Iterative Image Reconstruction Methods for Non-Cartesian MRI

Jeff Fessler
EECS Department

Doug Noll
BME Department

The University of Michigan

ISMRM Workshop on Non-Cartesian MRI
Feb. 27, 2007

Acknowledgements: Brad Sutton
Outline

- MR image reconstruction problem description
- Overview of image reconstruction methods
- Model-based image reconstruction
- Regularization
- Iterations and computation (NUFFT etc.)
- Myths about iterative reconstruction
- Example for partial non-Cartesian k-space

Image reconstruction toolbox:
http://www.eecs.umich.edu/~fessler
Why Iterative Image Reconstruction?

- Statistical modeling may reduce noise
- Incorporate prior information, e.g.:
  - support constraints
  - (piecewise) smoothness
  - phase constraints
- No density compensation needed
- “Non-Fourier” physical effects such as field inhomogeneity
- Incorporation of coil sensitivity maps
- Improved results for under-sampled trajectories (?)
- …

(“Avoiding k-space interpolation” is not a compelling reason!)
Primary drawbacks of Iterative Methods

- Choosing regularization parameter(s)
- Algorithm speed
Introduction
Non-Cartesian MR Image Reconstruction

“k-space” data
\[ y = (y_1, \ldots, y_M) \]

image
\[ f(\vec{r}) \]

k-space trajectory:
\[ \vec{k}(t) = (k_x(t), k_y(t)) \]

spatial coordinates:
\[ \vec{r} \in \mathbb{R}^2 \]
Ignoring *lots* of things, the standard measurement model is:

\[ y_i = s(t_i) + \text{noise}_i, \quad i = 1, \ldots, M \]

\[ s(t) = \int f(\vec{r}) e^{-i2\pi \vec{\kappa}(t) \cdot \vec{r}} \, d\vec{r} = F(\vec{\kappa}(t)). \]

\( \vec{r} \): spatial coordinates  
\( \vec{\kappa}(t) \): k-space trajectory  
\( f(\vec{r}) \): object’s unknown transverse magnetization  
\( F(\vec{\kappa}) \): Fourier transform of \( f(\vec{r}) \)

Goal of image reconstruction: find \( f(\vec{r}) \) from measurements \( \{y_i\}_{i=1}^M \).

The unknown object \( f(\vec{r}) \) is a continuous-space function, but the recorded measurements \( y = (y_1, \ldots, y_M) \) are finite.

Inherently under-determined (ill posed) problem \( \implies \) no canonical solution.

*All MR scans provide only “partial” k-space data.*
Image Reconstruction Strategies

- **Continuous-continuous formulation**
  Pretend that a continuum of measurements are available:
  \[ F(\vec{\kappa}) = \int f(\vec{r}) e^{-i2\pi \vec{\kappa} \cdot \vec{r}} \, d\vec{r}. \]
  The “solution” is an inverse Fourier transform:
  \[ f(\vec{r}) = \int F(\vec{\kappa}) e^{i2\pi \vec{\kappa} \cdot \vec{r}} \, d\vec{\kappa}. \]

Now discretize the integral solution:
\[
\hat{f}(\vec{r}) = \sum_{i=1}^{M} F(\vec{\kappa}_i) e^{i2\pi \vec{\kappa}_i \cdot \vec{r}} w_i \approx \sum_{i=1}^{M} y_i w_i e^{i2\pi \vec{\kappa}_i \cdot \vec{r}},
\]
where \( w_i \) values are “sampling density compensation factors.” Numerous methods for choosing \( w_i \) value in the literature.
For Cartesian sampling, using \( w_i = 1/N \) suffices, and the summation is an inverse FFT.
For non-Cartesian sampling, replace summation with gridding.
• **Continuous-discrete formulation**
  Use many-to-one linear model:
  \[ y = Af + \epsilon, \text{ where } A : L_2(\mathbb{R}^2) \rightarrow \mathbb{C}^M. \]
  Minimum norm solution (cf. “natural pixels”):
  \[
  \min_{\hat{f}} \|\hat{f}\| \text{ subject to } y = A\hat{f}
  \]
  \[
  \hat{f} = A^* (AA^*)^{-1} y = \sum_{i=1}^{M} c_i e^{-i2\pi \kappa_i \cdot \vec{r}}, \text{ where } AA^* c = y.
  \]

• **Discrete-discrete formulation**
  Assume parametric model for object:
  \[
  f(\vec{r}) = \sum_{j=1}^{N} f_j p_j(\vec{r}).
  \]
  Estimate parameter vector \( \mathbf{f} = (f_1, \ldots, f_N) \) from data vector \( y \).
Substitute series expansion of unknown object:

\[ f(\vec{r}) = \sum_{j=1}^{N} f_j p(\vec{r} - \vec{r}_j) \quad \leftarrow \text{usually 2D rect functions} \]

into signal model \( y_i = s(t_i) + \varepsilon_i \), where

\[ E[y_i] = s(t_i) = \int f(\vec{r}) e^{-i2\pi \vec{k}_i \cdot \vec{r}} \, d\vec{r}, \]

yields:

\[
E[y_i] = \int \left[ \sum_{j=1}^{N} f_j p(\vec{r} - \vec{r}_j) \right] e^{-i2\pi \vec{k}_i \cdot \vec{r}} \, d\vec{r} = \sum_{j=1}^{N} \int p(\vec{r} - \vec{r}_j) e^{-i2\pi \vec{k}_i \cdot \vec{r}} \, d\vec{r} \quad f_j
\]

\[
= \sum_{j=1}^{N} a_{ij} f_j, \quad a_{ij} = P(\vec{k}_i) \quad e^{-i2\pi \vec{k}_i \cdot \vec{r}_j}, \quad p(\vec{r}) \overset{\text{FT}}{\longleftrightarrow} P(\vec{k}).
\]

Discrete-discrete measurement model with system matrix \( A = \{a_{ij}\} \):

\[
y = Af + \varepsilon.
\]

Goal: estimate coefficients (pixel values) \( f = (f_1, \ldots, f_N) \) from \( y \).
Small Pixel Size Need Not Matter

![Images of different pixel sizes](image-url)
Profiles

\[ |f(x,0)| \]

- **true**
- \( N=32 \)
- \( N=64 \)
- \( N=128 \)
- \( N=256 \)
- \( N=512 \)

**horizontal position [mm]**
Regularized Least-Squares Estimation

Estimate object by minimizing a cost function:

\[
\hat{f} = \arg \min_{f \in \mathbb{C}^N} \Psi(f), \quad \Psi(f) = \|y - Af\|^2 + \alpha R(f)
\]

- **data fit term** \(\|y - Af\|^2\) corresponds to negative log-likelihood of Gaussian distribution
- **regularizing term** \(R(f)\) controls noise by penalizing roughness,
  
  \[e.g.: \quad R(f) \approx \int \|\nabla f\|^2 \, d\vec{r}\]

- **regularization parameter** \(\alpha > 0\) controls tradeoff between spatial resolution and noise
- Equivalent to Bayesian MAP estimation with prior \(\propto e^{-\alpha R(f)}\)

**Issues:**
- choosing \(R(f)\)
- choosing \(\alpha\)
- computing minimizer rapidly.
Quadratic regularization

1D example: squared differences between neighboring pixel values:

\[ R(f) = \sum_{j=2}^{N} \frac{1}{2} |f_j - f_{j-1}|^2. \]

In matrix-vector notation, \( R(f) = \frac{1}{2} \|Cf\|^2 \) where

\[ C = \begin{bmatrix}
-1 & 1 & 0 & 0 & \ldots & 0 \\
0 & -1 & 1 & 0 & \ldots & 0 \\
0 & \ldots & 0 & 0 & \ldots & -1 \\
0 & \ldots & 0 & 0 & -1 & 1
\end{bmatrix}, \text{ so } Cf = \begin{bmatrix}
f_2 - f_1 \\
\vdots \\
f_N - f_{N-1}
\end{bmatrix}. \]

For 2D and higher-order differences, modify differencing matrix \( C \).

Leads to closed-form solution:

\[ \hat{f} = \arg \min_f \|y - Af\|^2 + \alpha \|Cf\|^2 \]

\[ = [A'A + \alpha C'C]^{-1} A'y. \]

(a formula of limited practical use for computing \( \hat{f} \))
Choosing the Regularization Parameter

Spatial resolution analysis (Fessler & Rogers, IEEE T-IP, 1996):

\[
\hat{f} = \left[ A'^{T}A + \alpha C'^{T}C \right]^{-1} A'^{T}y
\]

\[
\mathbb{E}\left[ \hat{f} \right] = \left[ A'^{T}A + \alpha C'^{T}C \right]^{-1} A'^{T}\mathbb{E}[y]
\]

\[
\mathbb{E}\left[ \hat{f} \right] = \left[ A'^{T}A + \alpha C'^{T}C \right]^{-1} A'^{T}A f_{\text{blur}}
\]

\(A'A\) and \(C'C\) are Toeplitz \(\implies\) blur is approximately shift-invariant.

Frequency response of blur:

\[
L(\omega) = \frac{H(\omega)}{H(\omega) + \alpha R(\omega)}
\]

- \(H(\omega_k) = \text{FFT}(A'A e_j)\) (lowpass)
- \(R(\omega_k) = \text{FFT}(C'C e_j)\) (highpass)

Adjust \(\alpha\) to achieve desired spatial resolution.
Spatial Resolution Example

Radial k-space trajectory, FWHM of PSF is 1.2 pixels
Spatial Resolution Example: Profiles

\[ H(\omega) \]

\[ R(\omega) \]

\[ L(\omega) \]
Resolution/noise tradeoffs

Noise analysis:

$$\text{Cov}\{\hat{f}\} = \left[A'A + \alpha C'C\right]^{-1} A' \text{Cov}\{y\} A \left[A'A + \alpha C'C\right]^{-1}$$

Using circulant approximations to $A'A$ and $C'C$ yields:

$$\text{Var}\{\hat{f}_j\} \approx \sigma^2 \sum_k \frac{H(\omega_k)}{(H(\omega_k) + \alpha R(\omega_k))^2}$$

- $H(\omega_k) = \text{FFT}(A'A e_j)$ (lowpass)
- $R(\omega_k) = \text{FFT}(C'C e_j)$ (highpass)

$\implies$ Predicting reconstructed image noise requires just 2 FFTs. (cf. gridding approach?)

Adjust $\alpha$ to achieve desired spatial resolution / noise tradeoff.
In short: one can choose $\alpha$ rapidly and predictably for quadratic regularization.
Iterative Minimization by Conjugate Gradients

Choose initial guess \( f^{(0)} \) (e.g., fast conjugate phase / gridding).

Iteration (unregularized):

\[
\begin{align*}
g^{(n)} &= \nabla \Psi(f^{(n)}) = A'(Af^{(n)} - y) \quad \text{gradient} \\
p^{(n)} &= Pg^{(n)} \quad \text{precondition} \\
\gamma_n &= \begin{cases} 
0, & n = 0 \\
\frac{\langle g^{(n)}, p^{(n)} \rangle}{\langle g^{(n-1)}, p^{(n-1)} \rangle}, & n > 0
\end{cases} \\
d^{(n)} &= -p^{(n)} + \gamma_n d^{(n-1)} \quad \text{search direction} \\
\alpha_n &= \frac{\langle d^{(n)}, -g^{(n)} \rangle}{\langle Af^{(n)}, Af^{(n)} \rangle} \quad \text{step size} \\
f^{(n+1)} &= f^{(n)} + \alpha_n d^{(n)} \quad \text{update}
\end{align*}
\]

Bottlenecks: computing \( Af^{(n)} \) and \( A' r \).

- \( A \) is too large to store explicitly (not sparse)
- Even if \( A \) were stored, directly computing \( Af \) is \( O(MN) \) per iteration, whereas FFT is only \( O(M \log M) \).
Computing $A f$ Rapidly

$$[A f]_i = \sum_{j=1}^{N} a_{ij} f_j = P(\vec{k}_i) \sum_{j=1}^{N} e^{-i 2\pi \vec{k}_i \cdot \vec{r}_j} f_j, \quad i = 1, \ldots, M$$

- Pixel locations $\{\vec{r}_j\}$ are uniformly spaced
- k-space locations $\{\vec{k}_i\}$ are unequally spaced

$\Rightarrow$ needs nonuniform fast Fourier transform (NUFFT)
NUFFT (Type 2)

- Compute over-sampled FFT of equally-spaced signal samples
- Interpolate onto desired unequally-spaced frequency locations
- Dutt & Rokhlin, SIAM JSC, 1993, Gaussian bell interpolator

NUFFT toolbox: http://www.eecs.umich.edu/~fessler/code
Worst-Case NUFFT Interpolation Error

Maximum error for K/N=2

- Min–Max (uniform)
- Gaussian (best)
- Min–Max (best L=2)
- Kaiser–Bessel (best)
- Min–Max (L=13, $\beta=1$ fit)
Further Acceleration using Toeplitz Matrices

Cost-function gradient:

\[ g^{(n)} = A'(Af^{(n)} - y) = Tf^{(n)} - b, \]

where

\[ T \triangleq A'A, \quad b \triangleq A'y. \]

In the absence of field inhomogeneity, the Gram matrix \( T \) is Toeplitz:

\[ [(A'A)]_{jk} = \sum_{i=1}^{M} |P(\mathbf{k}_i)|^2 e^{-i2\pi \mathbf{k}_i \cdot (\mathbf{r}_j - \mathbf{r}_k)}. \]

Computing \( Tf^{(n)} \) requires an ordinary \((2 \times \text{over-sampled})\) FFT. \((\text{Chan \\& Ng, SIAM Review, 1996})\)

In 2D: block Toeplitz with Toeplitz blocks (BTTB).

Precomputing the first column of \( T \) and \( b \) requires a couple NUFFTs. \((\text{Wajer, ISMRM 2001, Eggers ISMRM 2002, Liu ISMRM 2005})\)

This formulation seems ideal for “hardware” FFT systems.
Toeplitz Acceleration

Example: $256^2$ image. radial trajectory, $2\times$ angular under-sampling.

(Iterative provides reduced aliasing energy.)
Toeplitz Acceleration

<table>
<thead>
<tr>
<th>Method</th>
<th>$\mathbf{A}'\mathbf{Dy}$</th>
<th>$b = \mathbf{A}'\mathbf{y}$</th>
<th>$T$</th>
<th>20 iter</th>
<th>Total Time</th>
<th>NRMS (50dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conj. Phase</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
<td>0.3</td>
<td>7.8%</td>
</tr>
<tr>
<td>CG-NUFFT</td>
<td></td>
<td>12.5</td>
<td>12.5</td>
<td></td>
<td></td>
<td>4.1%</td>
</tr>
<tr>
<td>CG-Toeplitz</td>
<td>0.3 0.8</td>
<td>3.5</td>
<td>4.6</td>
<td></td>
<td></td>
<td>4.1%</td>
</tr>
</tbody>
</table>

- Toeplitz aproach reduces CPU time by more than $2\times$ on conventional workstation (Xeon 3.4GHz)
- Eliminates k-space interpolations $\implies$ ideal for FFT hardware
- No SNR compromise
- CG reduces NRMS error relative to CP, but $15\times$ slower...
  (More dramatic improvements seen in fMRI when correcting for field inhomogeneity.)
Myths

- Choosing $\alpha$ is difficult
- Sample density weighting is desirable
Sampling density weighted LS

Some researchers recommend using a weighted LS cost function:

$$\Psi(f) = \|y - Af\|^2_W$$

where the weighting matrix $W$ is related to the k-space sample density compensation factors (DCF).

Purported benefits:
- Faster convergence
- Better conditioning

But, Gauss-Markov theorem from statistical estimation theory states that lowest estimator variance is realized by choosing $W = \sigma^2 I$, the inverse of the data noise covariance.
Resolution/Noise Tradeoff: Example with Weighting

$$\text{cond}(A'WA) \approx 1.04673 \times 10^6$$

$$\text{cond}(A'A) \approx 1.11761 \times 10^8$$
Don’t just take it from me...

**Figure 5.** Reconstructed images for simulated spiral sampling of the noisy $k$-space of the SL phantom (60% noise) using conventional gridding (a), F-CG without (b) and with spiral DCF (c), and DING (d).

Fig. 5 of Gabr *et al.*, MRM, Dec. 2006
Figure 8. The RMS error of F-CG and DING vs. the number of CG iterations for the simulated noiseless SL the random trajectory. Errors stabilize after approximately 15 iterations.

Fig. 8 of Gabr et al., MRM, Dec. 2006. Zero initialization!
Acceleration via Initialization

NRMS Error vs. Iteration for different initializations:
- Unweighted, 0 initialized
- Weighted, 0 initialized
- Unweighted, CP initialized
Summary

- Iterative reconstruction: much potential in MRI
- Quadratic regularization parameter selection is tractable
- Computation: reduced by tools like NUFFT / Toeplitz
- But optimization algorithm design remains important (cf. Shepp and Vardi, 1982, PET)

Some current challenges

- Sensitivity pattern mapping for SENSE
- Through-voxel field inhomogeneity gradients
- Motion / dynamics / partial k-space data
- Establishing diagnostic efficacy with clinical data...

Image reconstruction toolbox:
http://www.eecs.umich.edu/~fessler