Fast Image Reconstruction from Non-Cartesian Data

Philip J. Beatty
MR Applied Science Laboratory
GE Healthcare
Menlo Park, California
Email: philip.beatty@ge.com

I. INTRODUCTION

In this session, I will focus on the gridding method for fast image reconstruction from non-Cartesian data [1-5]. While gridding is not the only game in town, it is the reconstruction of choice when reconstruction time is an issue. Other reconstruction techniques include rBURS [6] and non-Cartesian iterative methods [7,8]. Non-Cartesian iterative methods generally use gridding and inverse gridding as their core operation, so even when gridding is not the right solution for an application, an understanding of gridding can be useful.

I will cover the main steps of the gridding algorithm and then discuss some implementation tips that can significantly reduce the computation time. Finally, I will present some illustrations and viewpoints that I have found to aid in an intuitive understanding of the gridding method, which can be helpful when designing the density compensation function and judging the appropriateness of gridding as a solution for a particular application.

II. THE GRIDDING ALGORITHM

The gridding algorithm consists of the following steps:

1) Multiply data with a set of density compensation weights.
2) Convolve with a gridding kernel.
3) Sample onto uniformly spaced sample locations (the grid).
4) Perform FFT.
5) Trim the Field-of-View (FOV).
6) Multiply by an apodization correction function.

III. TIPS FOR FAST GRIDDING

One of the main advantages of gridding is that it can be performed very quickly, requiring a computation time of a comparable order-of-magnitude to the FFT for Cartesian data sets. However, there can be greater than an order-of-magnitude difference in computation time between gridding implementations; the devil is in the details. Here are a few tips that should help in efficient gridding implementations.

1) Use a compiled language like C; ‘for’ loops in MATLAB are significantly slower.
2) Use a minimal oversampling ratio [5]. I typically use 1.25X or 1.5X, depending on the situation. This is especially important for 3-D data sets, where memory is a concern.
3) Presample your gridding kernel and use a look-up-table (LUT) [5,9].
4) For 3-D data sets, use a ‘block’ data structure to store the grid [10].
5) Perform 1-D FFTs separately in each direction and trim the FOV after each direction.
6) When repeatedly gridding the same k-space trajectory, consider sorting data by k-space location.
IV. A DEEPER UNDERSTANDING

Gridding is a fast and accurate approximation to the weighted Inverse Discrete Fourier Transform (IDFT), also known as conjugate phase reconstruction [11]. The weighted IDFT can be expressed mathematically as:

\[ \hat{m}(r) = \sum_{i=1}^{N_s} w(k_i) d(k_i) e^{j2\pi(k_i \cdot r)} , \]

where \( \hat{m}(r) \) is the reconstructed image of the magnetization as a function of spatial location \( r \), \( w(k_i) \) is known as the weighting function or ‘density compensation function’ and \( d(k_i) \) is the value of the datum acquired at k-space location \( k_i \), where \( i \) indexes the \( N_s \) sampled k-space locations.

The difference between gridding and the weighted IDFT can be made arbitrarily small by varying the gridding parameters (oversampling ratio and gridding kernel), trading off accuracy for computation time [3]. Typically, image artifacts seen in a gridding reconstruction will be nearly identical to those in the weighted IDFT, so it is worth studying the weighted IDFT to better understand the behaviour of gridding. In considering gridding for a particular application, one must design the density compensation function and assess the suitability of the k-space trajectory. I have found three viewpoints of the weighted IDFT operation very useful in developing an intuition regarding these considerations: 1) as a Riemann Sum, 2) as a matrix operation and 3) as convolution with a point-spread-function.

The weighted IDFT can be viewed as a Riemann Sum, approximating the Inverse Fourier Transform integral that relates k-space and image space:

\[ \hat{m}(r) = \sum_{i=1}^{N_s} d(k_i) e^{j2\pi(k_i \cdot r)} \Delta k_i \approx \int d(k) e^{j2\pi(k \cdot r)} dk = m(r). \]

In this case the density compensation function \( w(k_i) \) is replaced by the partition \( \Delta k_i \). This viewpoint leads to the technique of computing the density compensation function using a Voronoi diagram [12].

To view the weighted IDFT as a matrix operation, the voxels of the reconstructed image can be organized into a vector \( \hat{m} \) and the weighted IDFT can be written as

\[ \hat{m} = E^H W d , \]

where \( E^H \) is the Hermitian conjugate of the gradient encoding matrix \( E \) (\( E_{\text{row}, \text{col}} = e^{j2\pi k_{\text{row}} r_{\text{col}}} \)), \( W \) is a diagonal matrix that contains the density compensation weights and \( d \) is a vector containing the acquired data values. Since

\[ d = E m , \]

the weighted IDFT should produce an accurate reconstruction when \( E^H W \approx E^\dagger \), a pseudo-inverse of \( E \). The matrix viewpoint leads to another approach for computing the density compensation function [13]. The matrix operation viewpoint makes it easy to spot many trajectories that simply won’t reconstruct well with gridding—when no diagonal matrix \( W \) can be found that will make \( E^H W \approx E^\dagger \). This viewpoint also shows why gridding is the building block
for many iterative reconstruction techniques for non-Cartesian image reconstruction: without
density compensation, gridding can be used to multiply a vector by $E^H$. Likewise, inverse-
gridding can be used to efficiently multiply a vector by the encoding matrix $E$.

As a convolution with a point-spread function, the reconstructed image can be expressed as

$$\hat{m}(r) = m(r) \ast \text{PSF}(r).$$

The point-spread-function, $\text{PSF}(r)$ is equal to the Fourier transform of the density compensated sampling function of the k-space trajectory. From this viewpoint, gridding works well when the PSF approximates a delta function, having a large central peak at $r = 0$ and very low amplitude for a radius corresponding to the desired field-of-view. The PSF viewpoint also leads to approaches for computing the DCF, exemplified by [14]. The PSF gives a very graphical view of the attributes of the reconstructed image. One of the aspects of this viewpoint that I really like is that it brings out the trade-offs inherent in the design of the density compensation function. By designing the density compensation function to suppress the side lobes of the PSF, at the expense of a slightly wider central peak, image resolution is traded off for a reduction in Gibbs ringing. So, what does it mean to have an ‘optimal’ density compensation function?

V. Summary

This ‘sunrise session’ is early in the morning and there is a beautiful beach outside, but now is your chance to become an expert in gridding. By the end of this session you should understand how gridding works, how to implement fast gridding and having an intuitive understanding of the strengths and weaknesses of the gridding method.

VI. References

1) J. O’Sullivan, “A fast sinc function gridding algorithm for Fourier inversion in computer
2) J. Jackson, C. Meyer, D. Nishimura, and A. Macovski, “Selection of a convolution function
4) H. Schomberg and J. Timmer, “The gridding method for image reconstruction by Fourier
5) P. Beatty, D. Nishimura, and J. Pauly, “Rapid gridding reconstruction with a minimal
6) D. Rosenfeld, “New approach to gridding using regularization and estimation theory,”
7) T. Harshbarger and D. Twieg, “Iterative reconstruction of single-shot spiral MRI with off
8) B. Sutton, D. Noll, and J. Fessler, “Fast, iterative image reconstruction for MRI in the
10) P. Beatty, J. Pauly, D. Nishimura, “Rapid 3D non-Cartesian SENSE reconstruction,” ISMRM
