REPORT
UMN 2012

MEHMET UGURBIL
SUPERVISOR: STEEN MOELLER

1. Grappa Reconstruction

1.1. **Theory.** To increase the data acquisition time, lines may be skipped and then reconstructed later using image reconstruction algorithms. Grappa uses data from around the missing point, in blocks, from all coils to reconstruct missing data in each coil. Mathematically, it can be expressed as $S_j(k_y - m\Delta k_y) = \sum_{l=1}^{L} \sum_{b=0}^{N_b-1} n(j,b,l,m) S_l(k_y - bA\Delta k_y)$, where l is the coils number, b is the block number, A is the acceleration factor, n is the weight function, j is the coil position, y is the y coordinate, and $m = \pm 1, ..., A-1$. The weight function is found by fitting a polynomial to ACS lines.

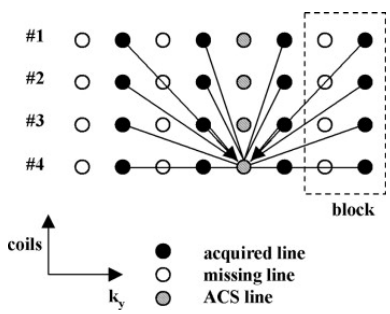

Figure 1. GRAPPA reconstruction[1]

1.2. **Results.** The program, grappa_new, by Steen Moeller was used to create GRAPPA reconstructed images. It was observed that using approximately 92-128 ACS lines for 512 by 512 images gives us the images with least error (lowest g-factor). This is against intuition where we would think that more ACS lines would always result in better weight functions. Different block sizes, images(coronal, sagittal, axial) resolutions, methods of calculating g-factor were used. The results were unchanged. As the number of ACS lines exceed one quarter of the image, more areas without signal are added. This might be causing the weighting function to have greater error while these are the areas with lowest SNR.

Date: 08/20/2012.

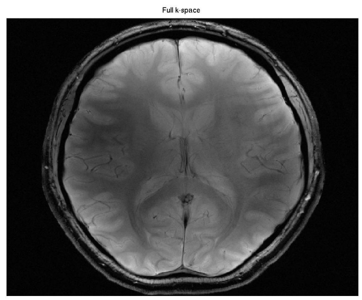

FIGURE 2. RSOS full k-space

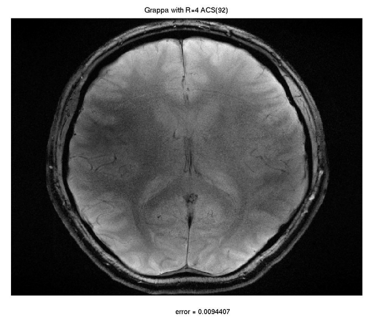

FIGURE 3. Grappa Recon

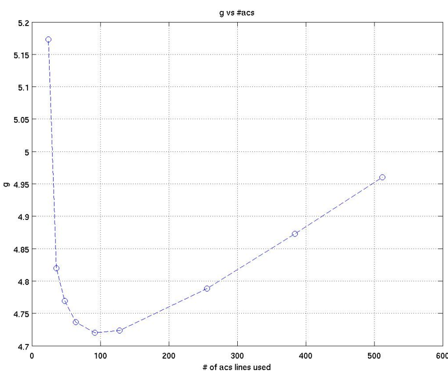

FIGURE 4. g-factor vs. acs lines

2. Coil Compression

2.1. **Theory.** MRI can be done using many coils to increase parallel imaging acceleration, but this creates huge amounts of data causing more computation time. To shorten the amount of computation required, the number of coils could be decreased to a virtual quantity of coils. Because real coil sensitivity functions do not form an orthogonal basis, a smaller set of orthogonal virtual coils could span the same space. Thus, in realigning coils, the goal is to keep as much signal as possible and to minimize the number of virtual coils.

Define the vector $\mathbf{v}(k)$ to represent data from the original coils at each k-space location. Let $A \in \mathtt{C}^{MxN}$ be the coil compression matrix, where N is the number of real coils and M is the number of virtual coils. Then, coil compression can be seen as: $\mathbf{v}'(k) = A\mathbf{v}(k)$. Since both sets of coils approximately span the space, this equation can be changed to $\mathbf{v}(k) = A^H\mathbf{v}'(k) + \epsilon$. Now, the coil compression problem can be turned into a minimization of the compression error: $\underset{A}{\mathtt{minimize}} \sum_k ||(A^H A - I)\mathbf{v}(k)||^2$ $\mathtt{subject\ to}$ $AA^H = I$.

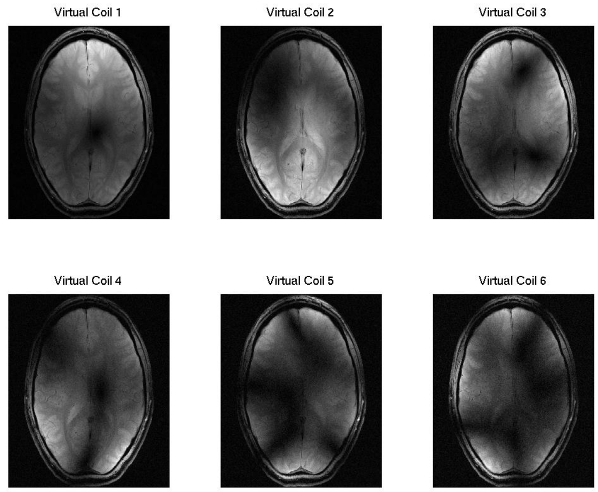

FIGURE 5. Individual Virtual Coil Images

2.2. **Geometric-Decomposition Coil Compression (GCC).** Here x is the readout direction. So k data would be 2D if image is 3D, and 1D if image is 2D.

1.Take Fourier transform along the readout direction.

2.Construct the data matrix X_x where each of its rows consist of all $\mathbf{v}_x(k)$ from one coil. So that X_x has N rows.

3.Perform SVD on X_x: $X_x = U_x \Sigma_x (V_x)^H$. Take the first M rows of $(U_x)^H$ to form compression matrix A_x.

4.Repeat 2 and 3 for all x.

2.3. **Virtual Coil Alignment.** The compression matrices built in GCC are not unique. For any MxM unitary matrix P, $A_x = PA_x^0$ is also a solution. It is likely that the compression matrices created by GCC are not smooth. So, we will call the compression matrices created by GCC A_x^0 and smooth them.

1. Let $A_x = A_x^0$. Then repeat the following steps until you define all A_x's.
2. Define $C_x = A_x^0(A_{x-1}^H)$.
3. Compute SVD of C_x: $C_x = U_x^C \Sigma_x^C (V_x^C)^H$.
4. Let $P_x = V_x^C (U_x^C)^H$ and $A_x = P_x A_x^0$.

2.4. **Commentary.** I followed two ways in compressing the coils for a 2D image. One is compressing the whole image at once in one big matrix while the other way is to take Fourier transform along readout direction and then transforming these subsets. The running time of the algorithms are equivalent for decomposing the matrix in the first method takes as long as the for loop in the second method takes. However, we can align the coil sensitivities to improve SNR and image quality in the second method.

I used standard value decomposition over principle component analysis because the results came out better. PCA does not construct unitary matrices thus rescaling image values. This is not as important in creating the matrices as in virtually smoothing them. Because we need unitary matrices to smooth out the coil sensitivities.

3. Sense reconstruction

3.1. Theory. When we skip lines in cartesian sampling, we are actually folding the image. So, if we have many coils with different sensitivities, we can have different weights on each coil and take a linear combination of the coil data to get an unfolded full image.

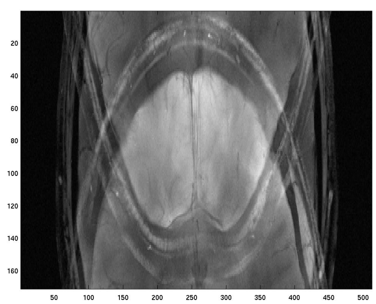

Figure 6. Reduced Data (Folded Image)

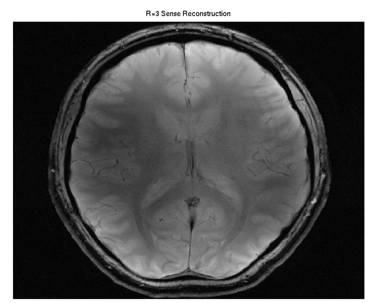

Figure 7. Sense Reconstructed Image A=3

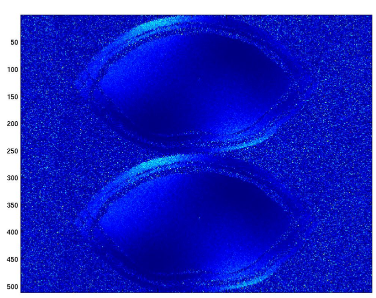

Figure 8. G-factors without smoothing

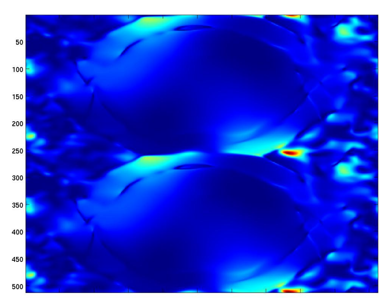

Figure 9. G-factors with smoothing

3.2. **Sensitivity Profiles.** Sensitivity profiles are created by dividing each coil image by the root sum of squares of all the coil images. That way we get a map that is independent of object (brain) structure. This image however is not smooth, to accomplish this we have to filter the image. We can either filter the initial data, or filter the sensitivity maps after they have been calculated. When filtering we need to also fill in the empty space (i.e. between fat tissue and brain tissue). We can do filtering by multiplying the k-space data with a gaussian aligned with the center of k-space.

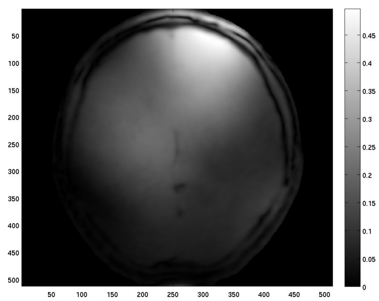

Figure 10. Sensitivity Map

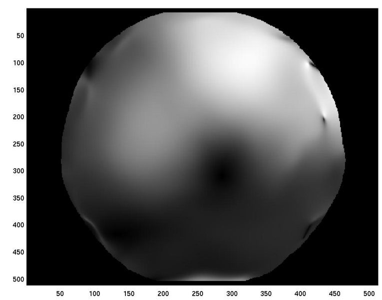

Figure 11. Sensitivity Map with smoothing

4. Masking

We only want the parts of the image where we get signal. So, we mask the image. But there is empty space between fat tissue and brain tissue on this mask, and we want to close it to get complete sensitivity profiles. Therefore we use morphological operations on the logical image(mask). To close the image, we first dilate and then erode the image.

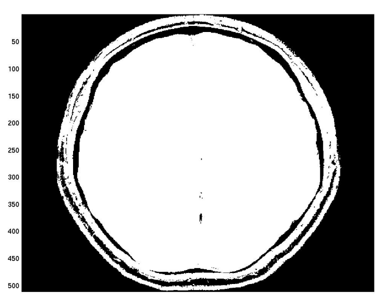

Figure 12. Mask

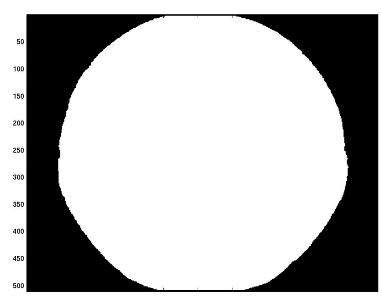

Figure 13. Closed Mask

5. References

[1]Zhang T, Pauly J, Vasanawala S, Lustig M. Coil Compression for Accelerated Imaging with Cartesian Sampling. Magn Reson Med 2012; 000:000-000.

[2]Griswold A, Jakob P, Heidmann R, Nittka M, Jellus V, Wang J, Kiefer B, Haase A. Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA). Magn Reson Med 2002; 47:1202-1210.

[3]Moeller S, Yacoub E, Olman C, Auerbach E, Strupp J, Harel N, Ugurbil K. Multiband Multislice GE-EPI at 7 Tesla, With 16-Fold Acceleration Using Partial Parallel Imaging With Application to High Spatial and Temporal Whole-Brain FMRI. Magn Reson Med 2009; 63:1144-1153.

[4]Pruessmann K, Weiger M, Scheidegger M, Boesiger P. SENSE: Sensitivity Encoding for Fast MRI. Magn Reson Med 1999; 42:952-962.