Self-Calibrating Gridding for 3D Radial Trajectories Using GRAPPA Operator Gridding (GROG)

N. Seiberlich[1], P. Ehses[1], S. Nielles-Vallespin[2], F. A. Breuer[3], M. Blaimer[3], P. M. Jakob[1,3], and M. A. Griswold[4]

[1]Experimental Physics 5, University of Wuerzburg, Wuerzburg, Germany, [2]Siemens AG Medical Solutions, Erlangen, Germany, [3]Research Center Magnetic Resonance Bavaria (MRB), Wuerzburg, Germany, [4]Department of Radiology, University Hospitals of Cleveland, Cleveland, OH, United States

Introduction: GRAPPA Operator Gridding (GROG) [1] has been shown to yield gridded non-Cartesian images comparable to those generated using the gold-standard convolution gridding. GROG works by shifting non-Cartesian points to their nearest Cartesian locations using parallel imaging weight sets; for the two dimensional case, two base weights sets, one for each logical k-space direction, are required. This method is adventageous for 3D gridding due to its low memory requirements, as no grid oversampling is required and an effective 1 x 1 kernel is employed, and can be performed quickly without the need for additional parameters. The necessary GROG weights for 2D gridding can be calculated from the non-Cartesian datapoints for 2D radial and spiral datasets [2] by generating angular weight sets which are themselves combinations of the base weights. This abstract demonstrates that a similar method can be employed to generate the GROG weights for 3D radial data, and this self-calibrating 3D GROG is used to grid an Ultra Short Echo Time (UTE) dataset.

Theory: As in the two dimensional case, adjacent read points along a single ray in the 3D radial dataset can be used to determine an angular weight set, as shown in Figure 1:

$$\vec{S}(\theta,\varphi,k_{r+1}) = \hat{G}_{\theta,\varphi} \cdot \vec{S}(\theta,\varphi,k_r) \quad \text{and thus} \quad \hat{S}(\theta,\varphi,k_{r+1}) \cdot pinv(\hat{S}(\theta,\varphi,k_r)) = \hat{G}_{\theta,\varphi}$$

where θ and φ are the azimuthal and polar angles, respectively, and k_r denotes the radius of the point along that ray. In the first equation, $\vec{S}(\theta,\varphi,k_r)$ is a vector containing the signal values for all coils at the appropriate k-space location, and $\hat{G}_{\theta,\varphi}$ is the weight set describing a shift along this direction; in the second equation, $\hat{S}(\theta,\varphi,k_r)$ is a matrix containing the signal values for all coils for the read points along the ray. The angular weights can be written as a combination of the base weight sets with the appropriate distance relationship:

$$\hat{G}_{\theta,\varphi} = G_x^{\,dx} \cdot G_y^{\,dy} \cdot G_z^{\,dz}$$

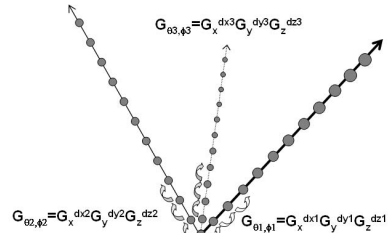

Figure 1: Three rays along a typical 3D radial trajectory. Using the datapoints along a ray in direction [θ,ϕ], the angular weight set $G_{\theta,\phi}$ can be calculated. The GROG weights for the three logical directions can be derived from these angular weights.

where dx, dy, and dz are the Cartesian distances between adjacent read points in the x-, y-, and z-directions respectively. It is important to note that these values must be constant over the portion of the ray used in order to employ this calibration method. The non-linear equation for the angular weight sets cannot be solved in this exponential form; assuming that the GROG weight sets for each logical direction commute and taking the matrix logarithm of the resulting set of equations yields the following equation for each projection, from 1 to P_{max}:

$$\ln(\hat{G}_{\theta1,\varphi1}) = dx_1 \cdot \ln(\hat{G}_x) + dy_1 \cdot \ln(\hat{G}_y) + dz_1 \cdot \ln(\hat{G}_z) \quad \cdots \quad \ln(\hat{G}_{\theta P_{max},\varphi P_{max}}) = dx_{P_{max}} \cdot \ln(\hat{G}_x) + dy_{P_{max}} \cdot \ln(\hat{G}_y) + dz_{P_{max}} \cdot \ln(\hat{G}_z)$$

These equations can be reordered into another set of linear matrix equations, which can be solved coil-by-coil (i.e. a and b run from 1 to N_{coils}):

$$\vec{L}_{a,b} = [\vec{dx},\vec{dy},\vec{dz}] \cdot [\ln(\hat{G}_x(a,b)),\ln(\hat{G}_y(a,b)),\ln(\hat{G}_z(a,b))]$$

where $\vec{L}_{a,b}$ is a vector of size P_{max} containing the matrix elements [a,b] from the matrix logarithms of each of the angular weight sets, the distance matrix of size P_{max} x 3 holds the Cartesian distances for each projection, and the far right-hand term is a vector of size 3 containing the unknowns for the appropriate matrix entries in the logarithms of the GROG weight sets. By taking the pseudo-inverse of the distance matrix, the matrix logarithm elements of the weights sets can be found; the base weights themselves are calculated by taking the matrix exponential of the logarithmic weights. GROG can then be performed as in the two dimensional case using the three weight sets along the logical Cartesian axes using these weight sets.

Methods: The 3D self-calibrating GROG method is demonstrated here for the UTE trajectory. In vivo UTE datasets (32 channel, 16384 projections, 128 read-out points, base matrix 128x128x128) were acquired using a 3 T clinical scanner (Tim Trio, Siemens Medical Solutions, Erlangen, Germany). Because the first 30 points of the trajectory were sampled during the gradient ramp up time and did not have the same Cartesian spacing as the other points along a given ray, the ramp-sampled points were not included in the GROG calibration. The base weights G_x, G_y, and G_z were calculated using the formulas given above, and the data were gridded using a version of GROG modified to shift points in three dimensions. Once the data were gridded, a Fourier transform followed by an adaptive recontruction [3] was performed to yield the final images.

Results: Four example slices from the UTE dataset gridded using the 3D radial self calibrating GROG method are shown in Figure 2. The 32-channel head coil delivers high signal at the edges of the head; these details can clearly be seen in the GROG images. The gridding time was approximately 8 minutes using unoptimized Matlab code, considerably less than the 20 hours required for an INNG [4] implementation on the same computer.

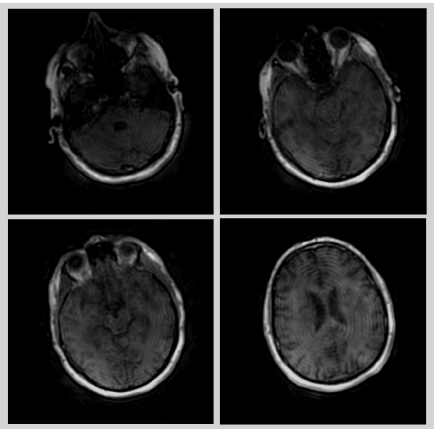

Figure 2: Example images from the UTE dataset gridded using the self-calibrating 3D GROG method.

Discussion: The SC-GROG method, previously demonstrated for 2D radial and spiral images, can also be employed to calculate the GROG weights needed to grid three dimensional radial data. This method does not require a density compensation function, convolution window, or additional gridding parameters besides the 3D k-space trajectory. Unlike in convolution gridding, GROG does not employ grid oversampling or large convolution kernels and thus has less demanding memory requirements. In addition, this method is more efficient than iterative methods which require two Fourier transformations per iteration, a time-consuming prospect for a large 32-channel dataset. In conclusion, the 3D SC-GROG method depicted here can be used as a fast, simple alternative to standard gridding methods given a receiver coil with sensitivity variations in three logical directions.

References

1 Seiberlich N et al. MRM 2007 Oct 29. 2 Seiberlich N et al. Proc. ISMRM 2007, pg. 153.
3 Walsh DO et al. MRM 2000 May;43(5):682-90. 4 Moriguchi H et al. MRM 2004 Feb;51(2):343-52.

Acknowledgements: The authors would like to thank Siemens AG Medical Solutions and the Deutsche Forschungsgemeinschaft project JA 827/4-4 for support.