Gridding Reconstruction

Assignment 1 due Thursday
Office hours on web site
Read 13.2 in Berenshein
More complete gridding algorithm
Density compensation
CONVOLVE DATA WITH KERNEL
RESAMPLE ON GRID

IN K-SPACE

\[
\hat{M}(k_x, k_y) = \left[ (m(k_x, k_y) \ast s(k_x, k_y)) \ast C(k_x, k_y) \right] \ast \left[ \frac{k_x}{\Delta x}, \frac{k_y}{\Delta y} \right]
\]

IN IMAGE SPACE

\[
\hat{m}(x, y) = \left[ (m(x, y) \ast s(x, y)) \ast C(x, y) \right] \ast \left[ \frac{x}{\text{FOV}_x}, \frac{y}{\text{FOV}_y} \right]
\]

IN 1D:

\[
\text{FROM } \hat{m} \left( \frac{k}{\text{FOV}_x}, \frac{y}{\text{FOV}_y} \right)
\]
A PODORIZATION AND \( (k\lambda/\sigma) \)

**Ideal APODIZATION**

\[ C(x) = \text{rect} \left( \frac{x}{\text{FOV}_x} \right) \]

\[ \text{FOV}_x = \frac{1}{\Delta k_x} \]

\[ C(k_x) = \mathcal{F} \{ C(x) \} \]

\[ = \text{FOV}_x \ \text{sinc}(\text{FOV}_x \ k_x) \]

\[ = \frac{1}{\Delta k_x} \ \text{sinc}(\frac{k_x}{\Delta k_x}) \]

**Ideal Kernel**

Decays slowly, used long segments for good approximation.

**Windowed Sinc**

\[ C(k_x) = \frac{1}{\text{FOV}_x} \ \text{sinc} \left( \frac{k_x}{\Delta k_x} \right) \ W(k_x) \]
\[ \frac{1}{k_{\text{eff}}} = \frac{1}{n \Delta x} = \frac{1}{n} F_{0Vx} \]

Window Width

\[ a \approx 2n \]

Computation

\[ (2n)^2 \text{ per data point} \]

\[ N^2 \text{ data points} \]

Want

\[ n \text{ small for computation} \]

\[ n \text{ large for aliasing limitation} \]
SMALL KERNELS

MINIMIZE COMPUTATION WITH A WINDOW FUNCTION

OVERSAMPLING

WE CHOOSE RECONSTRUCTION GRID

IF WE RECONSTRUCT ON A FINEER GRID

\[
\left( \frac{\Delta k_x}{a}, \frac{\Delta k_y}{k} \right) \quad \alpha > 1
\]

THEN

\[
\hat{m}(k_x, k_y) = \left[ (m(k_x, k_y) \cdot s(k_x, k_y)) \ast C(k_x, k_y) \right] \ast \mathcal{W} \left( \frac{k_x}{2\Delta k_x}, \frac{k_y}{2\Delta k_y} \right)
\]

\[
\hat{m}(x, y) = \left[ (m(x, y) \ast s(x, y)) \cdot C(x, y) \right] \ast \mathcal{W} \left( \frac{x}{a \Delta x}, \frac{y}{a \Delta y} \right)
\]
ALLOWS TRANSITION BANDS
LIMITS ALIASING
REDUCES APODIZATION
DIVIDE OUT REMAINING APODIZATION

IMPROVED GRID-DOWN RECONSTRUCTION

\[ \hat{m}(x, y) = \frac{[(m(x, y) + s(x, y)) \cdot c(x, y)] \cdot \omega \left( \frac{x}{\Delta \Phi_{OUx}}, \frac{y}{\Delta \Phi_{OUy}} \right)}{c(x, y)} \]

HOW DO WE CHOOSE \( \omega \), \( c(x, y) \)?

WE'LL COME BACK TO THIS.
Density Correction

 Sampling Pattern is

\[ S(k_x, k_y) = \sum_i^2 S(k_x - k_{xi}, k_y - k_{yi}) \]

Most acquisitions have non-uniform sampling densities.

This creates reconstruction artifacts.

The inverse FT of \( S(k_x, k_y) \) is \( s(x, y) \).

Reconstruction of sampling pattern.

Impulse response or point spread function of the acquisition.

We want \( s(x, y) \) to approximate an impulse.

Basic Idea

If we have uniform data (grid 1's) we want uniform magnitude k-space data.

The impulse response is FT^4 of the I0's. Best we can hope for!
Why is this a problem?

Sampling is seldom uniform rate along trajectory pattern density

Correction options

Ideal: precompensation

\[ \hat{M}(k_x, k_y) = \left[ \left( M(k_x, k_y) \cdot \frac{S(k_x, k_y)}{p(x, y)} \right) + C(k_x, k_y) \right] \int W \left( \frac{k_x}{x_k}, \frac{k_y}{y_k} \right) \]

Density \( p(x, y) \) is compensated on a data sample basis before convolution.

We need to know \( p(x, y) \) first.
Approximate: Post Compensation

\[
\hat{M}(k_x, k_y) = \left[ \frac{M(k_x, k_y) \cdot S(k_x, k_y)) \cdot C(k_x, k_y)}{\rho(k_x, k_y)} \right] W\left(\frac{k_x}{\Delta k_x}, \frac{k_y}{\Delta k_y}\right)
\]

Density is corrected on a grid point basis after convolution.

We can estimate density during gridding by gridding 1's (easier)

Estimating \( \rho(k_x, k_y) \)

**Geometry**
- SPIRAL, PR, LISSAU

In general
- Gridding numerical calculation
GEOMETRY

How much area is associated with each sample. This gives \( \frac{1}{D} \), compensation filter.

\[ \text{DC Disc} \]

\[ \frac{1}{N} \pi \left( \frac{\delta k}{2} \right)^2 \]

\[ \text{1st Sample} \]

\[ \frac{\pi}{2N} \left( \left( \frac{3\delta k}{2} \right)^2 - \left( \frac{\delta k}{2} \right)^2 \right) \]

\[ \text{n th Sample} \]

\[ \frac{\pi}{N} \left( \frac{\delta k}{2} \right)^2 n \]

Discrete Rho Filter

Not zero at Origin!

Similar Arguments for Spirals, Others

GRIDDING DENSITY

Estimate \( \rho(k_x, k_y) \) by gridding 1's:

\[ \hat{\rho}(k_x, k_y) = \left[ S(k_x, k_y) \ast C(k_x, k_y) \right] \cdot \left\lfloor \frac{k_x}{\delta x/n}, \frac{k_y}{\delta y/m} \right\rfloor \]

This is on Grid Points
Post compensation

OK if \( \rho(k_x, k_y) \) varies slowly

Fails when \( \rho(k_x, k_y) \) changes rapidly

Common in older papers
Example

\[ \rho(k_x, k_y) \sim \frac{1}{\sqrt{2\pi} \sigma} \]

\[ C(k_x, k_y) \]

\[ \tilde{\rho}(k_x, k_y) \]

Low frequencies blurred out

Produces baseline and low frequency artifacts

Odd! This is where we have the most data
Voronoi Diagram

Partition plane into polygons closest to each sample

Density is 1/Area
Fast, part of MATLAB
Edge samples unbounded

Another numerical perspective

Gridding 1/Density should produce uniform k-space data
Gridding convolution can be written as a matrix equation

\[
\begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
\end{bmatrix}
= \begin{bmatrix}
C(k_i-k_{i+1}) & 0 & \cdots \\
0 & C(k_i-k_{i+1}) & \cdots \\
\vdots & \vdots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
\end{bmatrix}
\]
OR

\[ d_s = G d_s \]

We want to find weighting vector

\[ w = \frac{1}{f} \]

Such that

\[ 1 = \underline{G} w \]

If we grid \( w \), we get 1's on the grid points.

Larger problem, but \( G \) is sparse.

Easily solved in MATLAB with \texttt{lsqr( )}

**Unusual characteristics**

Weighting no longer associated with area or density.

Depends somewhat on \( C(k_x, k_y) \).

Not always positive.
Density Compensation

Depends on your problem

Post compensation + training may be enough and is easy

Pre compensation is better

More difficult to compute

Can combine the two

Good initial estimate with pre compensation

Correct for actual data with post compensation

Next time

Kernel design

Oversampling ratio

Kernel sampling