresentados os resultados alcançados. Estes resultados são discutidos na Seção 5, e a Seção 6 apresenta conclusões e um plano de trabalho para futuros projetos.

2. Proposta Inicial

O desempenho dos scanners de imagens de ressonância magnética (MRI) deu um salto na última década graças ao advento da aquisição paralela, ou aquisição com múltiplas bobinas. Esse tipo de aquisição permite em princípio diminuir a resolução temporal - o que é sempre importante em aquisições de imagens médicas, particularmente em estudos de ressonância magnética funcional (fMRI) - ou melhorar a razão sinal-ruído (SNR) das imagens adquiridas.

Este projeto teve por objetivo o estudo da reconstrução de imagens de MR adquiridas com múltiplas bobinas. Planejou-se para o projeto uma fase teórica, que incluiria o estudo dos princípios físicos da geração do sinal de MR e da reconstrução das imagens de MR no caso de aquisição com uma única bobina; e uma fase prática, consistindo da familiarização com os arquivos de dados brutos (“raw”) fornecidos pela máquina, e desenvolvimento de rotinas para a leitura dos mesmos.

É importante ressaltar que nossa instituição possui um scanner de MR com campo permanente de 3T, adquirido dentro do Programa CInAPCe da Fapesp, que entrou em funcionamento no início de 2009. Um scanner desse tipo pode ser visto na Figura 1. Este scanner opera com aquisição por múltiplas bobinas, e faz uma reconstrução “caixa-preta” baseada no método SENSE, portanto o estudo deste método é de grande interesse para o grupo, e para possíveis projetos futuros visando à melhoria da reconstrução e/ou do método SENSE [1-5].



Figura 1 – Scanner Phillips Achieva 3T. Um scanner deste tipo está instalado no Hospital das Clínicas da Unicamp.

3. Tarefas Realizadas

Para a realização deste trabalho foram estipulados alguns objetivos que estão listados a seguir, com uma descrição do que foi realizado até o momento e considerações sobre sua realização.

3.1. Estudo básico sobre a geração do sinal de MR e formação da imagem de MR

Para a realização desta parte do trabalho, foi utilizado o livro *Functional Magnetic Resonance Imaging* [A] em conjunto com a participação do aluno em reuniões semanais sobre ressonância magnética ministradas pela professora Gabriela Castellano e alunos de pós-graduação na área de Neurofísica.

O fenômeno de ressonância magnética existe pois, dentre os átomos que compõem a matéria, existem alguns (^1H , ^2H , ^{10}B , ^{11}B , ^{13}C , ^{14}N , ^{15}N , ^{17}O , ^{19}F , ^{23}Na , ^{29}Si , ^{31}P , ^{35}Cl , ^{113}Cd , ^{195}Pt) que apresentam o que chamamos de spin nuclear. Embora não haja maneira simples de prever se um determinado átomo terá ou não spin, para o caso dos átomos cujos prótons e/ou nêutrons apresentam-se em quantidades ímpares, como no caso do ^1H e ^{13}C , esta característica sempre estará presente. Nos núcleos que possuem spin $1/2$, como o ^1H , existem dois estados de spins, chamados de “up” (acima) e “down” (abaixo). Como este é o núcleo utilizado para fazer a maioria das imagens de ressonância magnética, a partir daqui nos restringiremos à descrição apenas do sistema do núcleo de ^1H , ou um próton.

A presença de um spin (que funciona como um dipolo magnético) em um campo magnético B_0 gera um torque equivalente ao produto vetorial do vetor de dipolo com o vetor do campo. O torque exercido pelo campo faz com que esses “spins” precessionem ao redor da direção do campo. Os prótons então podem assumir dois auto-valores de energia (o que é previsto pelo efeito Zeeman), um de maior energia (“anti-paralelo ao campo”) e outro de menor energia (“paralelo ao campo”). A separação entre esses níveis de energia é dada por

$$\Delta E = \hbar \omega_0 \quad (1)$$

onde

$$\omega_0 = \gamma B_0 \quad (2)$$

ω_0 é a frequência de Larmor, proporcional à intensidade do campo magnético aplicado, e γ é uma constante de proporcionalidade, a razão giromagnética, que depende do tipo de núcleo observado.

Nos estudos de ressonância magnética aplicados a usos médicos, que usam o núcleo de ^1H já mencionado, para o qual $\gamma = 2\pi \cdot 42.576 \text{ MHz/T}$, e campos magnéticos com intensidades entre 0.5 e 7 Teslas, a diferença de energia prevista pelo efeito Zeeman corresponde à energia de um fóton com comprimento de onda na faixa de radiofrequência (RF), aproximadamente entre 100 a 300 MHz.

Isso possibilita que exista uma interação desses fótons de energia específica com os prótons da amostra (ou área de interesse no corpo), que promove a transição dos spins entre as populações de maior e menor energia.

A soma dos dipolos magnéticos dos núcleos da amostra é uma magnetização resultante, que é a grandeza passível de ser medida nos experimentos de RM. Essa magnetização total pode ser dividida em sua componente transversal (perpendicular ao campo) e longitudinal (paralela ao campo). Como os spins precessionam com fases aleatórias, a componente da magnetização perpendicular ao campo se cancela, sobrando apenas a componente longitudinal, que é proporcional à (pequena) diferença de prótons que ocupam os dois estados de energia. Geralmente o estado de menor energia possui um pequeno excesso de prótons em relação ao estado de maior energia, o que resulta numa magnetização total paralela ao campo magnético aplicado.

A magnetização longitudinal pode ser manipulada com pulsos de RF, os chamados pulsos de 90° e 180° , que “rotam” a magnetização para o plano transversal ou a invertem. Após a aplicação destes pulsos, a magnetização tende a voltar ao seu estado original, seguindo um processo conhecido como relaxação. A relaxação ocorre em dois processos independentes, conhecidos como relaxação longitudinal (ou spin-rede) e relaxação transversal (ou spin-spin).

A relaxação longitudinal é a recuperação da componente longitudinal da magnetização, é um processo exponencial caracterizado pela constante de tempo T_1 , e está relacionado à volta dos prótons no estado excitado (de maior energia) para o estado original (de menor energia). A relaxação transversal corresponde à destruição da componente transversal da magnetização, que ocorre devido à perda de coerência nas fases dos prótons. Este também é um processo exponencial, e é caracterizado pela constante de tempo T_2 .

A variação da componente transversal da magnetização durante a relaxação produz um sinal que induz uma pequena corrente nas bobinas externas (no caso de nossa máquina podem ser utilizadas até 32 dessas bobinas). Esse sinal é chamado de FID (*free induction decay*).

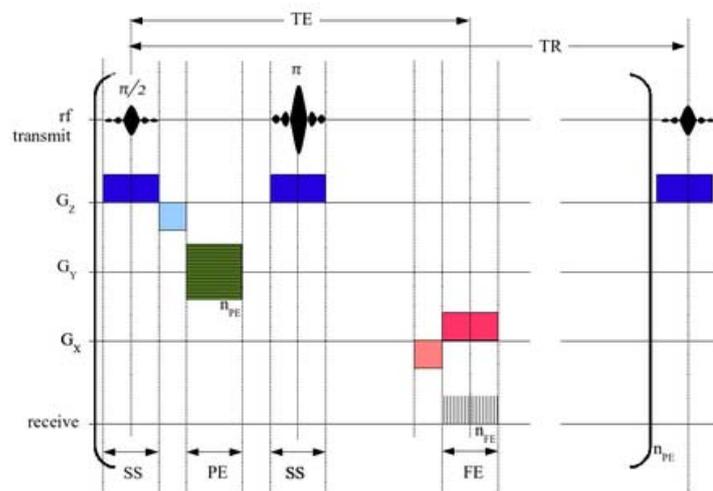


Figura 2 – Exemplo de sequência de pulsos para ressonância magnética. Na imagem acima rf transmit são os pulsos de RF, G_z , G_x e G_y são os gradientes e receive é a ligação das bobinas que captam o sinal do FID.

Em um experimento ideal, o FID decai com uma curva exponencial com constante de tempo T_2 . No entanto, em experimentos reais experimentamos pequenas diferenças devido a inhomogeneidades no campo magnético em diferentes posições espaciais. Essas inhomogeneidades fazem com que as frequências de Larmor variem através da área de interesse (ROI) de modo a criar uma interferência destrutiva que reduz o sinal do FID. Esse decaimento observado nos experimentos reais também é exponencial e chamado de T_2^* , que é sempre menor que T_2 .

Para conseguirmos “mapear” nossas imagens, temos campos que variam através do volume a ser escaneado, os chamados gradientes. A presença desses gradientes faz com que determinados locais tenham frequências de Larmor diferentes. Normalmente os gradientes são pulsados e podem existir infinitas variações de sequências de pulsos para ligar e desligar os gradientes e a RF, o que dá às imagens de ressonância magnética sua incrível versatilidade. Na Figura 2 podemos ver um exemplo de sequência de pulsos para ressonância magnética.

Normalmente tecidos moles têm T_1 de aproximadamente um segundo, e T_2 e T_2^* de alguns milissegundos. Vale lembrar que essas constantes variam bruscamente de tecido para tecido, dando às imagens de ressonância grande contraste para tecidos moles.

3.2. Estudo sobre a transformada de Fourier e suas propriedades

Na matemática, a Transformada de Fourier (FT) é uma operação que tem como principal objetivo levar do domínio original, tipicamente o tempo, para outro domínio, tipicamente a frequência. Esse fato é muito importante na análise de sinais, pois a transformada nos mostra quais frequências têm quais intensidades no sinal sendo analisado.

Em outras palavras, menos precisas, porém mais simples, a FT decompõe em senos e cossenos a função original (Figura 3), ou em exponenciais complexas. Matematicamente, se $y(t)$ é uma função do tempo, sua transformada de Fourier $Y(f)$, onde f é a frequência, é dada por:

$$Y(f) = \int_{-\infty}^{\infty} y(t) \exp(-i2\pi ft) dt. \quad (3)$$

Alternativamente, conhecendo-se $Y(f)$, a função $y(t)$ pode ser recuperada através da transformada de Fourier inversa, dada por:

$$y(t) = \int_{-\infty}^{\infty} Y(f) \exp(i2\pi ft) df. \quad (4)$$

Nos estudos de MR, em que o sinal medido (FID) é amostrado no tempo, o que se tem é um sinal discreto y_n com $n = 0, 1, \dots, N - 1$, onde N é o número de amostras. Nesse caso, usa-se a transformada de Fourier discreta (DFT), dada por:

$$Y_k = \frac{1}{N} \sum_{n=0}^{N-1} y_n \exp(-i \frac{2\pi kn}{N}) , k = 0 \text{ a } N-1, \quad (5)$$

onde k representa a amostragem no espaço das frequências.

Similarmente ao caso contínuo, o sinal y_n pode ser recuperado conhecendo-se a série discreta Y_k através da DFT inversa, dada por:

$$y_n = \sum_{k=0}^{N-1} Y_k \exp(i \frac{2\pi kn}{N}) , n = 0 \text{ a } N-1. \quad (6)$$

A transformada de Fourier possui uma série de propriedades importantes, sumarizadas na Tabela 1 [H]. Estas propriedades estão escritas para a transformada contínua, mas possuem seus equivalentes para a DFT.

Tabela 1 - Propriedades da Transformada de Fourier Contínua.

$y(t)$ é real	$Y(-f) = [Y(f)]^*$
$y(t)$ é imaginária	$Y(-f) = -[Y(f)]^*$
$y(t)$ é par	$Y(f)$ é par
$y(t)$ é ímpar	$Y(f)$ é ímpar
$y(at)$	$\frac{1}{ a } Y(\frac{f}{a})$
$\frac{1}{ b } y(\frac{t}{b})$	$Y(bf)$
$y(t - t_0)$	$Y(f) e^{-2\pi j f t_0}$
$y(t) e^{2\pi j f_0 t}$	$Y(f - f_0)$

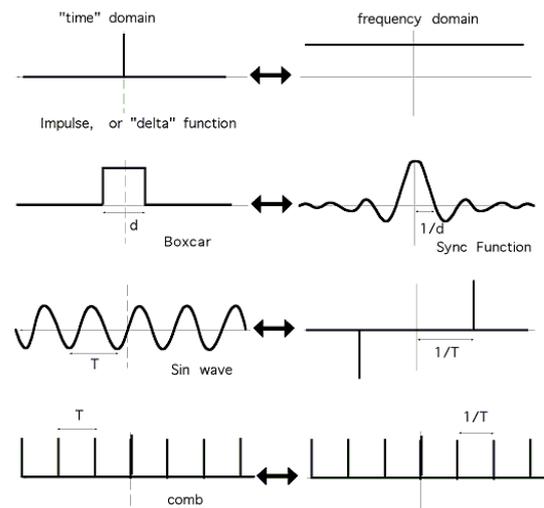


Figura 3 - Exemplos de transformadas de Fourier em 1 dimensão.

O estudo sobre FT se baseou em conhecimentos prévios adquiridos em disciplinas do curso e nas aulas ministradas pela Universidade de Stanford, que estão gratuitamente disponíveis no site da universidade, pelo projeto Academic Earth [G].

3.3. Estudo sobre a aquisição dos dados no espaço-k

Utilizando os livros *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective* [B] e *Introduction to Functional Magnetic Resonance Imaging: Principles and Techniques* [C], também foi estudada a aquisição de imagens no espaço-k. Este espaço é utilizado pela máquina de MR por ser o espaço de saída das imagens de ressonância magnética.

O espaço-k é o domínio no qual são adquiridos os dados de um experimento de MR. O sinal de MR $S(k_x, k_y)$, conhecido como FID (Seção 3.1), é medido nesse domínio, e se relaciona com a magnetização transversal M_{xy} da fatia sendo escaneada por uma transformada de Fourier bidimensional [A] (Figura 4):

$$S(k_x, k_y) = \iint_{x,y} M_{xy}(x, y, t) e^{-i2\pi k_x(t)x} e^{-i2\pi k_y(t)y} dx dy, \quad (7)$$

onde

$$M_{xy}(x, y, t) = M_{xy}(x, y) e^{-t/T_2} e^{-i\gamma B_0 t}. \quad (8)$$

Aqui consideramos que a fatia já foi selecionada, de modo que a magnetização não depende mais de z . Na Eq. (8), T_2 é a constante de tempo da relaxação transversal, B_0 é a intensidade do campo magnético estático do scanner (que geralmente se assume na direção z), e γ é a razão giromagnética do núcleo do hidrogênio (que é o núcleo usado no caso de imagens). As variáveis k_x e k_y da Eq. (7) definem o espaço-k, e dependem das intensidades dos gradientes de campo magnético G_x e G_y e do tempo de aplicação dos mesmos:

$$k_x(t) = \frac{\gamma}{2\pi} \int_0^t G_x(\tau) d\tau \quad \text{e} \quad (9)$$

$$k_y(t) = \frac{\gamma}{2\pi} \int_0^t G_y(\tau) d\tau.$$

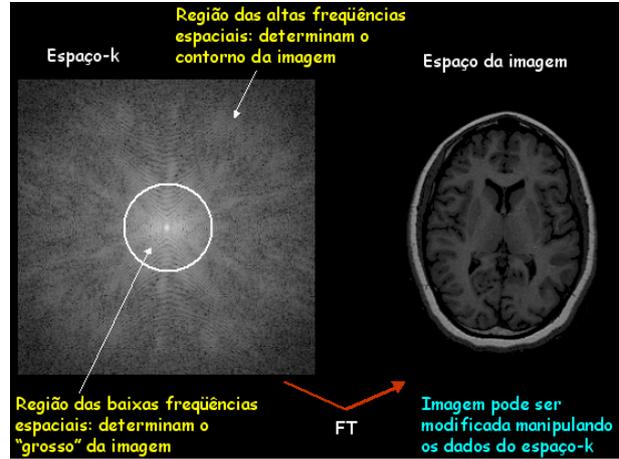


Figura 4 - Espaço-k de coleta de dados de MR (ou domínio das frequências espaciais) e correspondente espaço da imagem (ou domínio espacial). Ambos espaços estão relacionados pela transformada de Fourier bidimensional.

Podemos pensar nas Eqs. (9) como indicando *trajetórias* no espaço-k. Na prática, uma sequência de pulsos que preencha o espaço-k seguindo uma grade cartesiana (como as sequências *spin-echo* e *gradient-echo* citadas anteriormente), varre cada linha k_x de forma contínua, ou seja, apenas deixa o gradiente G_x ligado e vai medindo o sinal S ao longo do tempo, enquanto G_x permanecer ligado. Isto corresponde a codificar a frequência dos spins da fatia de acordo com sua posição, e portanto k_x é conhecido como a **direção de codificação de frequência**. Já o parâmetro k_y é varrido aplicando o gradiente G_y por um tempo determinado, antes da medida do sinal. Para que cada aplicação de G_y resulte numa linha diferente no espaço-k, deve-se variar a intensidade ou o tempo de aplicação de G_y . Isto resulta em que spins em diferentes posições terão fases acumuladas diferentes, e portanto k_y é conhecido como a **direção de codificação de fase** (Figura 5).

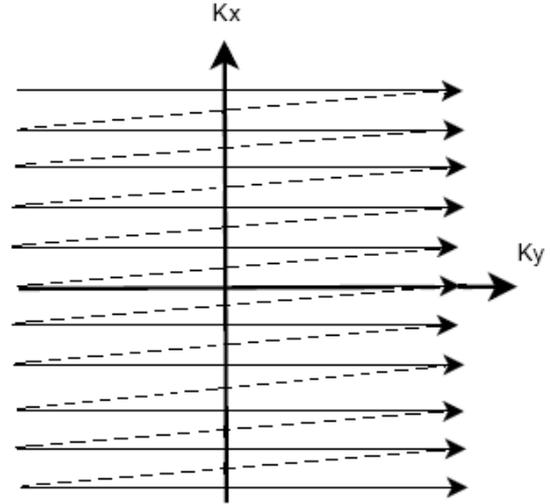


Figura 5 - Trajetória “tradicional” no espaço-k, onde cada linha é preenchida de forma contínua, e o “pulo” de uma linha para outra é feito através da variação da intensidade ou tempo de aplicação de G_y .

Antes de ser armazenado no computador, o sinal de MR medido é discretizado utilizando um conversor analógico-digital, de modo que podemos relacionar o sinal no espaço-k diretamente com a intensidade $I(l,m)$ da imagem de MR obtida, re-escrevendo a Eq.(7) como :

$$S(p, q) = \sum_l \sum_m I(l, m) \exp\left(-\frac{i2\pi l p}{N_x}\right) \exp\left(-\frac{i2\pi m q}{N_y}\right), \quad (10)$$

(4)

com

$$\begin{aligned} k_x &= p \Delta k_x, & k_y &= q \Delta k_y, \\ x &= l \Delta x, & y &= m \Delta y, \\ \Delta k_x &= \frac{1}{N_x \Delta x}, & \Delta k_y &= \frac{1}{N_y \Delta y}, \end{aligned} \quad (11)$$

onde $I(l,m)$ representa a intensidade da imagem no pixel (l,m) , N_x, N_y são respectivamente o número de pixels na matriz de imagem nas direções x e y , e $\Delta k_x, \Delta k_y, \Delta x, \Delta y$ são respectivamente os intervalos de amostragem nas direções x e y do espaço-k e do espaço da imagem.

Portanto, para reconstruir a imagem de MR a partir de um espaço-k cartesiano, basta aplicar uma transformada de Fourier discreta (DFT) nos dados medidos do espaço-k.

3.4. Estudo sobre o método SENSE para reconstrução de imagens a partir de dados coletados por múltiplas bobinas, no caso de grade cartesiana no espaço-k

Como mencionado anteriormente, a aquisição de dados por bobinas múltiplas e reconstrução através da codificação de sensibilidade (SENSE) permite reduzir o tempo de scan. Em particular, se o espaço-k for coletado de forma a gerar uma grade cartesiana, a codificação de sensibilidade permite reduzir o número de passos necessários durante a aquisição. Isso é feito aumentando a distância entre as posições amostradas no espaço-k, porém mantendo os valores máximos de k_x e k_y . Deste modo, o tempo de scan é reduzido, porém a resolução espacial (da imagem) é preservada. O fator pelo qual o número de amostras no espaço-k é reduzido é chamado de **fator de redução R**. A transformada de Fourier do espaço-k amostrado dessa forma mais grosseira resulta numa imagem com campo de visão (FOV) reduzido, e portanto ocorre *aliasing* (Figura 6). O primeiro passo da reconstrução SENSE consiste em criar uma imagem *aliased* para cada bobina do arranjo, utilizando a DFT.

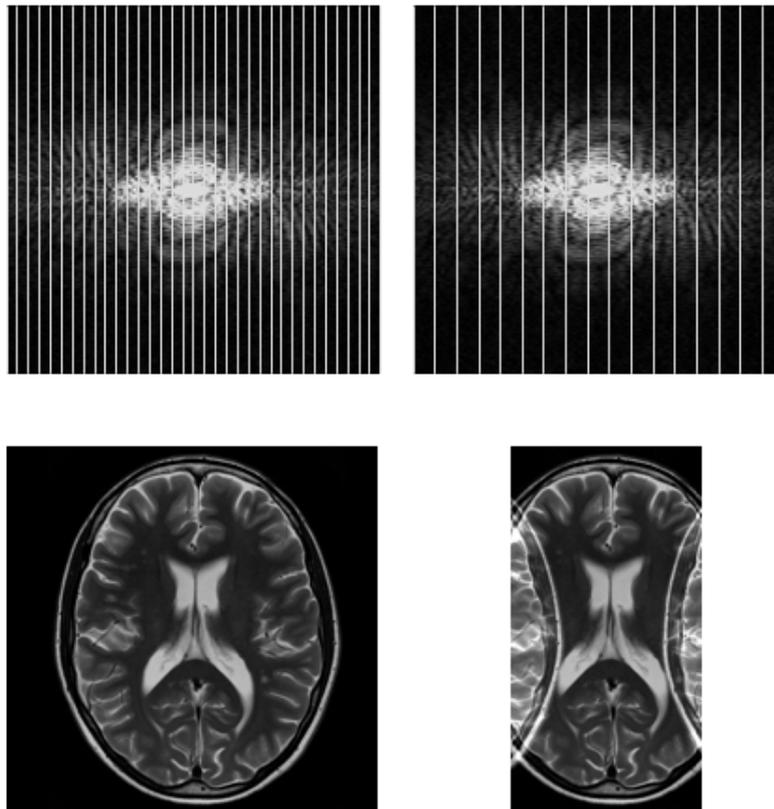
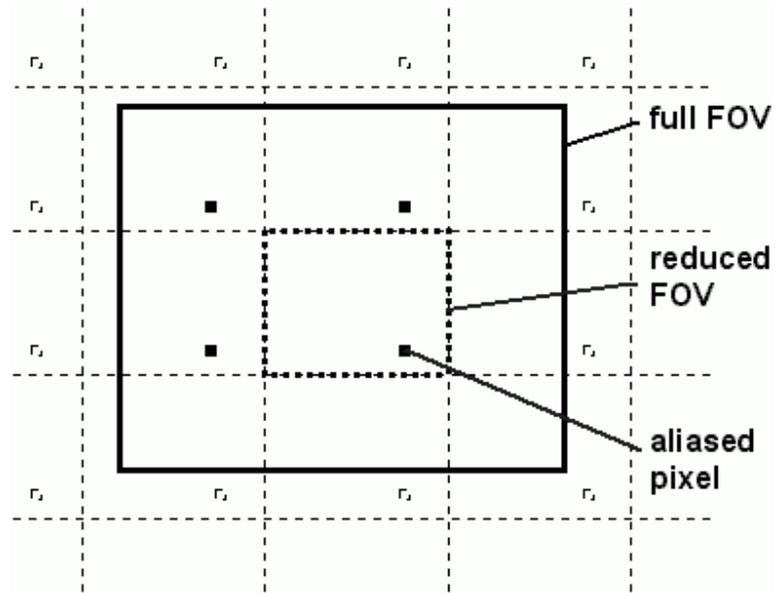


Figura 6 – As imagens acima ilustram o acontecimento do fenômeno chamado *aliasing*. As duas imagens de cima são o mesmo espaço-k colhido com $R=1$ e $R=2$ respectivamente, e abaixo suas respectivas reconstruções. É interessante observar que ambas as imagens reconstruídas contêm toda a informação de intensidade da imagem original, no entanto não podemos separar as contribuições superpostas pelo *aliasing* para $R > 1$.

O segundo passo da reconstrução é criar uma imagem de campo completo (*full FOV*) a partir do conjunto de imagens *aliased*. Para isso, é necessário desfazer a sobreposição de sinais que ocorre devido ao efeito de “sobreposição” causado pelo *aliasing*. Ou seja, cada pixel no FOV reduzido possui contribuições de um certo número de posições, e estas precisam ser separadas. Essas posições formam uma grade cartesiana com o tamanho do FOV reduzido (Figura 7).

Para poder separar os sinais sobrepostos em um pixel no FOV reduzido, usa-se o fato de que essa sobreposição ocorre com pesos diferentes em cada imagem gerada por cada bobina, devido à sensibilidade local da bobina naquela posição. Consideremos um pixel no FOV reduzido e os pixels sobrepostos correspondentes no FOV completo. Chamando de n_p o número de pixels sobrepostos, e de n_c o número de bobinas,



podemos montar um vetor \mathbf{a} (de dimensões $n_c \times 1$) para cada pixel no FOV reduzido contendo os valores (complexos) deste pixel para cada imagem coletada (de cada bobina). O que queremos é, a partir de \mathbf{a} , encontrar um vetor \mathbf{v} (de dimensões $n_p \times 1$), que contenha os valores dos pixels que foram sobrepostos :

$$\mathbf{v} = U \mathbf{a}. \quad (12)$$

Podemos ilustrar melhor o sentido da Eq. (12) observando a Figura 8, onde temos à esquerda a imagem final e à direita a imagem coletada pela bobina 1. Os pixels ρ_1 , ρ_2 e ρ_3 contribuem com pesos diferentes para cada pixel do vetor \mathbf{a}_1 . Essa montagem nos mostra claramente uma das limitações do método, pois o número de incógnitas (n_p) não pode ultrapassar o número de equações (n_c), ou, em outras palavras, o fator de redução não pode ser maior que o número de bobinas utilizadas para fazer a imagem.

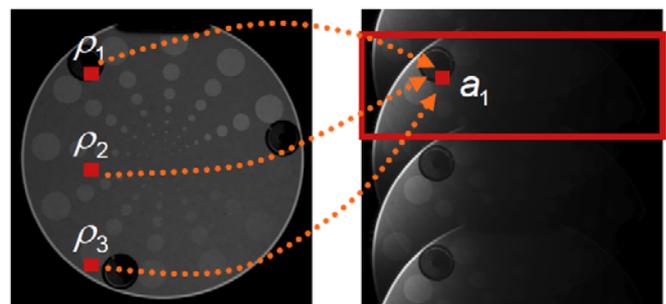


Figura 8 – Ilustração do desdobramento de pixels sobrepostos apresentado na Eq. (12), onde os ρ_n representam os pixels da imagem final e a_1 os pixels sobrepostos.

A matriz U é chamada de **matriz de desdobramento** (*unfolding matrix*), e é calculada a partir da matriz S de sensibilidade das bobinas e da matriz Ψ de correlação de ruído entre estas (de dimensões $n_C \times n_C$):

$$U = (S^H \Psi^{-1} S)^{-1} S^H \Psi^{-1}, \quad (13)$$

onde o índice H indica a matriz complexa conjugada transposta. A matriz de sensibilidade S do conjunto de bobinas possui dimensões $n_C \times n_P^1$, e é formada pelos valores de sensibilidade de cada bobina para cada pixel sobreposto:

$$S_{\gamma, \rho} = s_{\gamma}(\mathbf{r}_{\rho}), \quad (14)$$

onde os índices γ e ρ se referem às bobinas e aos pixels sobrepostos respectivamente, \mathbf{r}_{ρ} é a posição do ρ -ésimo pixel e s_{γ} é a sensibilidade espacial da γ -ésima bobina.

Como já comentado anteriormente o desdobramento (Eq. (12)) é possível desde que as matrizes Ψ e $S^H \Psi^{-1} S$ que aparecem na Eq. (13) sejam inversíveis. Em particular, o número de pixels a serem separados, n_P , não pode exceder o número de bobinas n_C . Ou seja, o fator de redução R está limitado por n_C .

A matriz de correlação do ruído pode ser calculada a partir de dados das imagens correspondentes a regiões “vazias” das mesmas (sem “objeto”, i.e., sem paciente). Esses dados consistem basicamente de valores de ruído branco obtidos por cada bobina. De posse de uma grande quantidade desses valores organizados em um vetor para cada bobina, calculamos a covariância entre os vetores e assim obtemos uma matriz Ψ com dimensões $n_C \times n_C$.

A matriz de ruído (ou correlação) Ψ que aparece na Eq. (13) serve para otimizar a razão sinal-ruído (SNR) ao reconstruir as imagens. No entanto, ela pode ser substituída pela matriz identidade – nesse caso, o desdobramento ainda é possível, mas à custa de SNR reduzida, o que ficará mais evidente quanto mais diferentes forem as bobinas em relação a ganho, carga e acoplamento mútuo.

A Eq. (13) é válida independente de se a sub-amostragem é feita nas 2 ou apenas em 1 direção do espaço-k. Para a redução de tempo de scan em imageamento 2D, geralmente usa-se sub-amostragem apenas na direção de codificação de fase.

A matriz de sensibilidade S pode ser determinada utilizando imagens de FOV completo adquiridas com cada bobina, e uma imagem (também de FOV completo) adquirida com uma bobina de corpo (*body coil*). A homogeneidade da bobina de corpo é usada como referência, e mapas de sensibilidade para cada bobina podem ser obtidos dividindo ponto a ponto a imagem obtida por cada bobina pela imagem obtida com a bobina de corpo [1]. Após a divisão, é feito um refinamento dos mapas, que consiste em eliminar ruído e extrapolar as sensibilidades (geralmente é feito utilizando um ajuste por *splines* numa grade 2D), e

¹ Note-se que existe uma matriz de sensibilidade com estas dimensões para cada pixel da imagem final.

também em zerar regiões da imagem que não contribuem de forma significativa para o sinal(Figura 9). Ou seja, regiões onde se sabe de antemão que não há “objeto”, são zeradas, pois o grau de *aliasing* é localmente reduzido quando um ou mais componentes *aliased* são nulos.

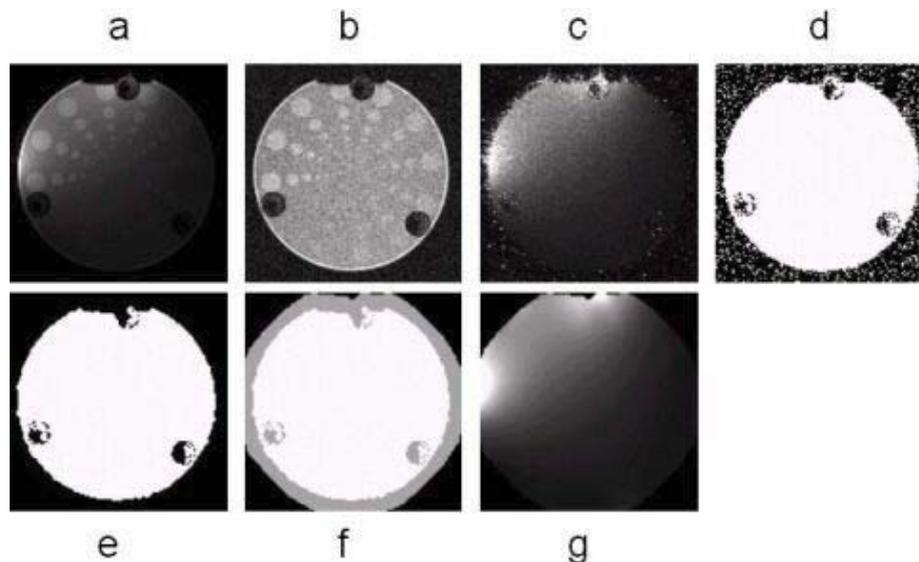


Figura 9 – A figura acima mostra os passos para a montagem dos mapas de sensibilidade. Primeiro temos uma imagem de uma bobina “a” e uma imagem da body coil “b”. Então temos uma divisão de “a” por “b” formando “c”. “d” é o resultado da aplicação de um limiar em “c”, e após um filtro do tipo “salt and pepper” temos “e”. Com uma extrapolação das bordas da imagem “e”² chegamos em “f” [D]. Para completar é utilizado um “spline 2-D” para completar a região de interesse em “f” com valores médios colhidos em “c”, formando assim nosso mapa final “g” (1). Esse mapa final mostrado é que será usado como peso para o sistema de equações representado pela Eq. (12), e corresponde ao mapa de sensibilidade S apresentado na Eq. (13). No caso de não ser possível adquirir uma imagem de bobina de corpo (*body coil*), pode-se usar como referência uma imagem criada a partir da raiz quadrada da soma dos quadrados das imagens de cada bobina.

3.5.Familiarização com o software Matlab

Partindo de conhecimentos prévios sobre programação, o aluno desenvolveu rapidamente certa fluência na linguagem do Matlab. Para isso foi utilizado o manual do Matlab e suas seções de “ajuda”. É possível encontrar essa mesma documentação no site www.mathworks.com [F], onde também é possível obter exemplos e algoritmos já desenvolvidos compartilhados por outros programadores.

Um aspecto interessante do programa é a utilização de técnicas complicadas e algoritmos super otimizados como o FFT, “Fast Fourier Transform”, que simplificam processos que são bastante complexos mesmo em nível computacional.

² Isso pode ser feito através de uma operação chamada “dilatação”, que é uma das operações básicas de morfologia matemática [ref].

Outra coisa que deve ser lembrada é a otimização do software para trabalhar com matrizes (esparsas ou não), o que no caso de um estudo com imagens deixa o trabalho mais dinâmico, diminui a chance de erros, abordagens erradas e/ou sub-ótimas.

Também foi estudado o livro “*Digital Image Processing*” [E] para o estudo da manipulação de imagens digitais e da teoria de Morfologia Matemática, que vai ser bastante utilizada durante o trabalho de reconstrução das imagens. É interessante notar que no livro são discutidas varias abordagens sobre como programar ou atacar cada problema, e o Matlab quase sempre aparece com soluções mais simples e ótimas.

Esse grande poder vem do fato dele ter seu núcleo completamente programado em “C” (linguagem de programação), e portanto apesar de ser uma aplicação não estática, não perde muita velocidade e flexibilidade.

3.6.Familiarização com os dados *raw* do scanner Intera 3T da Philips

Os dados mais puros que se pode obter do nosso aparelho são chamados de dados “*raw*” ou “*crus*” na tradução direta para o português. Esses dados são divididos em dois arquivos, um “.list” e um “.data” (a partir de agora os chamaremos de LIST e DATA). Os arquivos LIST são formados por descritores que contêm informação sobre parâmetros da aquisição, mapas para a leitura do arquivo DATA, fatores de correção e outras informações relevantes para a reconstrução. Os arquivos DATA são basicamente um grande vetor de números complexos; esses números representam intensidades de cada espaço-k coletado durante a aquisição. Para organizar esses dados é necessário o mapa de descritores que se encontra no LIST. A Figura 10 mostra, de forma bastante simplificada, como são os arquivos LIST (acima) e DATA (abaixo).

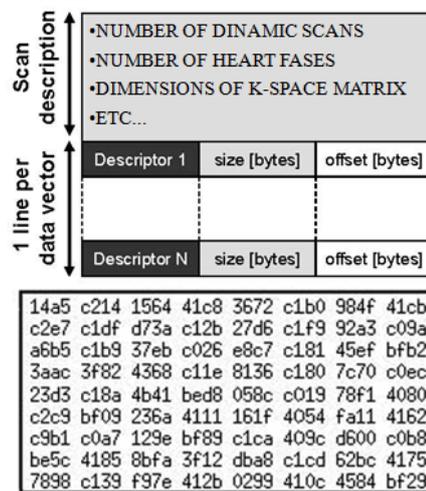


Figura 10 – Versão simplificada de como estão organizados os arquivos LIST (acima) e DATA (abaixo).

Para se ter uma ideia do volume de dados envolvidos em uma reconstrução SENSE feita com dados “*raw*”, temos que uma imagem comum, feita com um fator SENSE de 2 (R = 2), tem aproximadamente 100.000 (cem mil) linhas de descritores para que possa ser lido o arquivo DATA, que contém aproximadamente 500.000 números complexos e 100Mbytes de dados. Para se organizar esses dados é necessário um poder de processamento e velocidade de endereçamento dificilmente alcançados nos computadores comuns, o que explica o fato de existir um servidor dedicado para essa tarefa no equipamento de MR, e todas suas comunicações internas importantes serem feitas com cabos de fibra ótica.

Um fato observado durante os estudos foi que as informações contidas no LIST não eram necessariamente iguais para todas as imagens adquiridas, ou seja, alguns parâmetros introduzidos pelo usuário geravam arquivos LIST diferentes com descritores diferentes e/ou em ordens diferentes. Essa falta de redundância das informações torna um desafio criar uma

rotina de leitura para qualquer LIST. Essa dificuldade foi contornada durante o projeto utilizando uma teoria de programação chamada de Expressões Regulares (REGEXP ou Regular Expressions) [F], que está muito bem descrita no “Help” do programa (Matlab) utilizado para criar as rotinas.

3.7.Desenvolvimento de rotinas para leitura dos dados *raw* do scanner Achieva 3T da Philips

Utilizando o software Matlab foram criados uma série de rotinas e algoritmos para a leitura e organização dos dados *raw*. Primeiramente foi necessário ler e organizar os dados contidos no LIST, e de posse do esquema de organização do DATA, organizar os dados que vão formar nossos futuros espaços-k.

Para a leitura do LIST foi gerado um algoritmo que vai separando e analisando linha a linha enquanto “lê” o arquivo. Este foi o método mais eficiente encontrado, pois linhas com informações diferentes começam com um caractere diferente. Por exemplo, ao encontrar um “#”, o programa sabe que aquela é uma linha de comentário e não se dá ao trabalho de continuar absorvendo suas informações. Outros descritores importantes são o “.” (que chamaremos de INFO) para linhas que contêm dados (parâmetros da aquisição) e “ ” (que chamaremos de MAPA) para linhas que contêm informações sobre a organização do arquivo DATA.

Os descritores INFO são formados por 3 números inteiros, um nome e um conjunto de valores. Por exemplo

```
. 0 1 2 this_is_an_example : 22 06 89
```

é uma linha com informações que serão armazenadas para auxiliar na reconstrução futura. Sem esses valores seria impossível descobrir como e quando processar cada dado obtido. Entre esses descritores estão desde parâmetros inseridos pelo usuário, como fator SENSE e número de fatias que serão reconstruídas, até fatores de correção gerados durante a aquisição de dados, como fatores de correção de fase devido a inomogeneidades do campo.

Os descritores MAPA são mais complexos que os abordados anteriormente, sendo constituídos de um “tag” (símbolo de 3 letras que descreve o tipo de dado no DATA para o qual o mapa aponta) e um conjunto de 22 valores numéricos, que guardam informações como o byte inicial no arquivo DATA, seu tamanho em bytes, sua posição no espaço-k, a fatia à qual esses dados pertencem, entre outras (não menos importantes) informações. É importante falar sobre os tipos de descritores MAPA que encontramos, pois eles nos dizem o tipo de informação que esta guardada no nosso arquivo DATA.

Podemos ter descritores MAPA dos seguintes tipos (que chamaremos de sub-descritores): STD (*standard*), que apontam para valores de pixels dos espaços-k que serão reconstruídos; REJ (*rejected*), que também apontam para pixels dos espaços-k, mas que não entrarão na reconstrução final; PHC (*phase correction*), apontam para parâmetros relacionados à correção de fase dos pixels em alguns tipos especiais de aquisição; FRC (*frequency correction*), apontam para parâmetros relacionados à correção de frequência;

NAV (*navigation*), apontam para dados relacionados à navegação para determinados tipo de aquisição; NOI, apontam para medidas do ruído branco nas imagens.

Para nosso estudo, os sub-descritores (do descritor MAPA) realmente importantes são os: 1) STD, que carregam informações das imagens parciais colhidas por cada bobina - estas formarão o vetor \mathbf{a} (Eq. (12)) e serão usadas para a determinação das matrizes \mathbf{S} (Eq. (13)); e 2) NOI, que carregam informações de ruído branco – estas medidas de ruído serão usadas para o cálculo da matriz Ψ (Eq. (13)) de cada reconstrução.

Os descritores do tipo MAPA carregam muitas informações, mas principalmente o tamanho e o byte inicial de cada vetor de valores do espaço-k que estão contidos no arquivo DATA. Graças a isso somos capazes de gerar espaços-k para as imagens contidas em um conjunto de arquivos LIST e DATA.

4. Resultados

Após o fim do projeto, as rotinas de leitura dos dados LIST e DATA são capazes de interpretar e armazenar todos os dados relevantes no arquivo LIST. Esses dados são de extrema importância para o entendimento do funcionamento do aparelho e da relevância de cada parâmetro na reconstrução final.

Com os dados obtidos até o momento foi possível obter um maior entendimento sobre como é a física por trás das sequências de pulso específicas para cada tipo de imagem. Um exemplo disso foi a descoberta de como o número de bobinas de radiofrequência (*number_of_rf_channels*) afeta os dados colhidos no espaço-k. Essa diferença no método de aquisição de dados pode ser relevante ou não para determinados tipos de imagens e/ou tecidos que se deseja analisar.

Em relação à eficiência das rotinas desenvolvidas para a leitura e organização dos arquivos LIST, temos que o tempo de execução da primeira versão das rotinas era de aproximadamente de 10 a 20 minutos para a leitura de cada arquivo LIST envolvido no processo de reconstrução. Após um aprofundamento no funcionamento do método e dos programas utilizados, foi conseguido um tempo de aproximadamente 15 segundos para a leitura e organização de qualquer LIST. Um fato observado interessante é que mesmo testado em computadores mais ou menos poderosos que o do equipamento de MR, o tempo de execução das rotinas permaneceu muito parecido.

5. Discussão

O projeto foi completado sem grandes problemas técnicos. Como apresentado acima, os principais empecilhos encontrados foram a imensa quantidade de informações relevantes contidas nos arquivos LIST. Esse problema foi contornado com a criação de uma rotina para sua leitura e organização que se mostrou tão eficiente quanto possível, deixando de depender do computador utilizado.

Embora o desenvolvimento de rotinas para a leitura e organização dos arquivos DATA já tenha sido iniciado, foi considerado que isto fazia parte das rotinas de reconstrução propriamente ditas, o que constitui um segundo projeto (a ser desenvolvido no próximo semestre), e portanto seu detalhamento não está presente neste relatório. Podemos adiantar que foram encontrados alguns problemas para a criação de rotinas que pudessem manipular as matrizes de até 12 dimensões criadas durante o processo de organização dos valores contidos em cada DATA. Além disso, desde já podemos esperar que a pouca variação de tempo de execução encontrada para as rotinas de manipulação dos arquivos LIST para diversos computadores não ocorrerá para as rotinas de manipulação dos arquivos DATA, já estas envolvem a execução de inúmeras transformadas de Fourier e desdobramento de pixels *aliased*.

6. Considerações Finais

Apresentamos neste relatório um resumo dos tópicos estudados ao longo deste projeto, assim como uma descrição das rotinas desenvolvidas para a manipulação dos dados *raw* expelidos pelo equipamento de MR Achieva 3.0T da Philips. Os tópicos estudados envolveram os princípios físicos da geração do sinal de MR, o espaço-k de coleta dos dados de MR, a transformada de Fourier, o básico sobre sequências de pulsos, a reconstrução das imagens de MR adquiridas com uma bobina, o método SENSE de reconstrução de imagens de MR adquiridas com múltiplas bobinas, o software Matlab, e o entendimento dos arquivos de dados *raw* expelidos pelo equipamento de MR.

Os próximos passos do projeto incluem a continuação dos estudos apresentados neste relatório, e também a implementação das rotinas para a reconstrução das imagens adquiridas com múltiplas bobinas. Também será necessário um estudo mais detalhado de cada aspecto dos dados *raw* que nossa máquina de MRI fornece, e a implementação de rotinas cada vez mais eficientes para sua leitura e organização.

É de extrema importância o estudo cada vez mais detalhado desses dados pois isso permite um melhor entendimento do funcionamento real do equipamento de MR, o que por sua vez permitirá a geração de imagens cada vez melhores, de forma mais rápida e mais eficaz. Somente esses dados são capazes de nos dar uma real ideia de como cada parâmetro introduzido pelo usuário modifica as imagens e seu processo de captação e reconstrução.

7. Comentário da Orientadora

O aluno realizou um trabalho excelente. Os diversos temas estudados estão relativamente bem descritos no relatório. A descrição das rotinas desenvolvidas deixa um pouco a desejar, no entanto, elas podem ser vistas no Apêndice. O aluno deverá dar continuação a este trabalho no próximo semestre.

8. Referências

8.1. Artigos [1, 2, 3, ...]

- [1] Pruessmann KP, Weiger M, Scheidegger MB, Boesiger P. SENSE: *Sensitivity Encoding for Fast MRI*. *Magnetic Resonance in Medicine* 42:952-962, 1999.
- [2] Roemer PB, Edelstein WA, Hayes CE, Souza SP, Mueller OM. *The NMR phased array*. *Magnetic Resonance in Medicine* 16:192–225, 1990.
- [3] Dydak U, Weiger M, Pruessmann KP, Meier D, Boesiger P. *Sensitivity encoded spectroscopic imaging*. *Magnetic Resonance in Medicine* 46 (4):713-722, 2001.
- [4] Golay X, Pruessmann KP, Weiger M, Crelier GR, Folkers PM, Kollias SS, Boesiger P. *PRESTO-SENSE: An ultra-fast whole brain fMRI technique*. *Magnetic Resonance in Medicine* 43:779-786, 2000.
- [5] Pruessmann KP, Weiger M, Boernert P, Boesiger P. *Advances in sensitivity encoding with arbitrary k-space trajectories*. *Magnetic Resonance in Medicine* 46 (4):638-651, 2001.

8.2. Livros e Sites [A, B, C, ...]

- [A] Huettel SA, Song AW, McCarthy G. *Functional Magnetic Resonance Imaging*. Sinauer Associates, Sunderland, Massachusetts, 2004.
- [B] Liang, Zhi-Pei; Lauterbur, Paul C. *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Wiley-IEEE Press, ISBN 0-7803-4723-4, 1999.
- [C] Buxton, Richard B. *Introduction to Functional Magnetic Resonance Imaging: Principles and Techniques*. Wiley-IEEE Press, ISBN: 0-521-58113-3, 2001.
- [D] Gonzalez, Rafael C.; Woods, Richard E. *Digital Image Processing, Second Edition*. Prentice Hall, ISBN 0-201-18075-8. 2001.
- [E] Bernstein MA, King KF, Zhou XJ . *Handbook of MRI Pulse Sequences*. Elsevier Academic Press, ISBN: 0-120-92861-2, 2004.
- [F] “Matlab Documentation”
<http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?access/helpdesk/help/techdoc/&http://www.mathworks.com/support/product/product.html?product=ML>
- [G] “Academic Earth – Free academic video courses from leading universities” -
<http://academicearth.org/courses/the-fourier-transform-and-its-applications>
- [H] Press WH et al. *Numerical Recipes in C: The Art of Scientific Computing*, 2a ed. Cambridge University Press, 1992. Cap. 12.

Apêndice – Rotinas desenvolvidas

1. handle_list.m

Esta rotina lê os dados do arquivo LIST.

```
InputText=textscan(fid,'%s','delimiter','\n');

Ful_Info=char(InputText{1});

fclose(fid);
clear fid;
clear InputText;

fields = {'mix', 'dyn', 'card', 'echo', ...
          'loca', 'chan', 'extr1', 'extr2', 'ky', 'kz', ...
          'na', 'aver', 'sign', 'rf', 'grad', 'enc', ...
          'rtop', 'rr', 'size', 'offset'};

cont=1;
cont2=1;
cont3=1;
cont4=1;

while cont < size(Ful_Info,1)
    if strncmp(Ful_Info(cont,:), '.', 1) == 1
        Infos(cont2,:)=Ful_Info(cont,:);
        cont2 = cont2 + 1;

        elseif strncmp(Ful_Info(cont,:), 'NOI', 3) == 1
            [crap AUX] = strtok(Ful_Info(cont,:));
            Mapa_NOI(:,cont3)=strread(AUX, '%d',...
            'delimiter', ' ');
            cont3 = cont3 + 1;

            elseif strncmp(Ful_Info(cont,:), 'FRX', 3) == 1
                [crap AUX] = strtok(Ful_Info(cont,:));
                Mapa_FRX(:,cont3)=strread(AUX, '%d',...
                'delimiter', ' ');
                cont3 = cont3 + 1;

                elseif strncmp(Ful_Info(cont,:), 'STD', 3) == 1
                    [crap AUX] = strtok(Ful_Info(cont,:));
                    Mapa_STD(:,cont4)=strread(AUX, '%d',...
                    'delimiter', ' ');
                    cont4 = cont4 + 1;
                end
            cont = cont + 1;
        end

clear ans Ful_Info crap AUX;

Mapa_NOI=transpose(Mapa_NOI);
Mapa_STD=transpose(Mapa_STD);
```

```

Mapa_NOI = struct('mix', Mapa_NOI(:,1), 'dyn', Mapa_NOI(:,2), ...
    'card', Mapa_NOI(:,3), 'echo', Mapa_NOI(:,4), ...
    'loca', Mapa_NOI(:,5), 'chan', Mapa_NOI(:,6), ...
    'extr1', Mapa_NOI(:,7), 'extr2', Mapa_NOI(:,8), ...
    'ky', Mapa_NOI(:,9), 'kz', Mapa_NOI(:,10), ...
    'na', Mapa_NOI(:,11), 'aver', Mapa_NOI(:,12), ...
    'sign', Mapa_NOI(:,13), 'rf', Mapa_NOI(:,14), ...
    'grad', Mapa_NOI(:,15), 'enc', Mapa_NOI(:,16), ...
    'rtop', Mapa_NOI(:,17), 'rr', Mapa_NOI(:,18), ...
    'size', Mapa_NOI(:,19), 'offset', Mapa_NOI(:,20));

Mapa_STD = struct('mix', Mapa_STD(:,1), 'dyn', Mapa_STD(:,2), ...
    'card', Mapa_STD(:,3), 'echo', Mapa_STD(:,4), ...
    'loca', Mapa_STD(:,5), 'chan', Mapa_STD(:,6), ...
    'extr1', Mapa_STD(:,7), 'extr2', Mapa_STD(:,8), ...
    'ky', Mapa_STD(:,9), 'kz', Mapa_STD(:,10), ...
    'na', Mapa_STD(:,11), 'aver', Mapa_STD(:,12), ...
    'sign', Mapa_STD(:,13), 'rf', Mapa_STD(:,14), ...
    'grad', Mapa_STD(:,15), 'enc', Mapa_STD(:,16), ...
    'rtop', Mapa_STD(:,17), 'rr', Mapa_STD(:,18), ...
    'size', Mapa_STD(:,19), 'offset', Mapa_STD(:,20));

clear cont cont2 cont3 cont4;

```

2. handle_infos.m

Organiza os dados do tipo INFO que foram adquiridos na rotina anterior. Ela foi feita a partir de estudos de como são organizados os dados adquiridos pela máquina durante a aquisição dos LIST.

```

contGERAL = 1;

for cont=1:9

    [crap InfosAUX(cont,:)] = strtok(Infos(cont,:), ':');
    contGERAL = contGERAL + 1;

end

number_of_mixes = ...
    str2double(strtok(InfosAUX(1,:), ':'));
number_of_encoding_dimensions = ...
    str2double(strtok(InfosAUX(2,:), ':'));
number_of_dynamic_scans = ...
    str2double(strtok(InfosAUX(3,:), ':'));
number_of_cardiac_phases = ...
    str2double(strtok(InfosAUX(4,:), ':'));
number_of_echoes = ...
    str2double(strtok(InfosAUX(5,:), ':'));
number_of_locations = ...
    str2double(strtok(InfosAUX(6,:), ':'));
number_of_extra_attribute_1_values = ...

```

```

        str2double(strtok(InfosAUX(7,:), ':'));
number_of_extra_attribute_2_values = ...
        str2double(strtok(InfosAUX(8,:), ':'));
number_of_signal_averages = ...
        str2double(strtok(InfosAUX(9,:), ':'));

for cont=1:number_of_locations

    [crap InfosAUX(cont,:)] = strtok(Infos(contGERAL,:), ':');
    number_of_coil_channels(cont,:) = ...
        str2double(strtok(InfosAUX(cont,:), ':'));
    contGERAL = contGERAL + 1;
end

for cont=1:number_of_encoding_dimensions
    [crap InfosAUX(cont,:)] = strtok(Infos(contGERAL,:), ':');
    contGERAL = contGERAL + 1;
end

aux = strtok(InfosAUX(1,:), ':');
[aux aux2] = strtok(aux);
kx_range(1,1) = str2double(aux);
kx_range(1,2) = str2double(aux2);

aux = strtok(InfosAUX(2,:), ':');
[aux aux2] = strtok(aux);
ky_range(1,1) = str2double(aux);
ky_range(1,2) = str2double(aux2);

if(number_of_encoding_dimensions == 3)
    aux = strtok(InfosAUX(3,:), ':');
    [aux aux2] = strtok(aux);
    kz_range(1,1) = str2double(aux);
    kz_range(1,2) = str2double(aux2);
else
    kz_range(1,1) = 0;
    kz_range(1,2) = 0;
end

for cont=1:number_of_encoding_dimensions
    [crap InfosAUX(cont,:)] = strtok(Infos(contGERAL,:), ':');
    contGERAL = contGERAL + 1;
end

kx_oversample_factor = ...
    str2double(strtok(InfosAUX(1,:), ':'));

ky_oversample_factor = ...
    str2double(strtok(InfosAUX(2,:), ':'));

if(number_of_encoding_dimensions == 3)
    kz_oversample_factor = ...
        str2double(strtok(InfosAUX(3,:), ':'));
else
    kz_oversample_factor = 0;
end

```

```

end

if(strcmp(strtok(Infos(contGERAL,:),':'),...
         '0 0 0 number_of_rf_echoes') == 1)
    [crap InfosAUX(1,:)] = strtok(Infos(contGERAL,:), ':');
    number_of_rf_echoes = ...
        str2double(strtok(InfosAUX(1,:), ':'));
    contGERAL = contGERAL + 1;
else
    number_of_rf_echoes = 1;
end

for cont=1:number_of_encoding_dimensions
    [crap InfosAUX(cont,:)] = strtok(Infos(contGERAL,:), ':');
    contGERAL = contGERAL + 1;
end

X_resolution = ...
    str2double(strtok(InfosAUX(1,:), ':'));
Y_resolution = ...
    str2double(strtok(InfosAUX(2,:), ':'));

if(number_of_encoding_dimensions == 3)
    Z_resolution = ...
        str2double(strtok(InfosAUX(3,:), ':'));
else
    Z_resolution = 0;
end

for cont=1:number_of_encoding_dimensions
    [crap InfosAUX(cont,:)] = strtok(Infos(contGERAL,:), ':');
    contGERAL = contGERAL + 1;
end

X_direction_SENSE_factor = ...
    str2double(strtok(InfosAUX(1,:), ':'));

Y_direction_SENSE_factor = ...
    str2double(strtok(InfosAUX(2,:), ':'));

if(number_of_encoding_dimensions == 3)
    Z_direction_SENSE_factor = ...
        str2double(strtok(InfosAUX(3,:), ':'));
else
    Z_direction_SENSE_factor = 0;
end

for cont=1:number_of_encoding_dimensions*number_of_locations
    [crap InfosAUX(cont,:)] = strtok(Infos(contGERAL,:), ':');
    contGERAL = contGERAL + 1;
end

cont=1;

```

```

cont2=1;

while cont<number_of_encoding_dimensions*number_of_locations

    aux = strtok(InfosAUX(cont,:), ':');
    [aux aux2] = strtok(aux);
    X_range(cont2,1) = str2double(aux);
    X_range(cont2,2) = str2double(aux2);

    cont = cont + 1;

    aux = strtok(InfosAUX(cont,:), ':');
    [aux aux2] = strtok(aux);
    Y_range(cont2,1) = str2double(aux);
    Y_range(cont2,2) = str2double(aux2);

    cont = cont + 1;

    if(number_of_encoding_dimensions == 3)
        aux = strtok(InfosAUX(cont,:), ':');
        [aux aux2] = strtok(aux);
        Z_range(cont2,1) = str2double(aux);
        Z_range(cont2,2) = str2double(aux2);

        cont = cont + 1;
    else
        Z_range(cont2,1) = 0;
        Z_range(cont2,2) = 0;
    end

    cont2 = cont2 + 1;
end

for cont=1:2*number_of_locations
    [crap InfosAUX(cont,:)] = strtok(Infos(contGERAL,:), ':');
    X_phase_correctionAUX(cont,:) = ...
        str2double(strtok(InfosAUX(cont,:), ':'));
    contGERAL = contGERAL + 1;
end

X_phase_correction = zeros(number_of_locations, 2);
X_phase_correction(:,1) = X_phase_correctionAUX(1:2:end);
X_phase_correction(:,2) = X_phase_correctionAUX(2:2:end);

clear X_phase_correctionAUX

% esse arquivo ainda não leva em consideração o coilchannelcombination
% for cont=1:number_of_locations
%     [crap InfosAUX(cont,:)] = strtok(Infos(contGERAL,:), ':');
%     coil_channel_combination(cont,:) = ...
%         strtok(InfosAUX(cont,:), ':');
%     contGERAL = contGERAL + 1;
% end

clear cont cont2 contGERAL crap aux aux2 InfosAUX;

```