

Optimized Algorithm for Sampling Density Correction

K. O. Johnson¹, J. G. Pipe¹

¹MRI Research, St. Joseph's Hospital, Phoenix, Arizona, United States

Introduction Non-Cartesian k-space trajectories have become commonplace in imaging. A common approach to reconstruction has been by use of gridding [1] data onto a Cartesian grid. The first step in the gridding algorithm corrects the non-uniform sampling density that is encountered. This is done by weighting each of the data samples by a factor, called density compensation factors (DCF), which are inversely proportional to the sampling density at that point. Among the several approaches to estimating these weighting factors is a simple approach proposed in [1]. The sampling density ρ is estimated as the sampling function S is convolved with the kernel used in the gridding process C , which is expressed as $\rho = S * C$ and therefore when evaluated at sample t the weight $w(t)$ is $w(t) = 1/\rho(t) = 1/[S * C]_t$. An iterative version of this was proposed in [2], which repeats this step recursively and is expressed as $w_{i+1} = \frac{w_i}{[(w_i \cdot S) * C]_t}$ where

$w_0 = 1$ and therefore w_1 is the same as the implementation described in [1]. Unfortunately for a dataset of N points and q iterations, this requires processing time of $O(q \cdot N^2)$. Despite state of the art hardware, this amount of time becomes clinically unfeasible for larger datasets especially in cases such as propeller where the coefficients need to be recalculated for every image. Therefore a method for optimizing this approach while preserving its mathematical integrity is proposed.

Theory and Methods Typically in light of computation speed, a finite extent kernel C can be assumed. When this is true, a large number of the points can be discarded when evaluating the convolution at a given sample. In this implementation the entire set of points is divided up into multiple compartments. This results in a 2-D array of compartments, which equally divide k-space. Each compartment contains the points that would fall into its respective k-space area. Due to compartmentalization, only a subset of the points need to be searched to calculate the convolution for density approximation. For the instance that the width of each compartment matches the width of the convolution kernel C then only four compartments of the whole set need to be searched, as is visible in figure 1. Therefore the operation time is proportional to the total number of points times the number of points per compartment n i.e. $O(q \cdot 4n \cdot N) = O(q \cdot n \cdot N)$, where $n \ll N$.

The major bottleneck with current hardware on this algorithm was noted to be the memory access time coupled with the kernel complexity. Therefore in order to reduce both the number of memory accesses and the number of kernel computations a simple approach of 'forward only' sequence was implemented. When the contribution of a neighboring point a_n is added to the convolution sum of a given point a_i , the contribution of point a_i can be added to the convolution of a_n with minimal additional computation. Therefore as the catalog of points is traversed, the algorithm need work in a 'forward only' form as illustrated in figure 2. This results, on average, of only half of the number of compartments to be traversed. Care was also taken in order to preserve adjacent memory allocation (with respect to searches).

Results and Discussion Several cases are presented in the table. Tests were conducted in C on a Xeon 3.2 GHz machine. The proposed implementation of the cited algorithms to sampling density correction dramatically reduce computation time and make it possible for calculation of the densities in an efficient manner yet while preserving the mathematical integrity of the algorithms. Minimal additional work would be necessary in order to thread the application for use on the ever-increasing multiprocessor systems, thus providing hypothetically a proportional speedup.

References [1] IEEE Trans Med Imag 10 473, [2] Mag Res Med 41:179

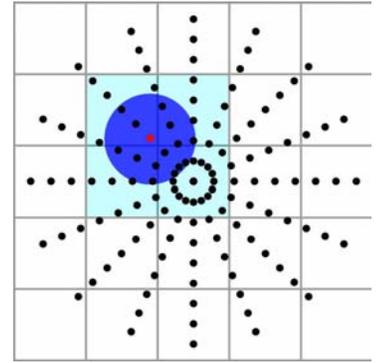


Figure 1: Only the cyan compartments need to be traversed when calculating the convolution for the red point.

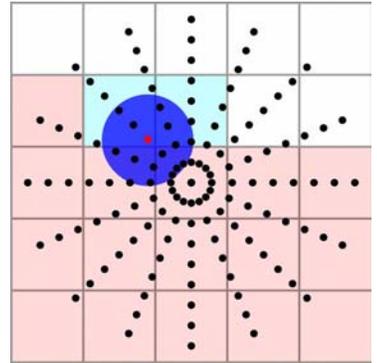


Figure 2: In the forward only mode the previously computed compartments (red) need not be traversed. Only the cyan compartments need to be traversed.

	Single-Shot Spiral	Propeller	3-D Spiral Projection
Effective Matrix	80x80	192x192	300x300x300
Number of Points	8,192	291,840	5,875,200
Number of Compartments	692	1662	2593
Iterations	10	1	1
Computation time optimized (sec)	0.06	0.73	58.24
Computation time unoptimized	5.21	664	* 343,706 ~ 4 days

Table 1: Computational time for the optimized implementation versus the standard unoptimized implementation for various different images. Note that the 3-D unoptimized test was estimated by running a fraction of the program.