

Rapid, Embeddable Design Method for Spiral Magnetic Resonance Image Reconstruction Resampling Kernels

Christopher Kumar Anand, Tamás Terlaky, Bixiang Wang
McMaster University

17 November 2003

Abstract. After formulating the design problem for Resampling Kernels used in Magnetic Resonance Spiral Image Reconstruction, we show that an iterative Gauss-Seidel-type interior-point optimization method is suitable (fast and light-weight) for embedded uses. In contrast to previous practice, we directly optimize a computationally efficient, piecewise-linear kernel rather than an analytic function (Kaiser-Bessel). We also optimize our kernels for worst-case (infinity-norm) signal aliasing, rather than the usual proxy energy function (2-norm) minimization. In numerical simulations of undesirable near-frequency systematic noise the new kernel significantly outperforms a conventional Kaiser-Bessel-based solution.

Keywords: magnetic resonance imaging, spiral imaging, regriding, non-uniform Fourier transform, interior point method

1. Introduction:

We discuss the design of continuous resampling kernels, functions from an interval to the real line, which are commonly used for nonuniform Fourier transforms, as they occur in magnetic resonance image (MRI) reconstruction software. Nonuniform Fourier transforms occur in other applications (nonuniform time series, and astronomy, see (Aldroubi and Gröchenig, 2001; Marvasti, 2000)), but we frame the problem in terms of the MR requirements.

Our interest in resampling kernels arises from a larger research program in high-performance signal processing. As a test problem, we are implementing a real-time MRI reconstruction software infrastructure implementing algorithms not currently feasible for real-time reconstruction, including corrections for systematic measurement errors, object inhomogeneities, and acquisition-time reduction strategies requiring constrained reconstruction. To do this, we need computationally simpler kernel designs than currently used, and in a future publication we will show that the kernel designed in this paper can be implemented for two dimensional spiral reconstruction entirely in vector registers of a PowerPC processor. Since on current-generation hardware, this algorithm is band-width constrained, this is a significant performance enhancement.

The kernels we design are optimal for the image and transform sizes chosen. Since these sizes vary with the image acquisition, which will be different from scan to scan, one has the choice of designing off-line and accepting sub-optimal results most of the time, predicting a range of possible imaging parameters and generating a family of resampling kernels which is close to optimal for all parameters in this range, or embedding the optimiser into the imaging device. The last solution is simpler from a software-engineering point of view, although more challenging from an algorithmic point of view. We took on the



Figure 1. Example of an optimized $2m = 16$ -segment, width 4 piecewise linear kernel. This kernel minimizes the energy of the six near-spectrum aliased images relative to the energy of the image.

challenge of producing a portable algorithm which depends only on standard linear algebra functions available for all architectures from which one would consider building an image reconstructor. Surprisingly, the method we developed was also more robust than a leading commercial non-linear optimizer.

Since many readers will be familiar with convolution filter design problems in digital signal processing, especially in communications, we will contrast them with the present problem. In the present problem as in the design of FIR filters (see (Hamming, 1983) for details), a kernel function must be designed with constraints and objective depending jointly on the function and its' Fourier transform. In standard digital signal processing, one-dimensional signals are sampled at equal intervals, so the function is essentially a vector, and the kernel function is discrete.

In the current problem, sampling is irregular and covers two or three dimensions. The required kernel function is therefore not discrete, but is multi-dimensional, both of which increase the implementation cost, restricting the possible practical kernel designs. These restrictions are balanced by the ability to correct kernel defects in reconstruction stages following the Fourier transform. This correction ability has no analogues in more common digital filtering. Our objective function is also nonlinear, while FIR filter design is a linear problem.

In order to fix notation, we review the fundamentals of imaging and resampling theory required to define our problem. This is standard material which is well-described in signal processing (Hamming, 1983) and MR (Liang and Lauterbur, 2000; Haacke et al., 1999) textbooks.

Let \mathbb{C} denote the set of complex numbers; let V be an n -dimensional real vector space and \hat{V} its dual; and let $s : V \rightarrow \mathbb{C}$ represent the signal we want to reconstruct. Let $g : \hat{V} \rightarrow \mathbb{C}$ represent the Fourier transform of that signal, which we sample. If we sample g at a finite set, $\mathcal{G} \subset \hat{V}$, of frequencies, the sampled function \tilde{g} is a sum:

$$\tilde{g} = \sum_{s \in \mathcal{G}} \delta_s \cdot g.$$

If \mathcal{G} is a finite rectangular subset of a lattice $Z^n \subset \hat{V}$, then a (fast) discrete Fourier transform can be used to reconstruct a sampled version of s from \tilde{g} . If the signal is contaminated with (uncorrelated, identically-distributed Gaussian) noise, this reconstruction minimizes the L^2 error in the sampled version of s .

If we sample on an arbitrary locus, we can also reconstruct a sampled version of s , by solving an inverse problem using a Moore-Penrose pseudo-inverse, or an iterative method,

but these are computationally expensive, since the matrix sizes involved are on the order of 10^5 . A much faster approach is to resample \tilde{g} onto a lattice and use a FFT, replacing the expensive matrix operations with the product of a sparse operation and a well-factored one. The simplest approach is to assign to each lattice point the value of \tilde{g} from the nearest sampled point. This is a variant of nearest neighbour resampling. Alternatively, we could interpolate from the sampled values to the lattice points. There are two ways of approaching interpolation: start with a lattice point and look for neighbouring samples or start with a sample and redistribute its value to neighbouring lattice points. The latter fits more conveniently into Fourier Theory, because it can be factored into two linear steps: convolving the sampled function \tilde{g} with a continuous kernel, f , see *e.g.*, Figure 1, and resampling the continuous function at the lattice points. The Fourier transform of this function can be computed as

$$\text{image} = FT^{-1} \left(\sum_{i \in \mathbb{Z}^n} \delta_i \cdot (f \star \tilde{g}) \right) \quad (1)$$

$$= FT^{-1} \left(\sum_{i \in \mathbb{Z}^n} \delta_i \right) \star (FT^{-1}(f) \cdot FT^{-1}(\tilde{g})) \quad (2)$$

$$= \text{effect of sampling} \star \left(FT^{-1}(f) \cdot \left(FT^{-1} \left(\sum_{s \in \mathcal{G}} \delta_s \right) \star FT^{-1}(g) \right) \right). \quad (3)$$

Sampling on a rectangular subset of a lattice has the effects of 1) introducing ringing, and 2) causing frequencies above the Nyquist frequency to alias (wrap around) onto adequately sampled frequencies. We will ignore the ringing, since this is an orthogonal problem, with known mitigating strategies. (See (Hamming, 1983) for a full discussion.) The aliasing can be ameliorated with a suitable choice of f . Aliasing typically occurs for four reasons:

1. the desired reconstructed-image size is smaller than the object being imaged;
2. the noise is broader in spectrum than the image, and aliases onto the image;
3. the non-uniformity of the sampling locus introduces higher-frequency artifacts (this is a complicated form of aliasing);
4. there is a systematic noise source outside of the imaging band which aliases into the image.

The last source includes the various clock signals present in all digital electronics (including the imager).

By choosing f such that $FT^{-1}(f)$, the *transfer function*, is large in the central image interval (the pass band) and small in the intervals which will alias onto the central image interval, we can significantly reduce all of these effects. (See Figure 2.) As in the representative example, we cannot expect the transfer function to be constant in the image window. Significant roll-off at the boundary is likely, and left uncorrected, this would

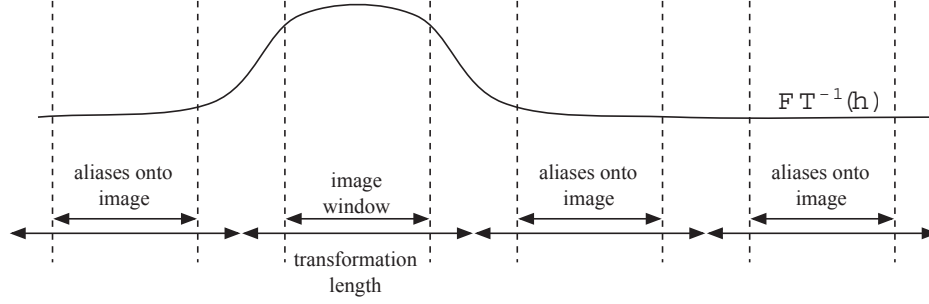


Figure 2. The transform of the kernel function, f , showing the desired imaging intervals as well as the intervals which will alias onto the image as a result of using the finite Fourier transform.

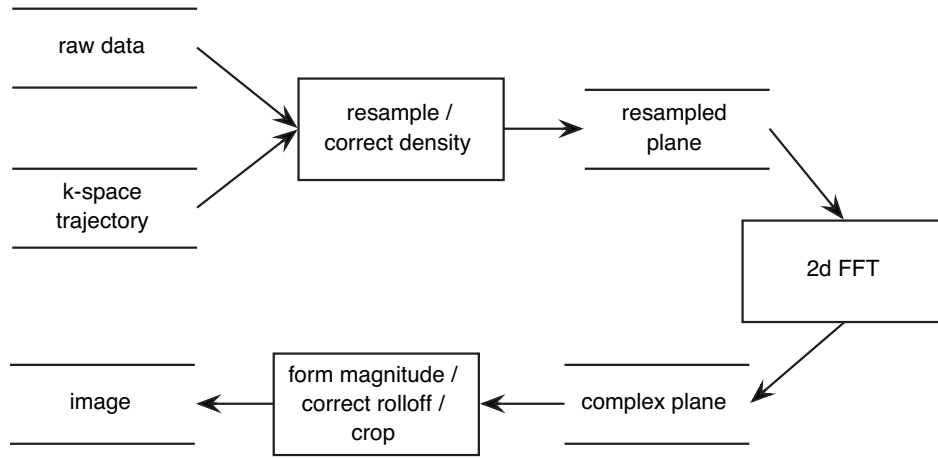


Figure 3. Dataflow of a non-raster image reconstruction including resampling.

lead to darkening of the images around the edges. As long as the roll-off does not result in numerical underflow, however, it can be corrected by dividing the transformed image by the transfer function. (See the Figure 3.) But this normalization effects the aliased component of the image equally, so the correct measurement of the alias reduction is

$$\max_{p \in \mathcal{A}} \frac{|FT^{-1}(f)(p)|}{FT^{-1}(f)(\iota(p))}, \quad (4)$$

where \mathcal{A} is the set of points which alias onto the image window and $\iota(p)$ is the point in the image window onto which p aliases. Note the importance of the L^∞ norm since we cannot trade off image quality from one image location to another. The set \mathcal{A} is an infinite union of intervals. The intervals are usually referred to as stop bands, and the image window as the pass-band. For practical purposes, we only need to consider a finite number of stop bands. So-called white noise never contains equal volume at all frequencies, and sampled signals

are naturally low-pass filtered, passively by the mechanical/optical/electrical properties of the sampling device, and actively by analogue and digital decimating filters. In rapid imaging applications, the effectiveness of alias reduction outside the first stop-bands is also limited by the smallest ratio between the nonuniform sampling rate and the uniform resampling rate, which is typically less than three.

Relationship with other work. In the literature, *e.g.*, (Jackson et al., 1991), this objective is replaced by one of a number of equivalent formulations in terms of L^2 norms that are much easier to optimize:

$$\frac{\int_{p \in \mathcal{A}} (FT^{-1}(f)(p))^2}{\int_{p \in \mathcal{A}} (FT^{-1}(f)(\iota(p)))^2}. \quad (5)$$

We note that for simple kernel shapes such as the one we treat in this paper, it is possible to use a software package such as MAPLE to determine the L^2 norm analytically, and symbolically calculate enough derivatives to minimize it using Newton's method. This problem was solved for discrete functions by Jackson et al (Jackson et al., 1991) using cyclic projections onto convex sets, requiring millions of Fourier transforms, and minutes of computation. Previous to Jackson et al, the problem was further simplified by minimizing the L^2 norm on the complement of the image band, which produces even less desirable results. The analytic function which is close to optimal in this sense is the Kaiser-Bessel kernel, which is still commonly used in practice, and in theoretical work on orthogonal aspects of non-raster image reconstruction, as listed below.

The reason we reformulated the problem using the L^∞ -norm is that the L^2 -optimized transfer function has uneven aliasing protection, suppressing aliasing in the interior of the image preferentially. Although the present work is the first to use an L^∞ formulation of this problem, Fessler and Sutton (2003) also minimize the maximum error in a different formulation of spiral imaging problem.

In the work of Jackson et al and in other previous work, the non-linear interaction of aliasing and roll-off correction was simplified to allow solution by L^2 methods. Since that paper, spiral reconstruction research has focussed on the orthogonal problem of how to best correct for sampling density variations, following the work of Pipe and Menon (Pipe and Menon, 1999). Sedarat and Nishimura, Rasche et al and Wajer et al, in a series of papers, (Sedarat and Nishimura, 2000), (Rasche et al, 1999), (Wajer et al, 2000; Wajer et al, 2001a; Wajer et al, 2001b), have put this work into a large framework, and proposed additional methods of improving image reconstruction by varying the sampling locus and density correction.

Another line of research, started by Rosenfeld (Rosenfeld, 1988), tries to solve the inverse problem associated with non-uniform sampling without using resampling to factor the problem through the fast Fourier transform. This is also an active area of research, with different approaches based on singular value decomposition, and regularized approaches based on truncated decompositions. Recent work on non-shift-invariant resampling, (Fahmy et al, 2003) and (Fessler and Sutton, 2003), can be seen as putting the two approaches together.

In summary, research into reconstruction of non-uniformly sampled MRI data is quite active. The method of directly optimizing numerically realizable kernel shapes should be combined with the other approaches in the future. In this paper we will focus on the case of numerically efficient kernels with remarkably good alias protection properties. The novelty in this paper lies in the use of the L^∞ norm, the nonlinear model (4) which takes the spatially varying effect of roll-off correction on aliasing, and the requirement that resampling kernels be optimized in real-time for the specific imaging parameters of each acquisition, using an embeddable optimizer.

In the body of the paper, we formulate the optimization problem concretely for the class of continuous, symmetric, piecewise-linear, equal-segment kernels. We show that, in the representative case of 16-segment, width-four kernels, directly optimizing this kernel shape provides significantly better aliasing suppression than approximated Kaiser-Bessel kernels, and discuss the results of optimizing with commercial non-linear solvers and our own iterative interior-point method which is fast and small enough for embedded applications.

2. Optimization problems

The design of the kernel function is similar to FIR filter design. In both cases we seek a function with compact support whose Fourier transform is essentially bounded. The domain of the resampling kernel, however, is an interval and not a discrete set of points. It is common practice to use a width four, Kaiser-Bessel (KB) kernel (Harris, 1978; Boernert, 2001), because this function is readily calculable and has a very small amount of energy outside of the passband for a range of kernel widths, but for practical purposes we approximate this function by a piecewise polynomial function. The effects of this approximation have not been analyzed. We show that directly optimizing the piecewise polynomial function produces better results, and discuss the choices of optimization techniques to produce optimal results reliably and efficiently.

To streamline the presentation, we will fix some design parameters:

- symmetric, width- $2l$, continuous, piecewise-linear kernels with $2m$ segments (see Figure 4),
- minimizing aliasing from the closest six aliasing bands,
- and where not specified otherwise, transform length is twice the image length.

We will show that optimized kernels significantly outperform approximate KB kernels when $2m = 16$ and $2l = 4$, and that this granularity is sufficient to produce excellent unwanted signal isolation. All of the techniques we discuss would be applicable to other kernel shapes, *e.g.*, higher-order uniform or non-uniform B-splines, and discontinuous piecewise-polynomial functions. (We did not investigate higher-order or non-uniform splines since the simplest splines were found to give adequate performance.) This shape is rich enough to deliver the necessary level of performance, yet simple enough to result in very efficient implementations on modern processors (using vector instructions such as AltiVec). In the

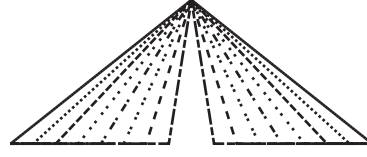


Figure 4. A basis for the set of symmetric, width-four, continuous, piecewise-linear kernels with $2m = 16$ segments.

case of MR image reconstruction, it almost eliminates the additional computational load attached to spiral and other non-raster 2d imaging, and significantly reduces that of 3d imaging.

The auto-convolution of a step function is a triangular function of twice the width, the set of m such symmetric functions form a basis for the set of symmetric, width-four, continuous, piecewise-linear kernels with $2m$ segments (see also Figure 4):

$$f_j(x) = \begin{cases} \frac{m^2}{j^2 l^2} \left| |x| - \frac{jl}{m} \right| & \text{if } |x| < \frac{jl}{m} \\ 0 & \text{otherwise} \end{cases}; \quad 1 \leq j \leq m.$$

Their Fourier transforms are

$$\hat{f}_j(x) = \left(\text{sinc} \left(\frac{\pi j l x}{m} \right) \right)^2, \quad (6)$$

where $\text{sinc}(x) = \sin(x)/x$.

In terms of this basis, our kernel function is

$$f = \sum_{j=1}^m a_j f_j$$

with transfer function

$$\hat{f} = \sum_{j=1}^m a_j \hat{f}_j.$$

Then expression (4) can be written concretely in the form:

$$\min_{a \in \mathbb{R}^m} \max_{1 \leq n \leq d} \max_{t \in [-w/2, w/2]} \frac{|\sum_{j=1}^m a_j \hat{f}_j(t+n)|}{\sum_{j=1}^m a_j \hat{f}_j(t)} \quad (7)$$

subject to the constraints

$$\sum_{j=1}^m a_j \hat{f}_j(t) > 0, \quad \forall t \in [-w/2, w/2], \quad (8)$$

where d is the number of aliasing bands on each side of the imaging window, w is the ratio of imaging band size to transform size, and a_1, \dots, a_m are decision variables.

Note that this is a continuous semi-infinite optimization problem because the variable t belongs to the interval $[-w/2, w/2]$. In order to solve the problem, we first have to discretize this interval and transform the continuous problem into the corresponding discretized version. Assume that the interval $[-w/2, w/2]$ is discretized with $2N + 1$ equally spaced points ($N > 0$), that is, the interval $[-w/2, w/2]$ is divided into $2N$ subintervals of the same length, with N subintervals on each side of the origin. Then the discretized optimization problem is given as:

$$\min_{a \in \mathbb{R}^m} \max_{1 \leq n \leq d} \max_{-N \leq i \leq N} \frac{|\sum_{j=1}^m a_j \hat{f}_j(\frac{w}{2N}i + n)|}{\sum_{j=1}^m a_j \hat{f}_j(\frac{w}{2N}i)} \quad (9)$$

subject to the constraints

$$\sum_{j=1}^m a_j \hat{f}_j\left(\frac{wi}{2N}\right) > 0, \quad 0 \leq i \leq N. \quad (10)$$

We note that the above constraint, the reality of the kernel, and symmetry of the Fourier transform imply that $\sum_{j=1}^m a_j \hat{f}_j\left(\frac{iw}{2N}\right) > 0$ for all $-N \leq i \leq N$. Since the objective is independent of scaling in a , we fix this scale by adding a constraint

$$\sum_{j=1}^m a_j = 1. \quad (11)$$

All of our illustrations will consider the case $m = 8$, $l = 2$, $d = 3$ and $w = \frac{1}{2}$, which corresponds to sixteen segment, width-four kernels, optimized to reduce aliasing from the closest six aliasing bands, since this is an applicable case. We mention that the above problem (9) with constraints (10) and (11) is a nonlinear, non-convex optimization problem.

3. Optimization Methods

In this section, we present three methods for solving the optimization problem (9)–(11). The three methods can be described as follows:

1. **Nonlinear Model:** Reformulate the problem to remove the absolute values and use the nonlinear solver LOQO (Vanderbei, 1999) to calculate the optimal solution. However, as shown in the next section, this approach takes more iterations to solve this non-convex nonlinear problem directly and sometimes fails to produce the correct solution.
2. **Linear Model:** Approximate the nonlinear problem by a linear optimization problem, and solve the linear problem by an interior point method. The linear problem is easy to solve, but it is not the original problem we want to solve.

3. Iterative Scheme: Develop an iterative scheme to solve the nonlinear problem (9)–(11) successively, by solving a sequence of linear optimization problems approximately.

Nonlinear Model: Now, let us look at the first method. In order to apply the software LOQO to solve the problem, we introduce a linear objective and split expression (9) into two parts without an absolute value. Set

$$\tau = \max_{1 \leq n \leq d} \max_{-N \leq i \leq N} \frac{|\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right)|}{\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right)}. \quad (12)$$

Then, taking (10) into account, the problem (9) is equivalent to the following:

$$\min \tau$$

subject to

$$\left| \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) \right| \leq \tau \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right), \quad -N \leq i \leq N, 1 \leq n \leq d, .$$

Thus, the complete discretized nonlinear model can be written as

$$\min \tau \quad (13)$$

subject to the constraints

$$\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) \leq \tau \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right), \quad -N \leq i \leq N, 1 \leq n \leq d, \quad (14)$$

$$-\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) \leq \tau \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right), \quad -N \leq i \leq N, 1 \leq n \leq d, \quad (15)$$

$$\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right) > 0, \quad 0 \leq i \leq N, \quad (16)$$

$$\sum_{j=1}^m a_j = 1, \quad (17)$$

where τ, a_1, \dots, a_m are decision variables.

The nonlinear problem (13)–(17) is non-convex, though it can be solved by a nonlinear solver. In this paper, we will apply LOQO (Vanderbei, 1999) to this problem and present the numerical results in the next section. Since directly solving the nonlinear problem is not an easy task, we develop a linear model to approximate the nonlinear problem.

Linear Model: Note that expression (9) can be approximated by the following:

$$\min_{a \in \mathbb{R}^m} \frac{\max_{1 \leq n \leq d} \max_{-N \leq i \leq N} \left| \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) \right|}{\min_{-N \leq i \leq N} \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right)}. \quad (18)$$

Again, since the objective is independent of scaling in a , by the constraint (10), we can choose the scale such that

$$\min_{-N \leq i \leq N} \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right) = 1. \quad (19)$$

Note that in this case we use scaling (19) instead of (11). By using constraint (19), expression (18) can be reduced to

$$\min_{a \in \mathbb{R}^m} \max_{1 \leq n \leq d} \max_{-N \leq i \leq N} \left| \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) \right|. \quad (20)$$

In order to simplify (20), we set

$$\tau = \max_{1 \leq n \leq d} \max_{-N \leq i \leq N} \left| \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) \right|.$$

Then (20) is equivalent to

$$\min \tau$$

subject to

$$\tau = \max_{1 \leq n \leq k} \max_{-N \leq i \leq N} \left| \sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) \right|.$$

Hence, the complete discretized linear model is given by the following:

$$\min \tau \quad (21)$$

subject to

$$\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) - \tau \leq 0, \quad -N \leq i \leq N, 1 \leq n \leq d, \quad (22)$$

$$\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} + n \right) + \tau \geq 0, \quad -N \leq i \leq N, 1 \leq n \leq d, \quad (23)$$

$$\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right) \geq 1, \quad 0 \leq i \leq N, \quad (24)$$

where τ, a_1, \dots, a_m are decision variables.

This linear model can be easily solved by an interior point method, and the numerical results produced by our program are given in the next section. However, the linear model is just an approximation to the nonlinear problem, not the real problem we want to solve.

Iterative Scheme: Now we design an iterative algorithm to solve the original nonlinear, non-convex model by successively solving a sequence of linear problems. More precisely, for each integer k , we will look for a vector $[a_{k,1}, a_{k,2}, \dots, a_{k,m}]$, which is the decision vector we want to determine. When $k = 0$, we define $a_{k,1} = a_{k,2} = \dots = a_{k,m} = \frac{1}{m}$. When $k > 0$, the decision variables $a_{k,j}$ ($j = 1, \dots, m$) are given by the following optimization problem (P_k):

$$\min_{a \in \mathbb{R}^m} \max_{1 \leq n \leq d} \max_{-N \leq i \leq N} \frac{|\sum_{j=1}^m a_{k,j} \hat{f}_j \left(\frac{iw}{2N} + n \right)|}{\sum_{j=1}^m a_{k-1,j} \hat{f}_j \left(\frac{iw}{2N} \right)} \quad (25)$$

subject to the constraints

$$\sum_{j=1}^m a_{k,j} \hat{f}_j \left(\frac{iw}{2N} \right) > 0, \quad 0 \leq i \leq N,$$

and

$$\sum_{j=1}^m a_{k,j} = 1.$$

We notice that the objective function (25) is a linearized version of (9), since $a_{k-1,j}$ are known when we solve $a_{k,j}$ ($j = 1, \dots, m$). Using a transformation analogous to the one we applied to derive the linear model, problem (P_k) can be equivalently formulated as the following iterative optimization model:

$$\min \tau_k \quad (26)$$

subject to the constraints

$$\sum_{j=1}^m a_{k,j} \hat{f}_j \left(\frac{iw}{2N} + n \right) \leq \tau_k \sum_{j=1}^m a_{k-1,j} \hat{f}_j \left(\frac{iw}{2N} \right), \quad -N \leq i \leq N, \quad 1 \leq n \leq d, \quad (27)$$

$$-\sum_{j=1}^m a_{k,j} \hat{f}_j \left(\frac{iw}{2N} + n \right) \leq \tau_k \sum_{j=1}^m a_{k-1,j} \hat{f}_j \left(\frac{iw}{2N} \right), \quad -N \leq i \leq N, \quad 1 \leq n \leq d, \quad (28)$$

$$\sum_{j=1}^m a_{k,j} \hat{f}_j \left(\frac{iw}{2N} \right) > 0, \quad 0 \leq i \leq N, \quad (29)$$

Input:
 a parameter $\gamma > 0$ for the relative accuracy of objective values;
 an integer K to limit the number of iterations;
 an initial vector $[a_{0,1}, a_{0,2}, \dots, a_{0,m}]$ with $\sum_{j=1}^m a_{0,j} = 1$;
 a number $d_0 > \gamma$ for the initial value of variable d .

begin
 $k := 1$;
 previous_obj := 0;
 current_obj := 0;
 $d := d_0$.
while $k \leq K$ and $d > \gamma$ **do**
begin
 solve problem (P_k) for $\tau_k, a_{k,1}, \dots, a_{k,m}$;
 previous_obj := current_obj;
 current_obj := value of expression (9) after substitution $a_{k,j}$ for
 a_j ($j = 1, \dots, m$);
 $d := | \text{current_obj} - \text{previous_obj} | / \text{current_obj}$;
 $k := k + 1$.
end
end

Figure 5. Psuedocode of the iterative non-linear method. Here, k stands for the number of iterations; current_obj for the value of expression (9) after a_j are replaced by the solution $a_{k,j}$ of current subproblem ($j = 1, \dots, m$); previous_obj for the previous value of current_obj; and d for the relative difference between current_obj and previous_obj.

$$\sum_{j=1}^m a_{k,j} = 1, \quad (30)$$

where $\tau_k, a_{k,1}, \dots, a_{k,m}$ are decision variables.

To solve this model, we first initialize $a_{0,j} = \frac{1}{m}$ ($j = 1, \dots, m$), and then solve problem (26) with (27)–(30) to obtain $a_{1,j}$ ($j = 1, \dots, m$), and so on. We repeat this procedure until the vector $[a_{k,1}, a_{k,2}, \dots, a_{k,m}]$ converges with respect to k . In the next section we will see that, in practice, this series does converge rapidly. See Figure 5 for a pseudocode description.

For a fixed integer k , we solve the problem (P_k) , *i.e.*, (26)–(30), using an interior point method. Since the interior point method itself is also an iterative scheme and we do not require the optimal solution to each subproblem, we restrict each subproblem to a limited number of iterations. The number of iterations for each subproblem and the number of subproblems required for convergence can be determined empirically. Since the solution to the linear problem is comparable to the nonlinear problem, we don't expect many more total iterations to be necessary, than would be required for the linear problem.

Table I. Testing results for optimal values of parameters with 71 discrete points.

iterations per IPM	subproblems	total iterations	time (s)
1	16	16	1.03
2	8	16	0.96
3	6	17	1.00
4	5	18	1.04
5	5	20	1.13
6	4	21	1.20
7	4	23	1.34

4. Computational Results

In this section, we discuss the numerical results produced by the solver LOQO (Vanderbei, 1999) for the nonlinear model, and our implementation of an Interior Point Method (IPM) that is used to solve the linear model and the nonlinear model. Our IPM is based on the algorithm described in (Andersen et al., 1996). We will also compare the cost of solving the nonlinear model (13)–(17), linear model (21)–(24), and iterative model (26)–(30).

To prevent possible numerical problems and loss of precision during image reconstruction the strict inequalities (16) and (29) in the nonlinear model and in the iterative scheme are replaced by

$$\sum_{j=1}^m a_j \hat{f}_j \left(\frac{iw}{2N} \right) \geq \epsilon, \quad 0 \leq i \leq N, \quad (31)$$

and

$$\sum_{j=1}^m a_{k,j} \hat{f}_j \left(\frac{iw}{2N} \right) \geq \epsilon, \quad 0 \leq i \leq N, \quad (32)$$

respectively, where ϵ is a small positive number. In practice, this constraint can be used to fix the scale of the decision variables, and the constraint (30) can be dropped. In this case $\epsilon = 1$ is a good choice.

The initial computational experiments were performed on an IBM RS/6000 44P model 270 workstation. Once the iterative algorithm was finalized, some of the linear algebra and memory allocation was optimized and ported to a 1GHz ppc7450, a processor suitable for embedded signal processing. The times reported for the iterative algorithm in Table II are for the PPC, all other times are for the RS/6000. Hereafter, we only consider the case where the model parameters are $m = 8$, $l = 4$, $d = 3$ and $w = \frac{1}{2}$.

To solve the iterative model as fast as possible, we determine the optimal number of iterations per subproblem which minimizes the total number of iterations to solve the

Table II. Numerical results obtained from LOQO, Iterative Method and Linear Method. For the discrete points indicated by * LOQO did not converge in 500 iterations with the default parameter setting.

Discrete Points	LOQO			Iterative Method			Linear Method		
	iter	time(s)	obj $\times 10^4$	iter	time(s)	obj $\times 10^4$	iter	lobj $\times 10^4$	obj $\times 10^4$
51	210	1.41	1.7329	16	0.01	1.7329	14	2.2649	2.2648
61	178	1.17	1.7368	16	0.01	1.7368	14	2.2617	2.2616
71	131	1.04	1.7330	16	0.02	1.7330	15	2.2648	2.2647
81	141	1.11	1.7371	16	0.02	1.7371	15	2.2691	2.2691
91	64	0.49	1.7368	16	0.02	1.7368	15	2.2680	2.2678
101	86	0.76	1.7354	17	0.02	1.7354	15	2.2676	2.2675
111	152	1.51	1.7379	17	0.03	1.7379	16	2.2680	2.2680
121	171	2.02	1.7375	17	0.02	1.7375	15	2.2694	2.2692
131	435	6.43	1.7368	17	0.03	1.7368	16	2.2690	2.2690
141	320	5.21	1.7380	17	0.02	1.7380	16	2.2685	2.2685
151*	500	10.19	83.69	17	0.03	1.7379	16	2.2690	2.2689
161	246	5.13	1.7375	17	0.03	1.7375	16	2.2697	2.2696
171	171	3.32	1.7381	17	0.05	1.7381	16	2.2692	2.2692
181*	500	16.58	460.63	17	0.03	1.7381	16	2.2692	2.2691
191*	500	13.32	2.0348	17	0.04	1.7379	16	2.2691	2.2691
201	482	13.8	1.7382	17	0.04	1.7382	16	2.2698	2.2698
211	96	1.76	1.7381	17	0.06	1.7381	16	2.2696	2.2696
221	102	1.99	1.7381	17	0.05	1.7381	16	2.2693	2.2692
231	81	1.68	1.7383	17	0.06	1.7383	16	2.2694	2.2693
241	500	17.74	1.7381	17	0.07	1.7380	16	2.2698	2.2698
251	114	2.66	1.7383	17	0.07	1.7383	16	2.2698	2.2697

iterative model. To this end, we discretize the interval $[-\frac{w}{2}, \frac{w}{2}] = [-\frac{1}{4}, \frac{1}{4}]$ with 71 equally spaced points, and solve the iterative model with different numbers of iterations per subproblem. The results are presented in Table I. The first column of Table I, labelled “iterations per IPM”, represents the number of iterations used to solve each subproblem approximately by the interior point method; the second column, labelled “subproblems”, represents the minimum number of subproblems that have to be solved in order to obtain the optimal solution of the nonlinear model; and the third column, labelled “total iterations”, represents the the total number of iterations when the optimal solution is found. From Table I, we see that the minimum total number of iterations to solve the iterative model is 16 when the iteration number per subproblem is set to 2, which we will use in the comparisons to follow. In this case, the execution time is 0.96 second, which is the fastest among all cases.

Next we compared the numerical results obtained by LOQO with our software. In Table II, for each individual discretization, we list all the information about the total iteration number, CPU time and the optimal objective value produced by LOQO for solving the nonlinear model (13)–(17), by the iterative method for solving the iterative model (26)–(30), and by the linear method for solving the linear model (21)–(24). Let’s

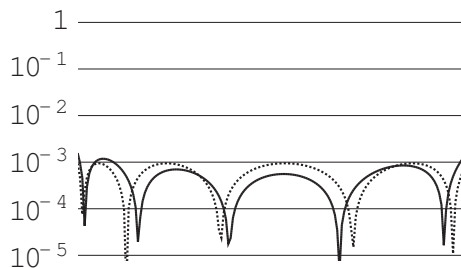


Figure 6. The first stop-band for the optimal kernels (for the approximate linear problem—solid; for the full non-linear problem—dotted). Note that the solution to the linear problem has lower aliasing in the middle of the band, and lower aliased L^2 energy on average, but has higher absolute aliasing, which occurs near the boundary. Exceptionally, we show the results for image window:transform length 2 : 3 instead of 1 : 2.

explain some notations in Table II: “iter” stands for the total number of iterations, which since we are using at most two IPM iterations per subproblem in the iterative case, is roughly twice the number of outer iterations; “time” for the CPU time which is comparable across CPU architectures; “obj” for the value of the original nonlinear objective (9) after the decision variables a_1, \dots, a_m are determined; and “lobj” for the objective value of the linear model (21)–(24) after the decision variables a_1, \dots, a_m are solved by the linear method. All results in Table II were obtained by using both the default parameter settings of LOQO and our software.

From Table II, we see that the iterative method successfully solved the nonlinear problem in all cases with different discretizations. In most cases, LOQO also successfully solved the nonlinear problem, but, in some cases, LOQO failed to produce the optimal solution in 500 iterations with the default parameters. Table II also shows that the iteration number of the iterative method (at most 17) is much less than the iteration number of LOQO (at least 64), and the CPU time required by the dedicated solver is significantly less than that required by LOQO.

Compared to the iterative method, the linear method is a little bit faster, as evidenced by the reduced number of iterations, but it produces only approximate optimal solutions, not the exact optimal solution, since the linear model is an approximation of the original nonlinear problem. Actually, from Table II, we see that, by using the decision variables a_1, \dots, a_m obtained by the linear method, the objective value of the linear model and the value of expression (9) are both significantly worse than the objective values obtained by LOQO and the iterative method. As shown in Figure 6, the solution to the linear problem is better in the centre of the aliasing band, and probably better in the L^2 sense, but worse at the boundaries of the aliasing band, as we would expect.

Using any of the three methods presented in this paper, we can obtain a kernel function by solving an optimization problem. Now let us compare our kernel functions with the commonly used kernel in the literature. In Figure 7, we see that our kernel performs significantly better than a piecewise-linear approximation of a Kaiser-Bessel kernel. Figure 8 also shows that the Kaiser-Bessel kernel has a transfer function which falls off more rapidly

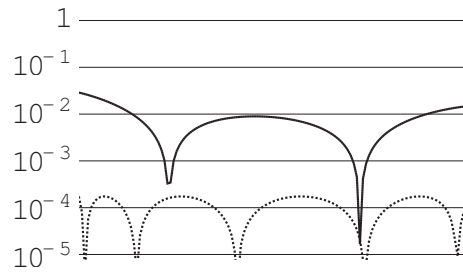


Figure 7. The first stop-band for the KB (solid $a_1 = -0.01097201305$, $a_2 = -0.02819949502$, $a_3 = -0.01753561254$, $a_4 = 0.04431120359$, $a_5 = 0.1459885419$, $a_6 = 0.2422379845$, $a_7 = 0.2301496146$, $a_8 = 0.3940197759$) and optimized kernels (dotted, $a_1 = -0.01642718191$, $a_2 = -0.03149300674$, $a_3 = 0.01406508711$, $a_4 = 0.08747566023$, $a_5 = 0.2503776262$, $a_6 = 0.2886939451$, $a_7 = 0.2146258540$, $a_8 = 0.1926820160$), with image-window correction applied.

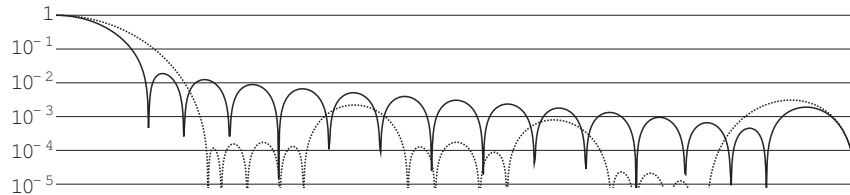


Figure 8. Transfer functions for the linearly-interpolated Kaiser-Bessel kernel (solid, as above), and the optimized kernel (dotted, as above).

than ours, but which allows much more aliasing from the aliasing bands. Including the effect of correcting the imaging window for roll-off, we see that the Kaiser-Bessel kernel has worst-case (L^∞ norm) two orders of magnitude less aliasing attenuation.

4.1. SIMULATION OF AN ALIASING NOISE-SOURCE.

To illustrate why it is advisable to fully optimize these resampling kernels, we consider an example from MRI.

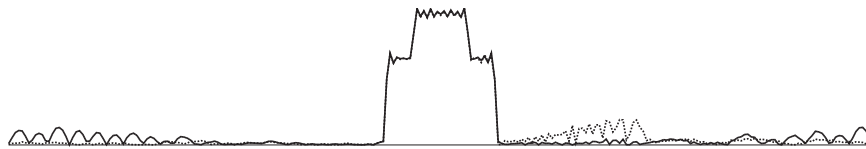


Figure 9. The effect of a pure-frequency noise source outside the imaging band on a sinusoidally-sampled acquisition, showing a two-level numerical phantom in the centre of the imaging interval. Resampling with the optimized kernel (solid line) reduces the level of aliased noise in the imaging interval (the middle half of the interval shown) significantly compared to the result of the Kaiser-Bessel kernel.

Of all the sources of aliasing mentioned, it could be argued that beyond attenuations of one in 300, white noise and object from outside the reconstructed field of view will not be visible in the image. Systematic noise (*e.g.*, coherent noise from digital clocks), however, may be present at any level, and will show up in different ways in the image. The most common type of non-raster imaging is echo planar imaging (EPI, see (Haacke et al., 1999)) in which the sampling position in the first dimension (and consequently the sampling density) are sinusoids. A sinusoidally-resampled single frequency will appear as a series of equally-spaced dots in the image. In the case of uniform k -space sampling, the clock frequency may be outside the imaging-band, and innocuous. but in the EPI case the series of dots will extend into the image. In Figure 9 we show that the optimized kernel significantly out-performs the Kaiser-Bessel kernel in the imaging band by allowing more aliasing outside the imaging band. This illustrates an important point: that the attenuation is of the signals as they appear after resampling. In this case the noise is a single frequency, but resampling converts it to a sum of frequencies, and it is the frequencies after resampling which are attenuated.

5. Conclusion

This paper presents three models to solve an optimization problem in the design of kernel functions for nonuniform Fourier transforms. These models are the nonlinear model (13)–(17), linear model (21)–(24), and iterative model (26)–(30). For the first model, we used the nonlinear solver LOQO to solve the nonlinear model. We then developed software to solve both the iterative and linear models. The computational results indicate that our software is embeddable in high end embedded signal processing applications, meeting requirements for speed, robustness and small software size.

References

- A. Aldroubi and K. Gröchenig, “Nonuniform Sampling and Reconstruction in Shift-Invariant Spaces,” *SIAM Review* vol. 4, pp. 585–620, 2001.
- E.D. Andersen, J. Gondzio, C. Mészáros, X. Xu, “Implementation of Interior-Point Methods for Large Scale Linear Programs,” in *Interior Point Methods of Mathematical Programming*, T. Terlaky ed., Kluwer, 1996.
- M.S. Bazzaraa, H.D. Sherali, C.M. Shetty, *Nonlinear Programming*, Wiley-Interscience, 1993.
- P. Boernert, *Morning course in image reconstruction*, ISMRM 2001, Glasgow.
- A.S. Fahmy and B.S. Tawfik and Y.M. Kadah, “Image reconstruction using shift-variant resampling kernel for magnetic resonance imaging,” *Medical Imaging 2002: Image Processing* M. Sonka and J.M. Fitzpatrick, eds., SPIE, pp.825–833, 2002.
- J. A. Fessler and B. P. Sutton, “Nonuniform fast Fourier transforms using min-max interpolation,” *IEEE Tr. Sig. Proc.* vol. 51, pp. 560-574, 2003.
- E.M. Haacke, R.W. Brown, M.R. Thompson, R. Venkatesan, *Magnetic Resonance Imaging: Physical Principles and Sequence Design*, John Wiley & Sons, 1999.
- R.W. Hamming, *Digital Filters*, Prentice-Hall, New Jersey, 1983.
- F.J. Harris, “On the use of windows for harmonic analysis with the discrete Fourier transform,” *Proc. IEEE* vol. 4, pp. 51–83, 1978.

- J. Jackson, C.H. Meyer, D.G. Nishimura, "Selection of a Convolution Function for Fourier Inversion Using Gridding," *IEEE Trans. Med. Imaging* vol. 10, pp. 473–478, 1991.
- Z.-P. Liang, P. C. Lauterbur, *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*, Wiley-IEEE, 2000.
- F. Marvasti, ed., *Nonuniform Sampling: Theory and Practice (Information Technology: Transmission, Processing, and Storage)*, Plenum, 2001.
- J. O'Sullivan, "A fast sinc function gridding algorithm for Fourier inversion in computer tomography," *IEEE Trans. Med. Imaging* vol. 4, 200–207, 1985.
- J.G. Pipe and P. Menon, Sampling Density Compensation in MRI: Rationale and an Iterative Numerical Solution, *Magn. Reson. Med.*, vol. 41, 179-186, 1999.
- V. Rasche, et al, Resampling of data between arbitrary grids using convolution interpolation, *IEEE Trans. Medical Imaging*, vol. 18, 385–392, 1999.
- D. Rosenfeld, An optimal and efficient new gridding algorithm using singular value decomposition, *Magn. Reson. Med.*, vol. 40, 14–23, 1998.
- H. Sedarat and D.G. Nishimura, On the optimality of the gridding reconstruction algorithm, *IEEE Trans. Medical Imaging*, vol. 19, 306–317, 2000.
- R.J. Vanderbei. LOQO user's manual—version 3.10. Interior point methods. *Optimization Methods and Software*, 11/12:485–514, 1999.
- F.T.A.W Wajer et al, Interpolation from Arbitrary to Cartesian Sample Positions: Gridding, *Proc. ProRISC/IEEE Workshop*, 2000.
- F.T.A.W Wajer et al, Simple formula for the accuracy of gridding, *Proc. Intern. Soc. Mag. Resonance Med.*, 2001.
- F.T.A.W Wajer et al, Interpolation from Arbitrary to Cartesian Sample Positions: Gridding, *Proc. ProRISC/IEEE Workshop*, 2001.