



# Simulated Tomography of Beam Transverse Phase Space

Adam Watts

Switchyard Meeting

August 2016

# Purpose

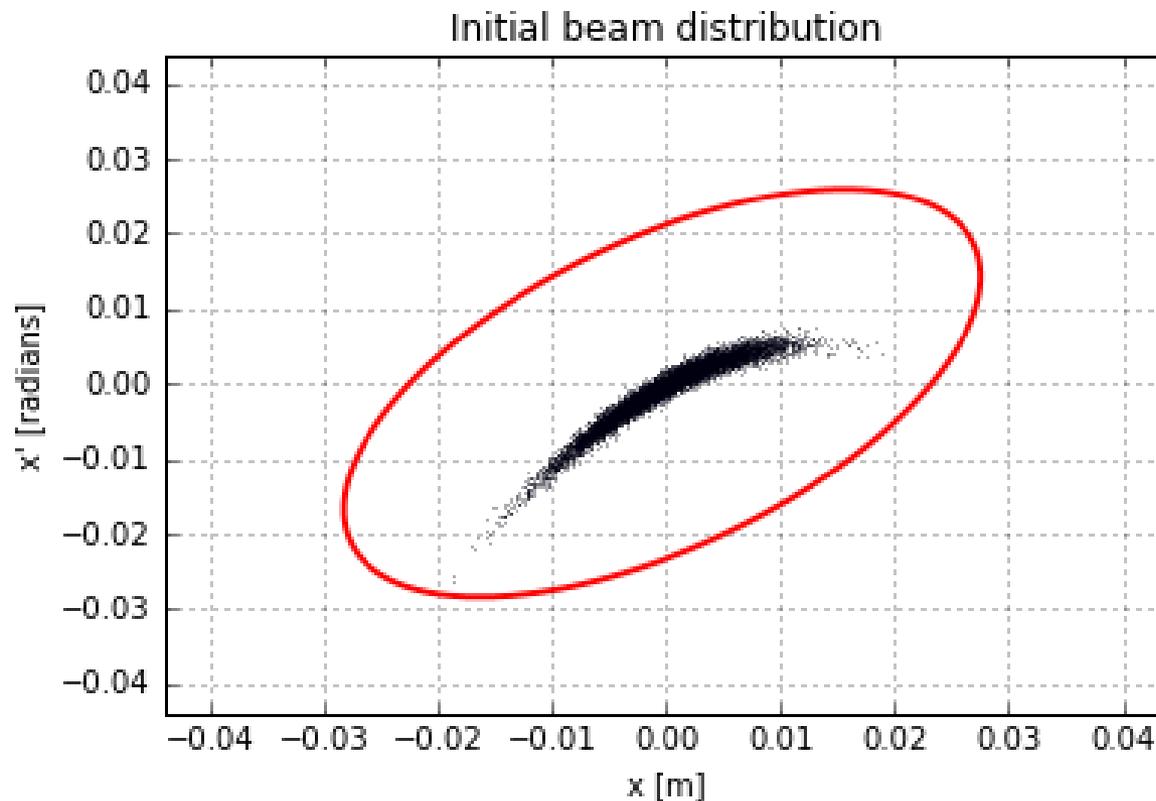
The non-linearities inherent to resonant extraction prevent us from reasonably assuming that the extracted phase space is Gaussian. Thus Courant-Snyder parameters are not sufficient to accurately describe the Switchyard beam.

In order to better match the Switchyard optics to the actual shape of the resonantly-extracted beam, we set out to directly reconstruct the phase space distribution using Computed Tomography.

So far, I have built a beam simulation to test reconstruction methods and determine their limitations. These simulations inform not only which methods we should use, but how we should take the beam data in Switchyard.

# Purpose

By imaging the phase space distribution of the beam entering the P1 line, we can determine the effective ellipse that encompasses the beam (tails and all). By matching the beamline with the encompassing ellipse as initial conditions, we can improve the efficiency of MI-SY extraction. Shown below is a sample beam distribution and the encompassing ellipse.



# Computed Tomography

A two-dimensional image may be reconstructed from several one-dimensional projections. The reconstruction process is known as “computed tomography”.

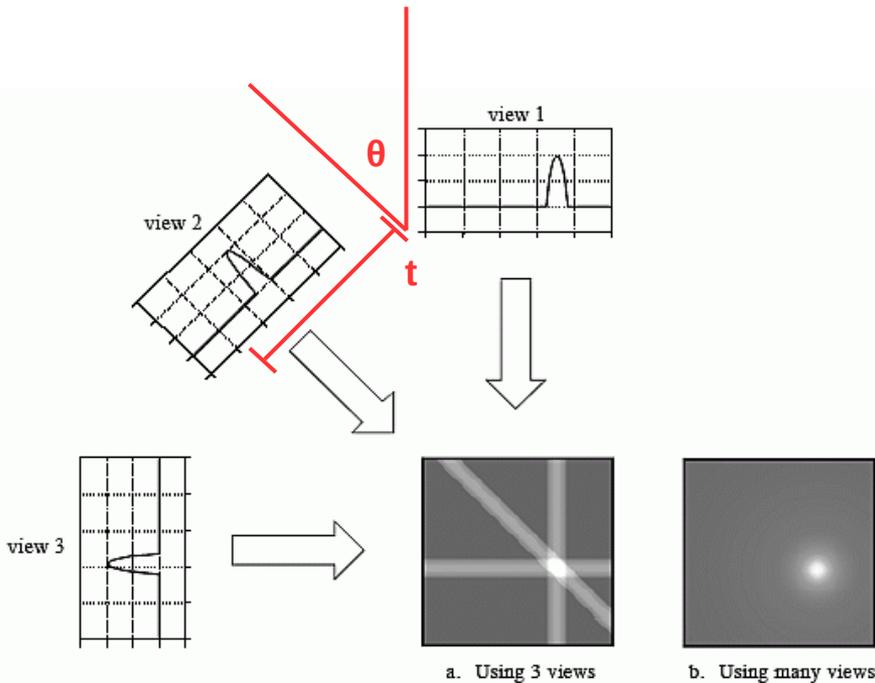


FIGURE 25-16 Backprojection. Backprojection reconstructs an image by taking each view and *smearing* it along the path it was originally acquired. The resulting image is a blurry version of the correct image.

**Source:** <http://www.dspguide.com/ch25/5.htm>

**Projection:**

$$P(t, \theta) = \iint dx dy f(x, y) \delta(x \cos \theta + y \sin \theta - t)$$

**Reconstruction:**

$$f(x, y) = \iint \omega d\omega d\theta e^{2\pi i(\omega x \cos \theta + \omega y \sin \theta)} R(\omega, \theta)$$

The reconstruction involves computing the Fourier Transform  $R(\omega, \theta)$  of each profile  $P(t, \theta)$ , then simultaneously back-projecting through  $t = x \cos(\theta) + y \sin(\theta)$  and computing the inverse Fourier Transform, then finally integrating over all projection angles.

# Filtered Back-Projection Computed Tomography

By filtering each projection before reconstruction, a more accurate depiction of the original object is achieved. This is one of the most common and efficient computed tomography algorithms, known as “Filtered Back-Projection” (FBP).

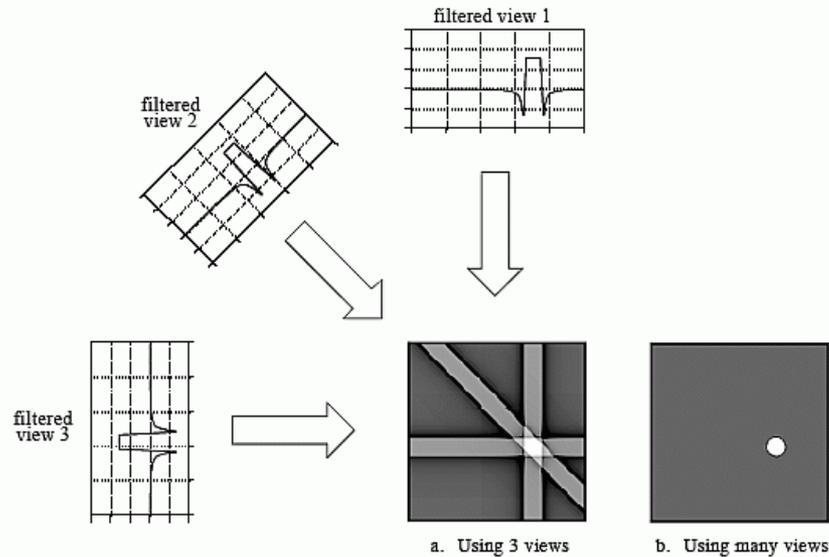


FIGURE 25-17

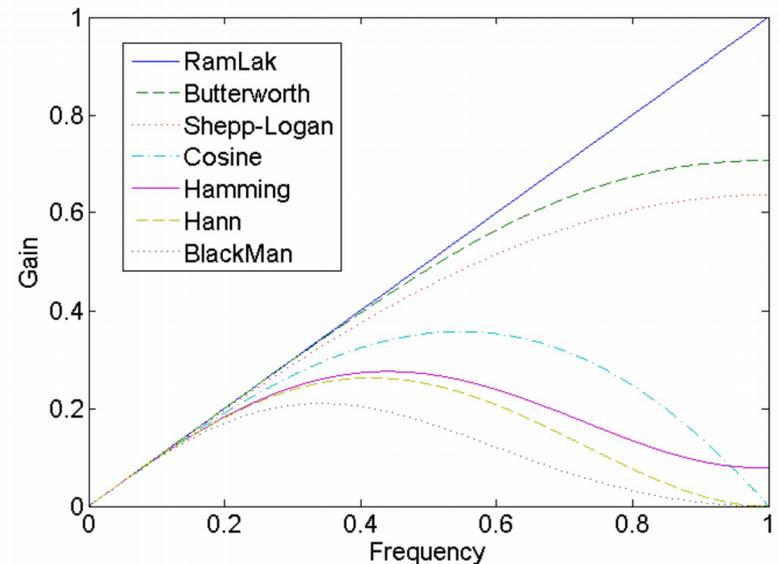
Filtered backprojection. Filtered backprojection reconstructs an image by filtering each view before backprojection. This removes the blurring seen in simple backprojection, and results in a mathematically exact reconstruction of the image. Filtered backprojection is the most commonly used algorithm for computed tomography systems.

Source: <http://www.dspguide.com/ch25/5.htm>

FBP Reconstruction:

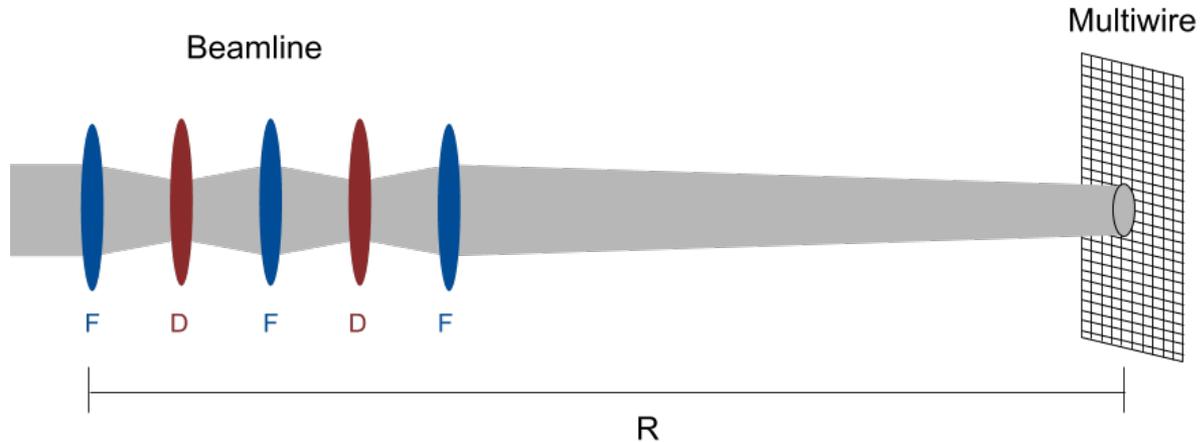
$$f(x, y) = \iint \omega d\omega d\theta e^{2\pi i(\omega x \cos\theta + \omega y \sin\theta)} F(\omega) R(\omega, \theta)$$

Filter



# FBP Beam Analogy

To apply the FBP algorithm, we must make an analogy between beam optics and projections of an object at several different angles by modifying the beam projections as measured by the multiwire.



**Projection (beam profile):**

$$\begin{pmatrix} x \\ x' \end{pmatrix} = R \begin{pmatrix} x_0 \\ x'_0 \end{pmatrix}$$

$$P(x_0, \theta) = \iint dx dx' f(x, x') \delta(m_{11}x + m_{12}x' - x_0)$$

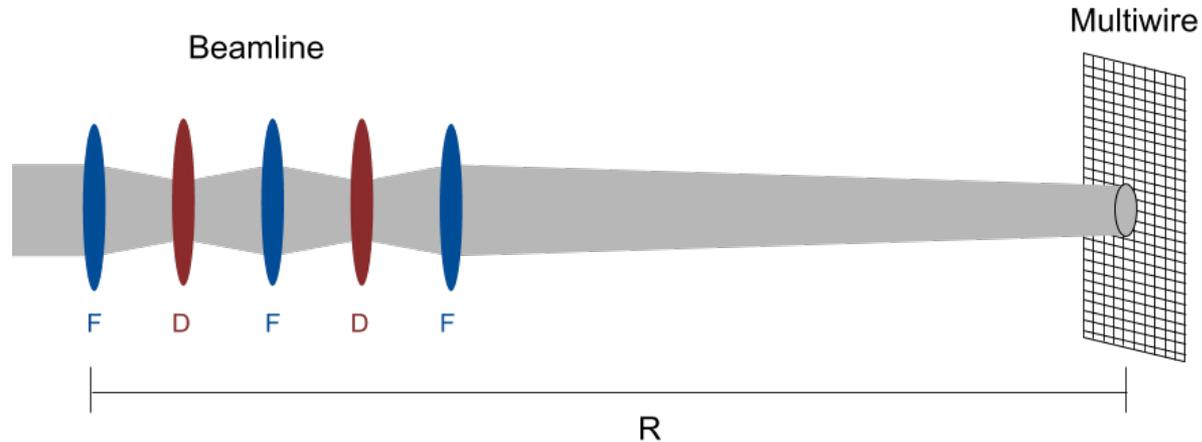
$$R = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$$

**Compare to general projection:**

$$P(t, \theta) = \iint dx dy f(x, y) \delta(x \cos \theta + y \sin \theta - t)$$

# FBP Beam Analogy

To apply the FBP algorithm, we must make an analogy between beam optics and projections of an object at several different angles by modifying the beam projections as measured by the multiwire.



$$s = \sqrt{m_{11}^2 + m_{22}^2}$$

$$\cos\theta = \frac{m_{11}}{s}$$

$$\sin\theta = \frac{m_{12}}{s}$$

$$\theta = \tan^{-1}\left(\frac{m_{12}}{m_{11}}\right)$$

**Modified beam projection:**

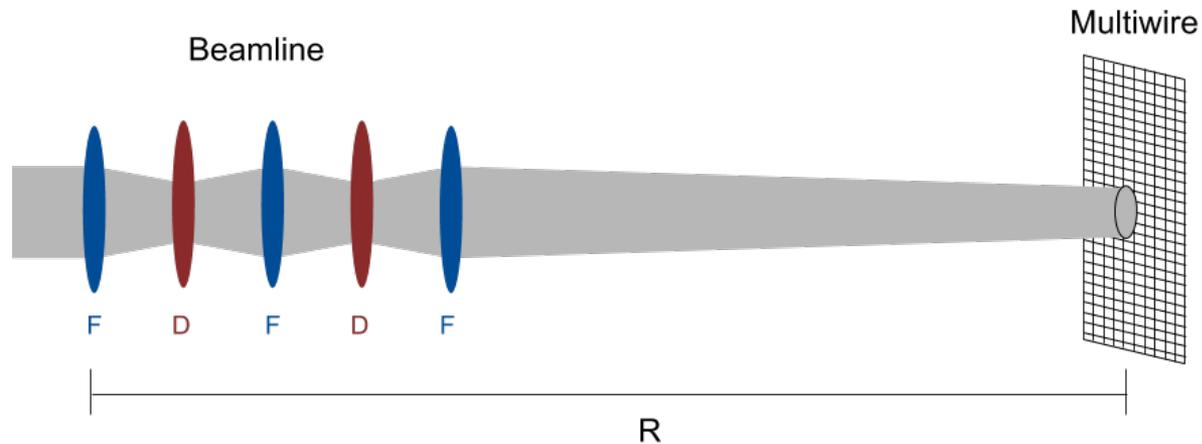
$$P(x_0, \theta) = \iint dx dx' f(x, x') \delta\left(s\left[m_{11}x + m_{12}x' - \frac{x_0}{s}\right]\right)$$

**Compare to general projection:**

$$P(t, \theta) = \iint dx dy f(x, y) \delta(x \cos\theta + y \sin\theta - t)$$

# FBP Beam Analogy

To apply the FBP algorithm, we must make an analogy between beam optics and projections of an object at several different angles by modifying the beam projections as measured by the multiwire.



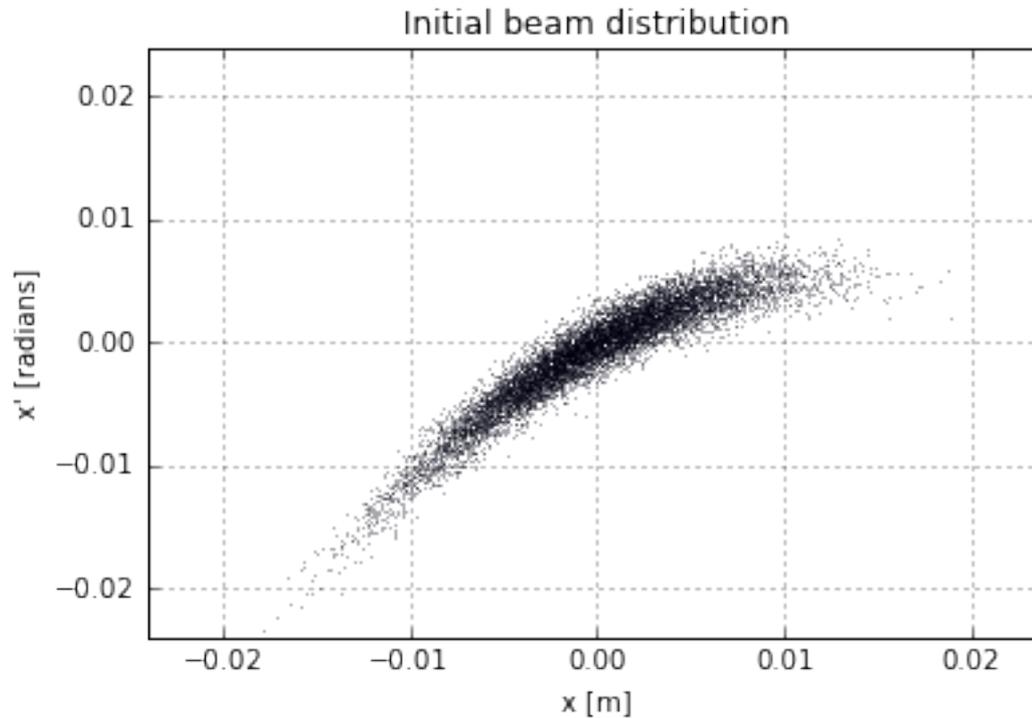
To summarize:

- Vary  $R$  matrix between multiwire and point at which we want to reconstruct beam phase space (i.e. scan quadrupoles).
- Take beam profile for each quadrupole setting.
- Scale beam profiles vertically by  $s$  and horizontally by  $1/s$ .
- Apply FBP reconstruction algorithm on scaled profiles, integrating over phase space orientation angle  $\theta$ .
- \*\*Note that  $\theta$  is *not* the betatron phase, and is independent of the beam.

# Beam simulation (Python code)

Generate initial random beam distribution with asymmetry and tails:

```
sigmax = 0.005  
sigmaxp = sigmax/4  
x0 = np.random.normal(0, sigmax, 10000)  
xp0 = -30*x0**2 + 0.8*x0 + np.random.normal(0, sigmaxp, 10000)
```



# Beam simulation (Python code)

Pass each particle through linear optics, i.e. simple FODO channel:

```
L = 1 # drift length [m]
O = np.array([[1,L],[0,1]])
F = np.array([
    [np.cos(k**0.5), (k**-0.5)*np.sin(k**0.5)],
    [-(k**0.5)*np.sin(k**0.5), np.cos(k**0.5)]
])
D = np.array([
    [np.cosh(k**0.5), (k**-0.5)*np.sinh(k**0.5)],
    [-(k**0.5)*np.sinh(k**0.5), np.cosh(k**0.5)]
])

R = np.linalg.multi_dot([O,D,O,F,O,D,O,F,O,D,O,F])

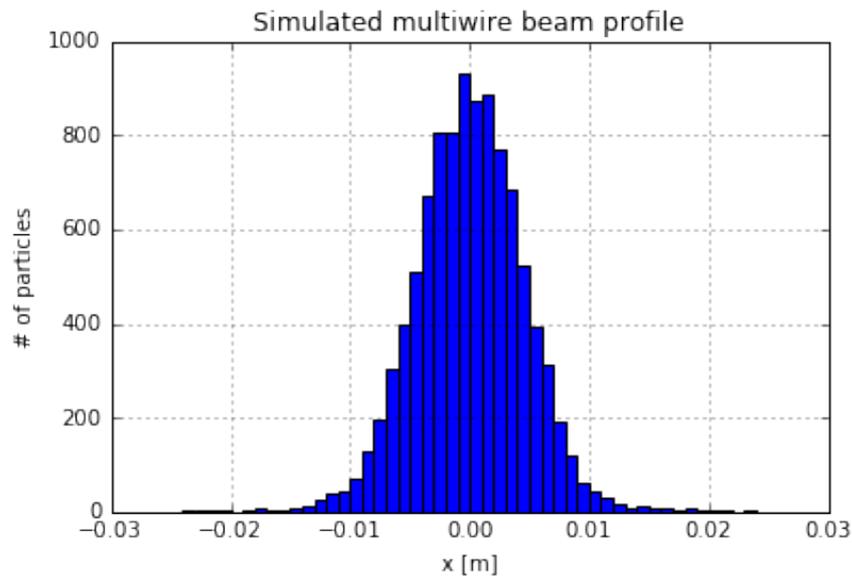
xnew = R[0][0]*x + R[0][1]*xp
xpnew = R[1][0]*x + R[1][1]*xp
```

# Beam simulation (Python code)

Simulate a multiwire profile by using a fixed-width fixed-bin histogram on all “x” values:

```
num_wires = 48
wire_pitch = 0.001
low_x_lim = -(num_wires/2)*wire_pitch
high_x_lim = (num_wires/2)*wire_pitch
binBoundaries = np.linspace(low_x_lim,high_x_lim,num_wires+1)

hystarray = plt.hist(xnew,bins=binBoundaries)[0]
```



# Beam simulation (Python code)

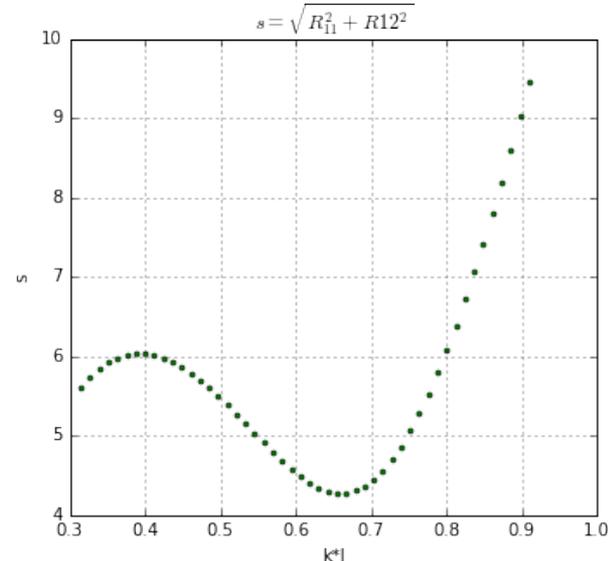
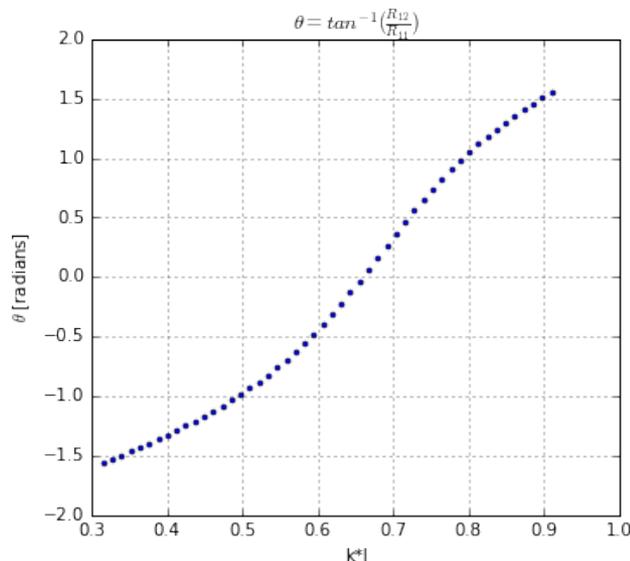
Repeat process for each new beamline tune by varying the k values of all the quadrupoles. Thus we collect beam profiles for each new phase space orientation angle  $\theta$ . We want to have “enough” projections at angles spanning as close to 180 degrees as possible.

```
k_array = np.linspace(0.315,0.91,50)
```

After each profile is taken, calculate the scaling factor and orientation angle for that tune.

```
theta = np.arctan(R[0][1]/R[0][0])  
s = np.sqrt(R[0][0]**2+R[0][1]**2)
```

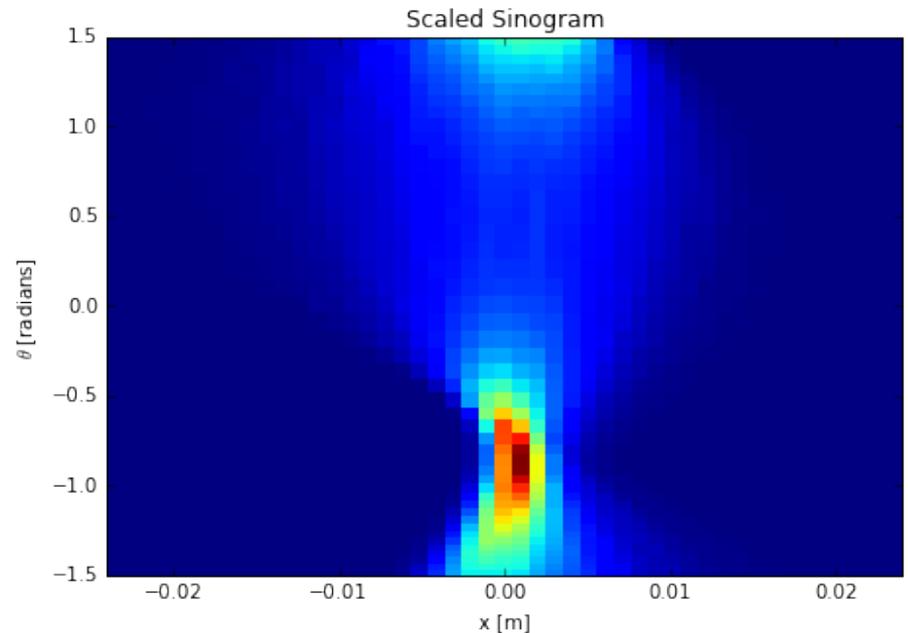
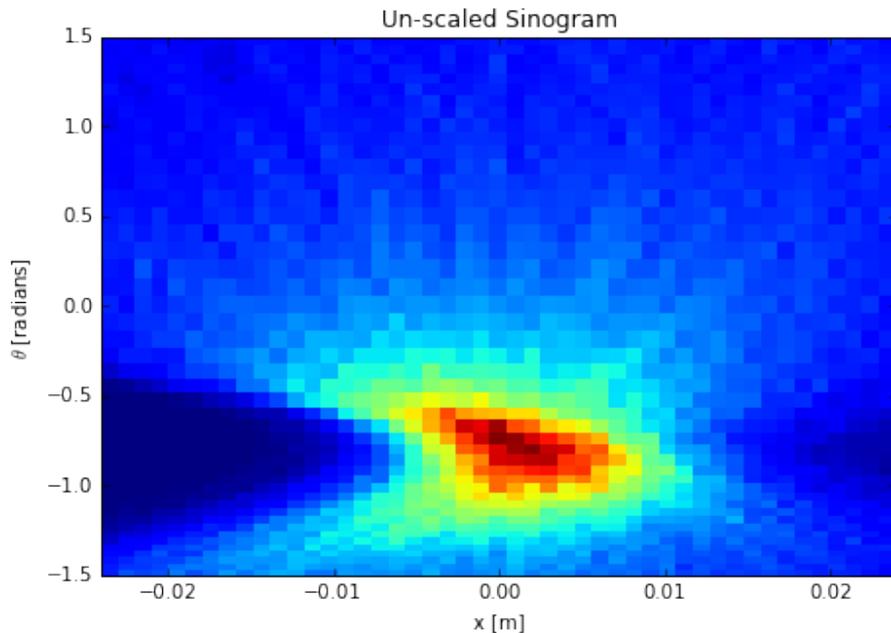
Choice of the k values in the array depends on how many projections we want (i.e. “a lot”), and how wide a range of  $\theta$  we can achieve.



# Beam simulation (Python code)

Collect all the profiles into a single structure known as a *sinogram* that summarizes how the beam profile changed as a function of orientation angle. Then scale each profile vertically by  $s$  and horizontally by  $1/s$ .

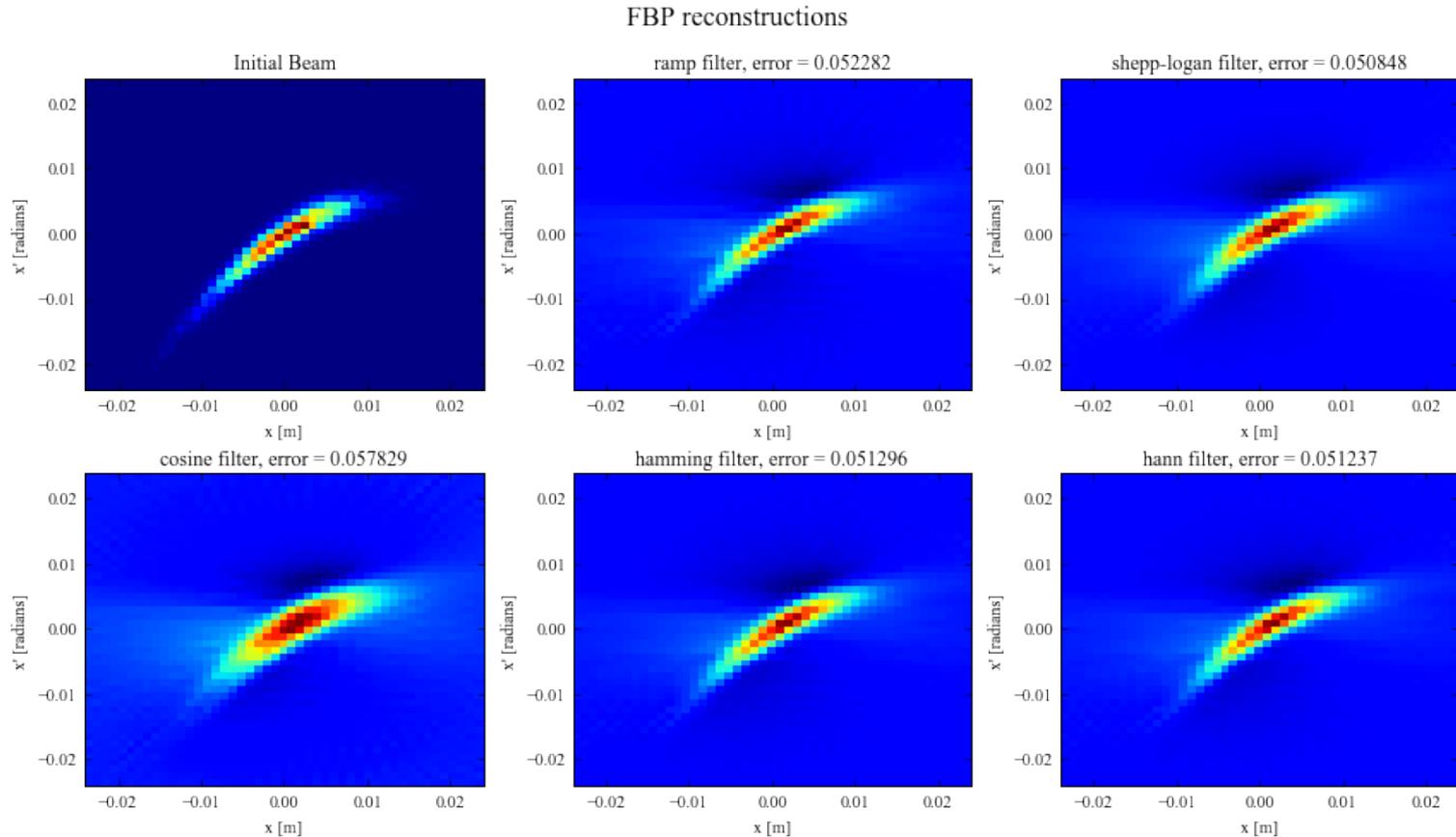
Scaled and un-scaled sinograms are shown below for comparison. Note that there is clipping of the beam tails due to the finite size of the multiwire.



# Beam simulation, FBP reconstruction

First we use the Python package “sci-kit image”, and in particular the “iradon” module, to convert the scaled sinogram data into a reconstructed transverse phase space. There are options for several different filters, and the following plot compares the performance of each.

Since we know the initial beam distribution we're trying to reconstruct, we can subtract the reconstruction from the original distribution to compute an RMS error for the reconstruction.

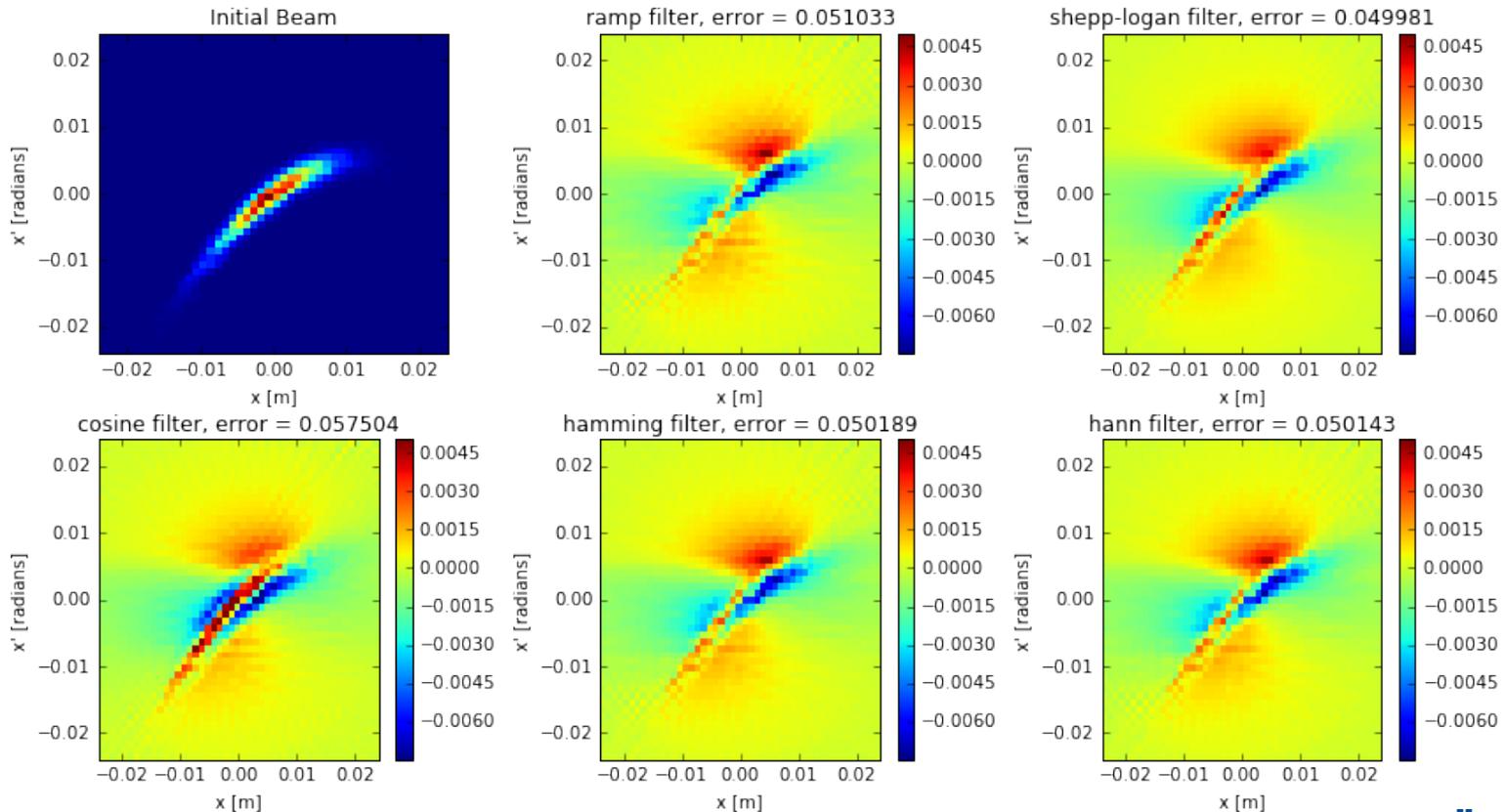


# Beam simulation, FBP errors

These plots show the difference error between the initial beam and the reconstruction. Both initial and reconstructed distributions have each pixel normalized to the sum of all pixel values; the resulting pixel values represent a probability, where the sum of all pixel values is 1.0.

Pictured below is the difference error *original* – *reconstructed*. Thus positive (red) errors correspond to pixels missing from the reconstruction, and negative (blue) errors correspond to artifacts added by reconstruction.

FBP errors



# Simultaneous Algebraic Reconstruction Technique

Another readily-available reconstruction method exists in the “sci-kit image” package, known as Simultaneous Algebraic Reconstruction Tomography (SART), which is advantageous when projection data is limited.

SART is an iterative method that solves an under-constrained system of linear equations to create a reconstruction. The reconstruction is then fed into the algorithm again as an initial “guess”.

The results are more promising than the single-iteration FBP algorithm, though there are free parameters that must be chosen to fine-tune the result.

Source: A. C. Kak and Malcolm Slaney, *Principles of Computerized Tomographic Imaging*, IEEE Press, 1988.

# Simultaneous Algebraic Reconstruction Technique

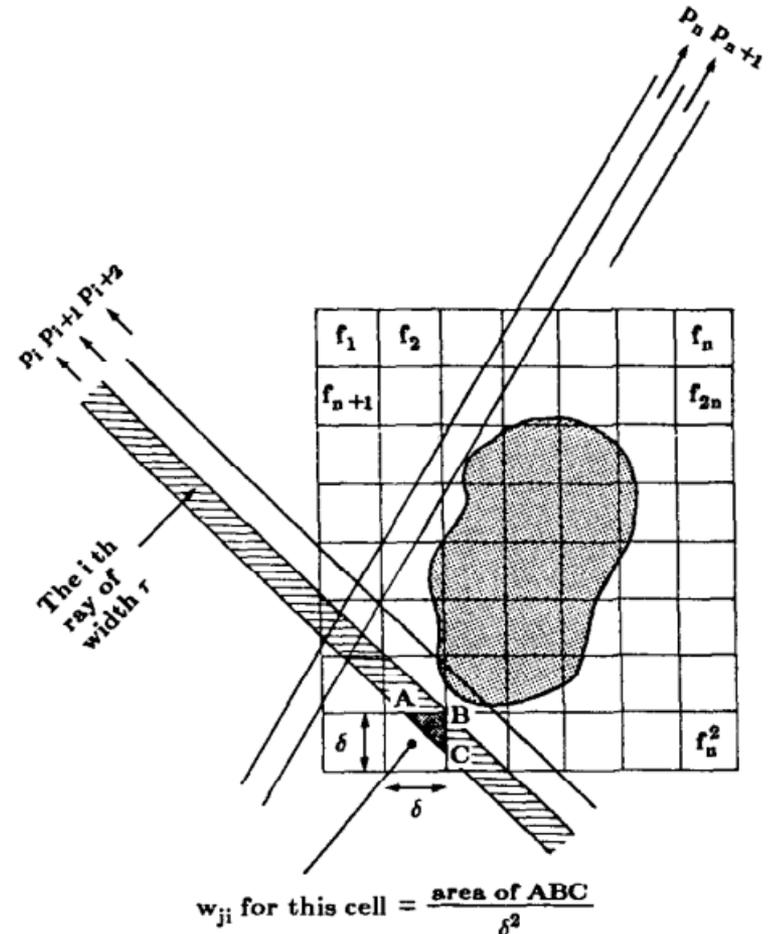
If the original image to be reconstructed is overlain with a grid, or represented as an array of pixels, we can write the following linear system of equations that describe each one-dimensional projection. For  $N$  pixels in the original image:

$$\sum_{j=1}^N w_{ij} f_j = p_i$$

The matrix elements  $w_{ij}$  represent the fractional area of a pixel subtended by the  $j^{\text{th}}$  imaging ray. Thus each profile  $p_i$  is the sum of the fractional area subtended by all imaging rays, weighted by the value of the original image's pixel  $f_j$ .

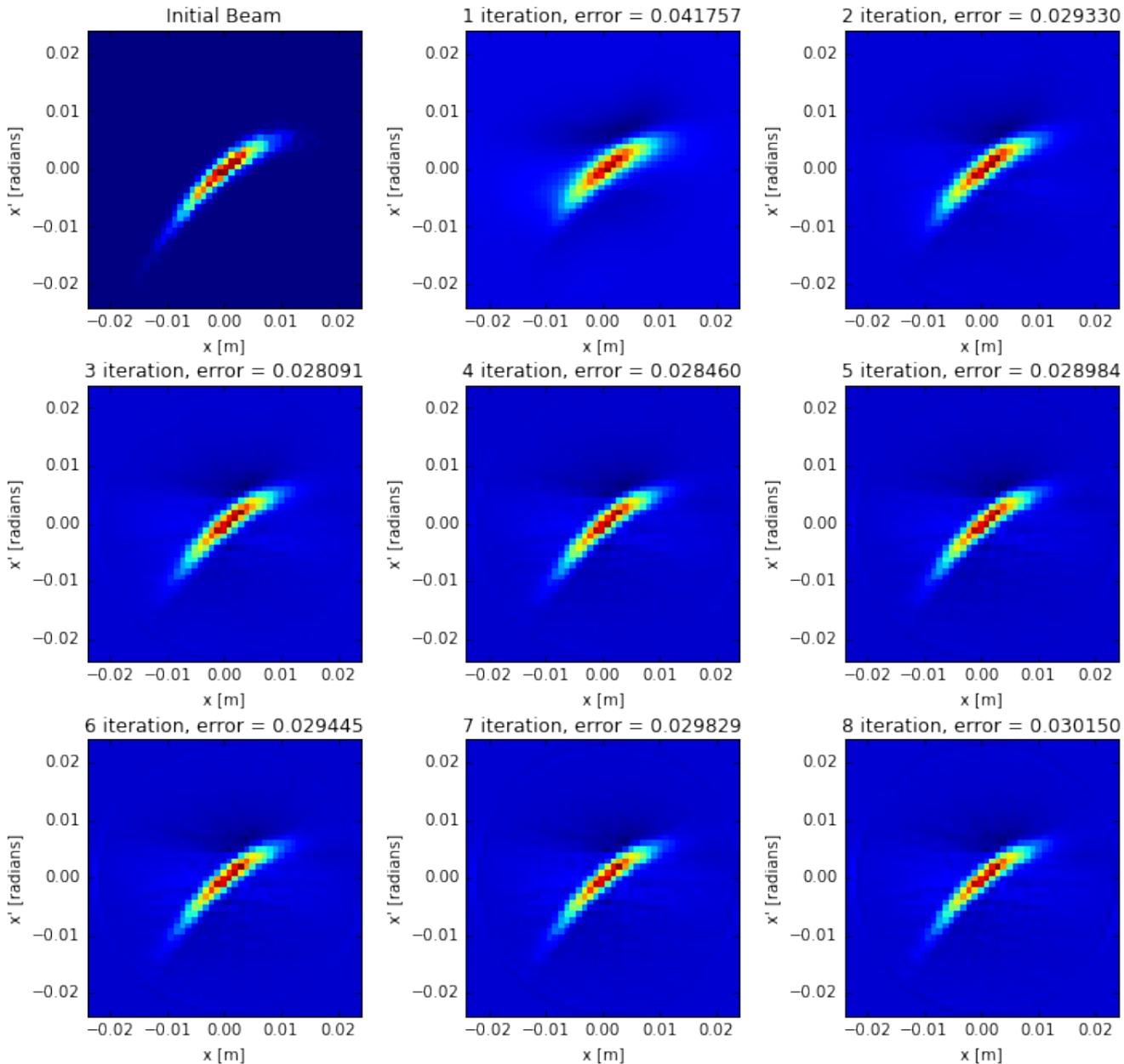
Inverting this system and solving for the values of every pixel  $f_j$  reconstructs the original image discretely, and is the purpose of SART.

For  $N$  pixels in the original image:

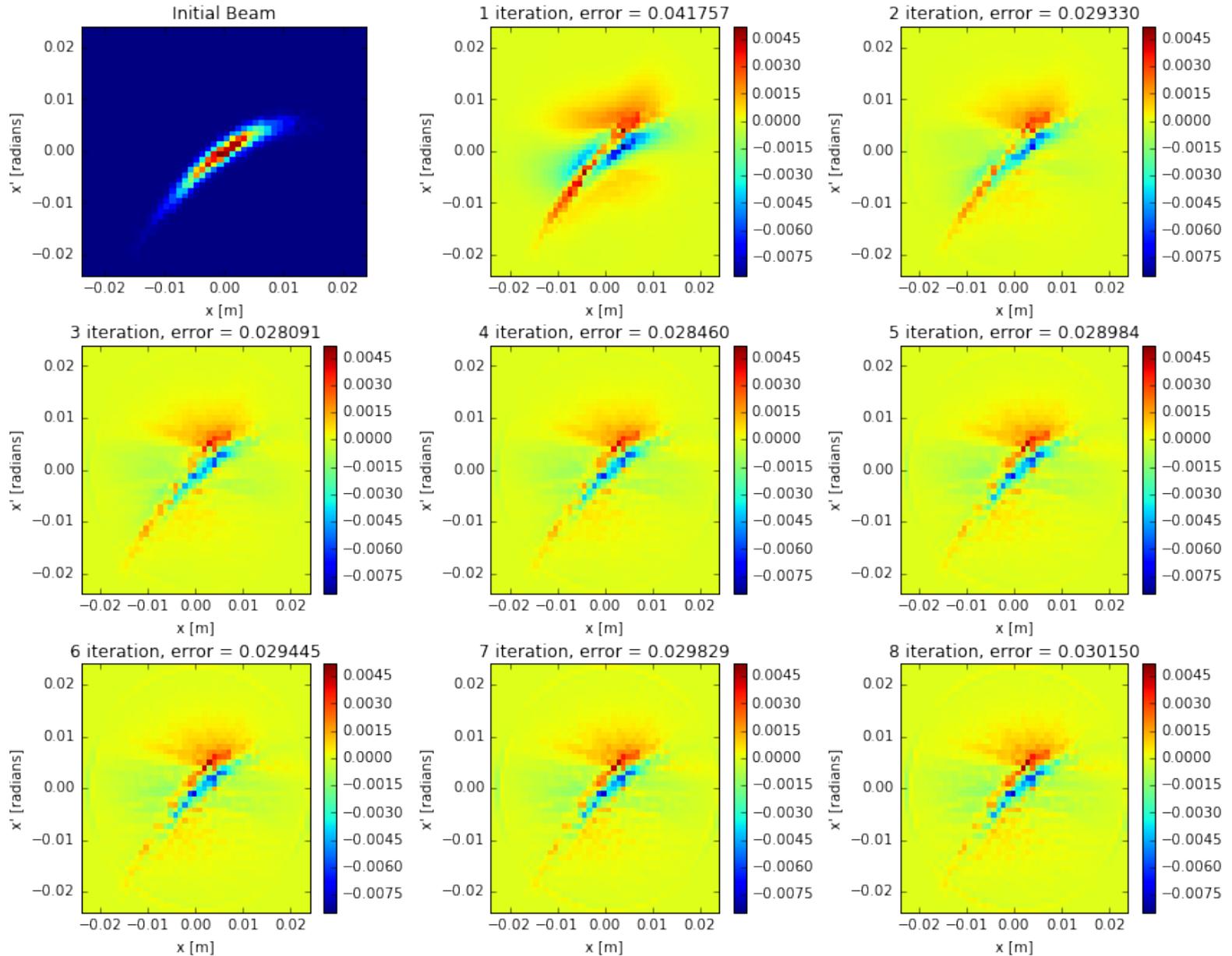


Source: A. C. Kak and Malcolm Slaney, *Principles of Computerized Tomographic Imaging*, IEEE Press, 1988.

# SART reconstructions, relax = 0.1



# SART errors, relax = 0.1

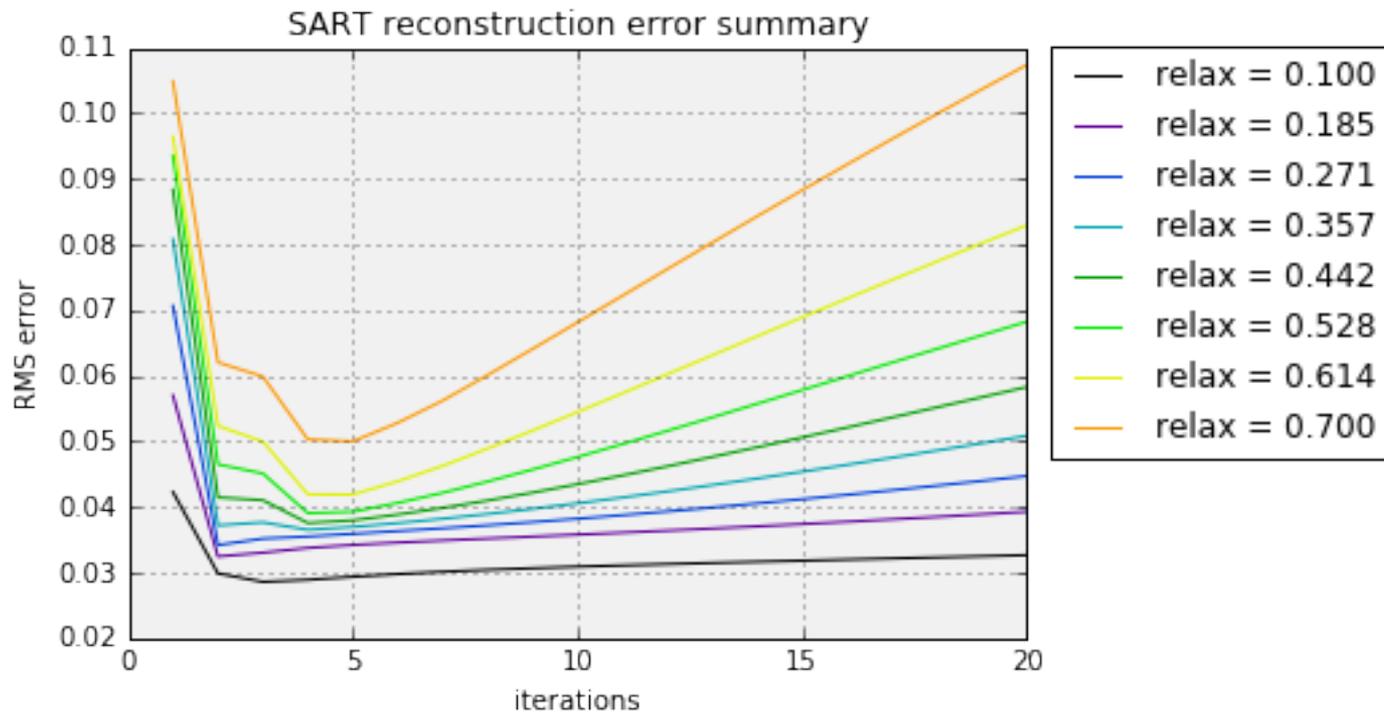


# Simultaneous Algebraic Reconstruction Tomography

Now we vary the “relaxation” parameter and investigate the quality of reconstruction as a function of successive iterations. The RMS error is used as the figure of merit to determine reconstruction quality.

The resulting plot helps determine the combination of relaxation parameter and number of iterations that produce the best reconstruction, i.e. that which is most faithful of the original distribution.

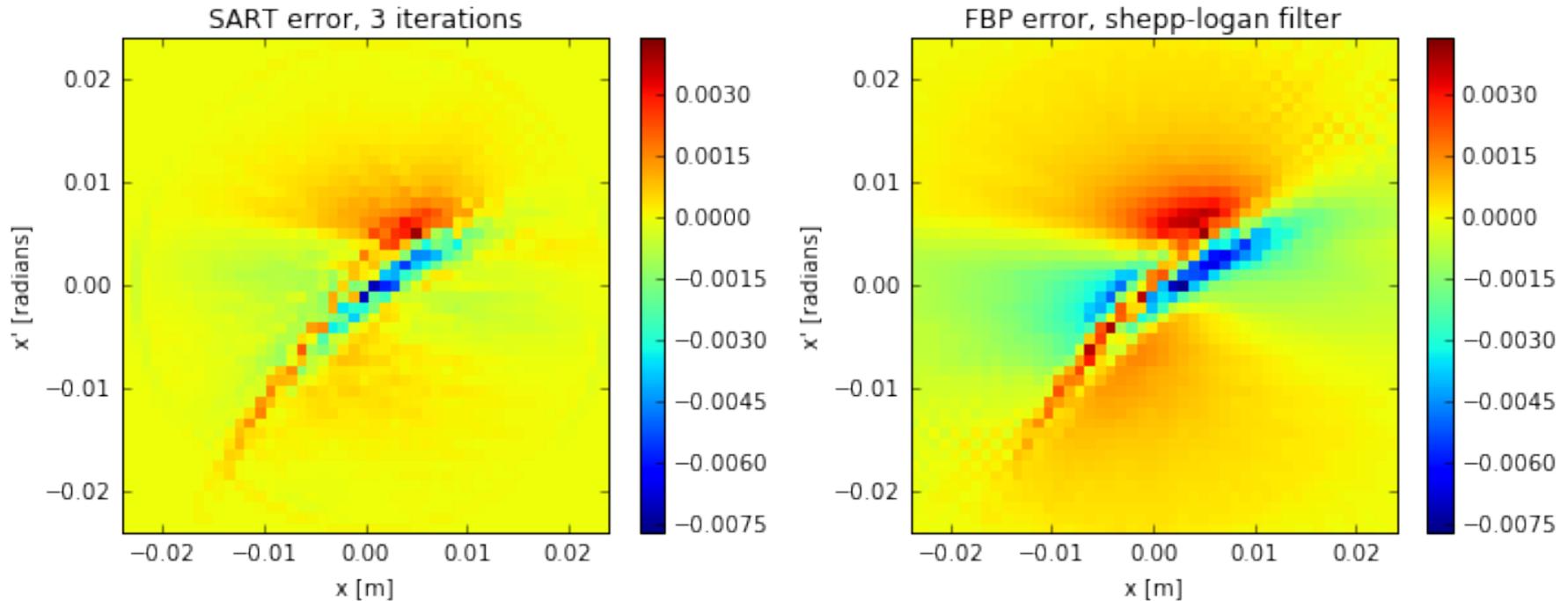
It is apparent that increased iterations and relaxation value contribute to increased artifacts, presumably due to the propagation of noise/artifacts from previous iterations. A balance between iterations and relaxation provides best overall result (i.e. lowest RMS error)



# SART vs. FBP

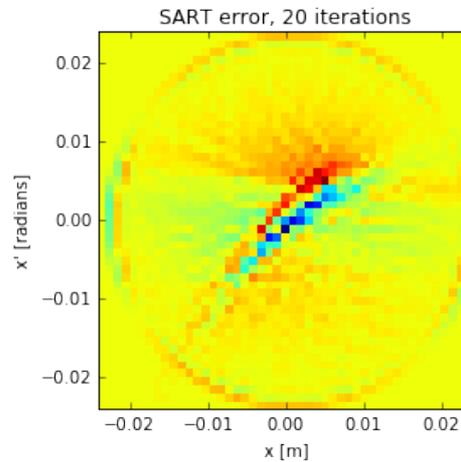
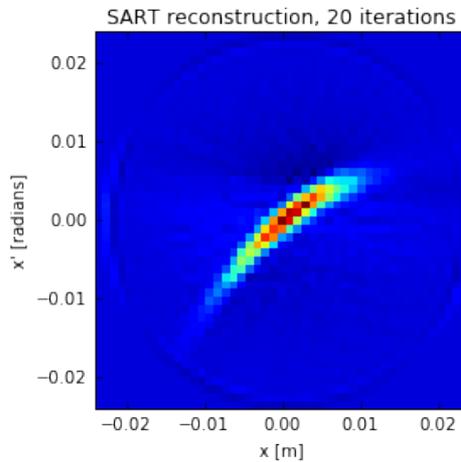
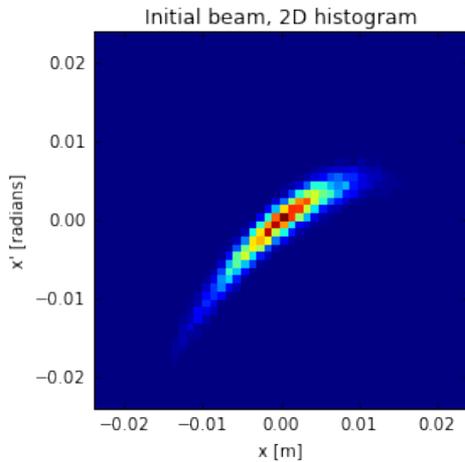
This is a qualitative comparison of the errors for both FBP and SART, using the parameters for each that returned the lowest RMS error respectively.

It is apparent that SART produces less artifacts (blue) *and* misses less data in the reconstruction (red) than FBP.

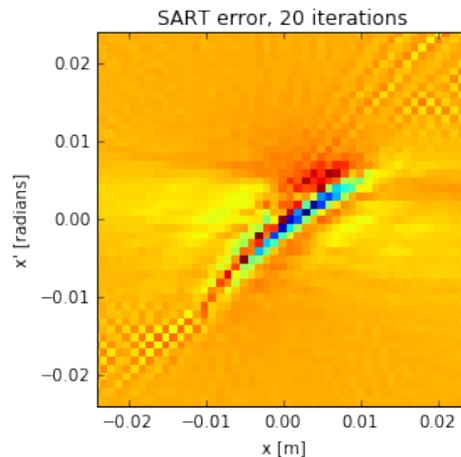
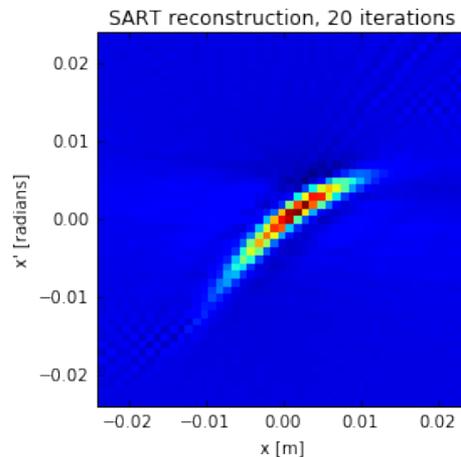
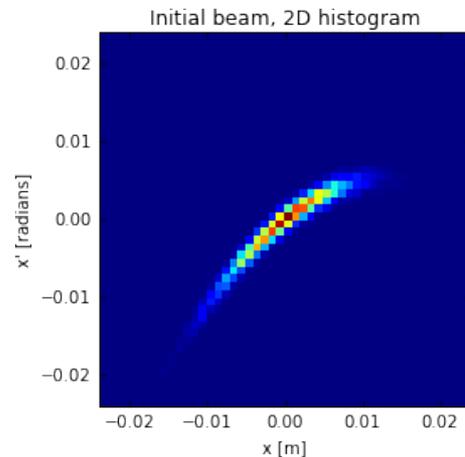


# SART with many iterations

Tail reconstruction appears improved for larger iteration values, though artifacts begin to dominate. Over-iteration seems to improve tail reconstruction somewhat, but at the cost of “salt and pepper” noise, as well as phantom ring surrounding the beam. This ring is not present for a bigger multiwire, so presumably the ring artifact is due to beam clipping on the multiwire aperture.



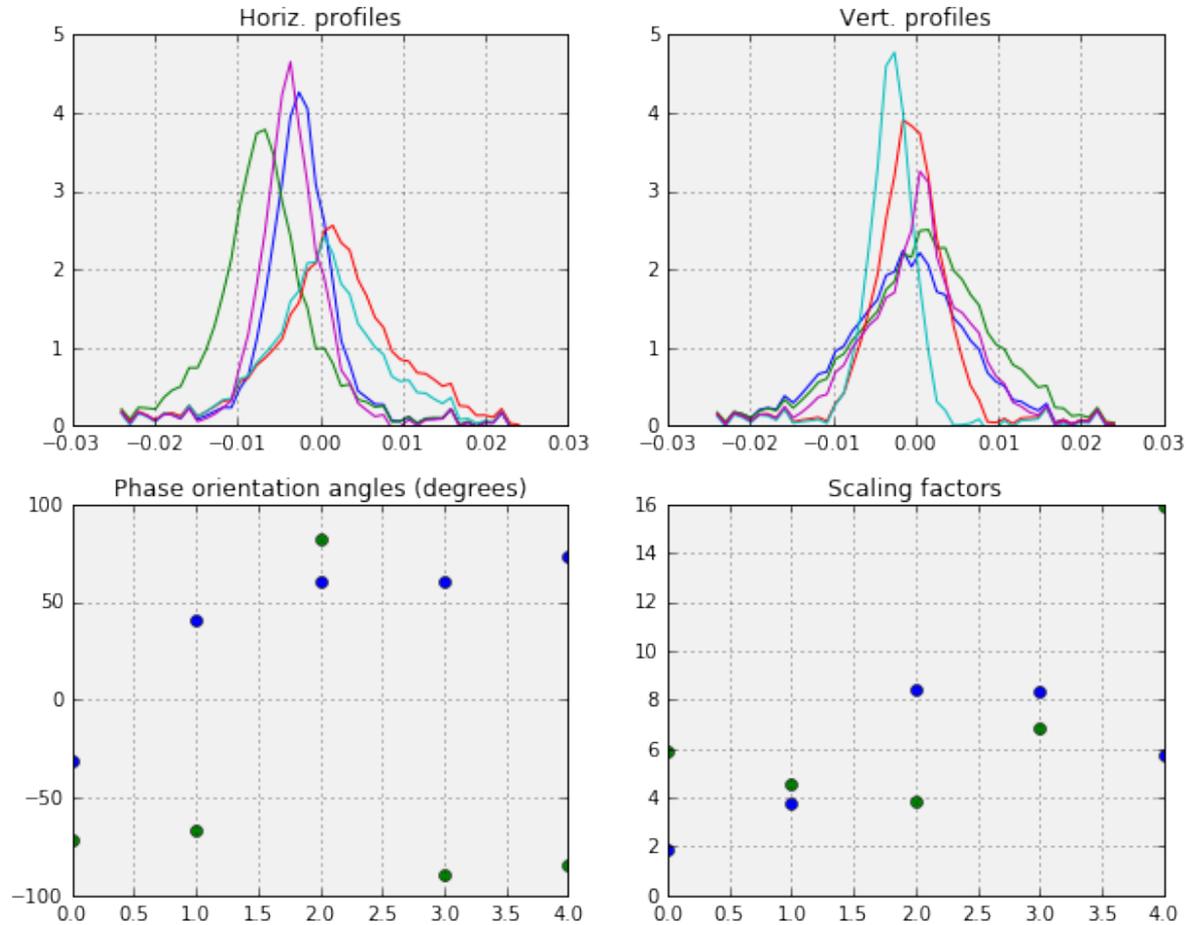
**4.8mm-wide  
multiwire**



**19.2cm-wide  
multiwire**

# MTA beam reconstructed @ C-magnet

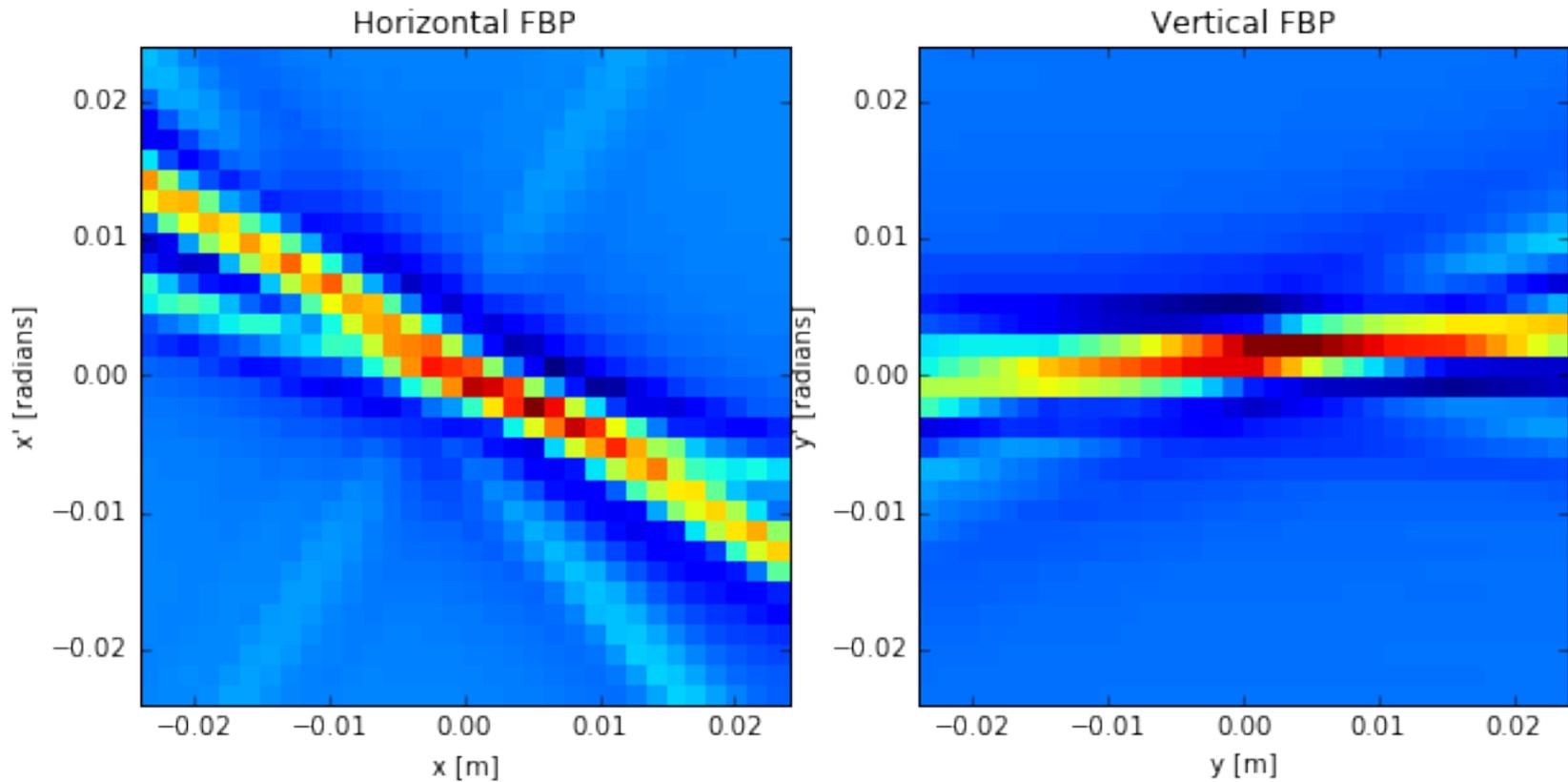
Multiwire data is limited for MTA. Only 5 tunes are recorded for this reconstruction, not completely spanning the full 180 degrees needed for the phase orientation angle. Note that we are particularly limited in phase orientation angle range in the vertical plane. Also note that the scaling factor and phase orientation angle were calculated using the MAD deck, which could be a source of error.



# MTA beam, FBP

FBP reconstruction with the Shepp-Logan filter is pictured below. Clearly more data is needed to make any conclusions from the reconstruction.

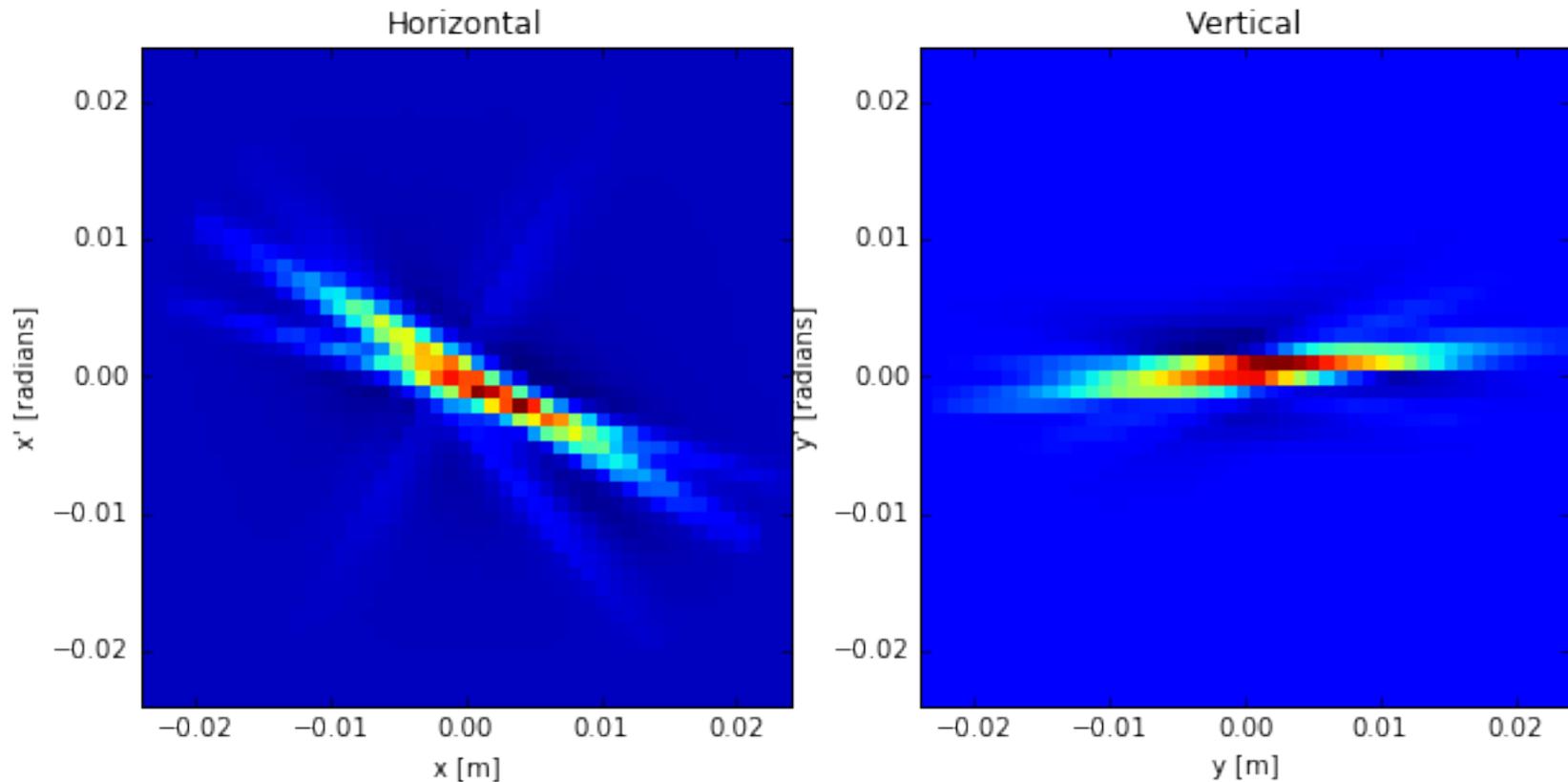
MTA FBP reconstruction, Shepp-Logan filter



# MTA beam, SART

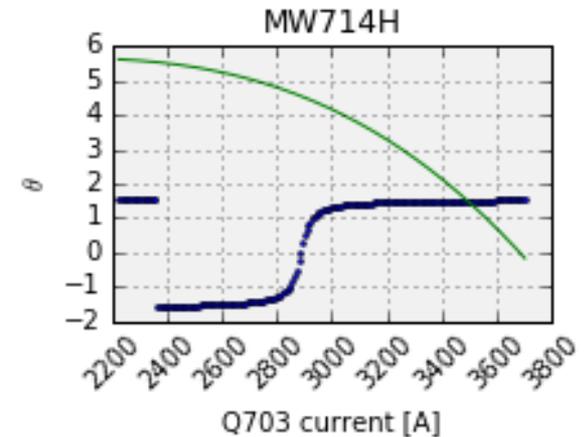
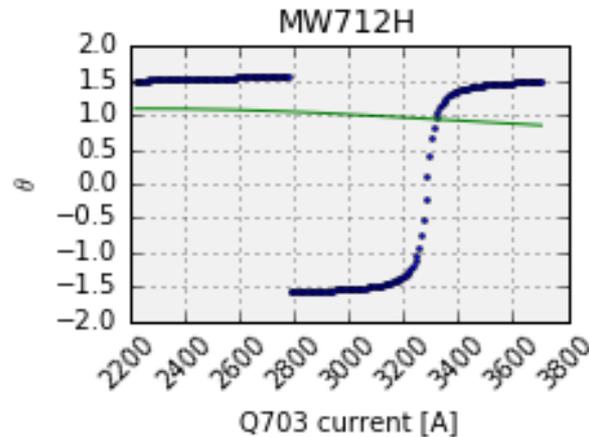
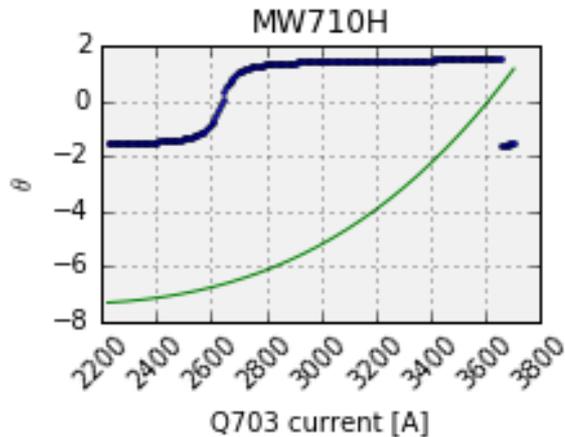
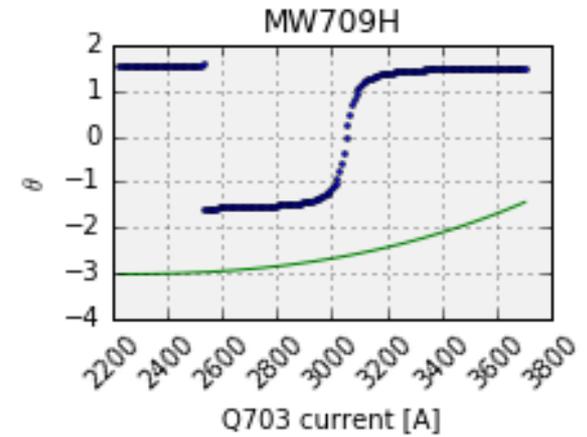
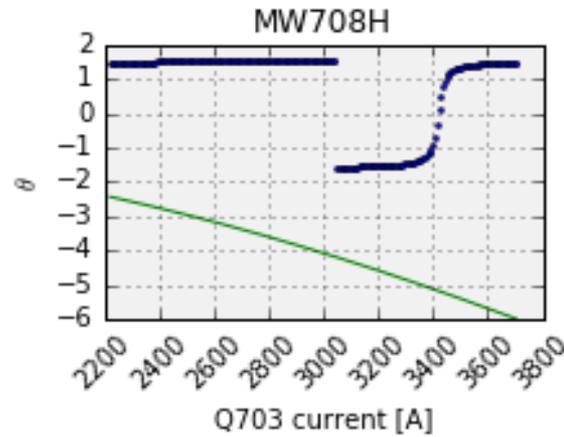
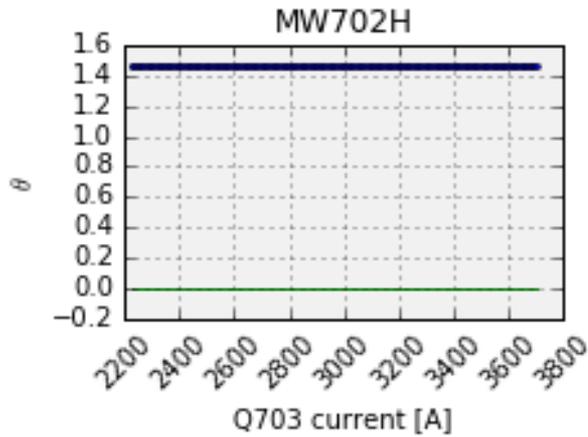
SART reconstruction with 2 iterations is pictured below. While better than the FBP reconstruction, more data is needed for a proper reconstruction to determine whether the beam can be considered elliptical.

MTA SART reconstruction, 2 iterations



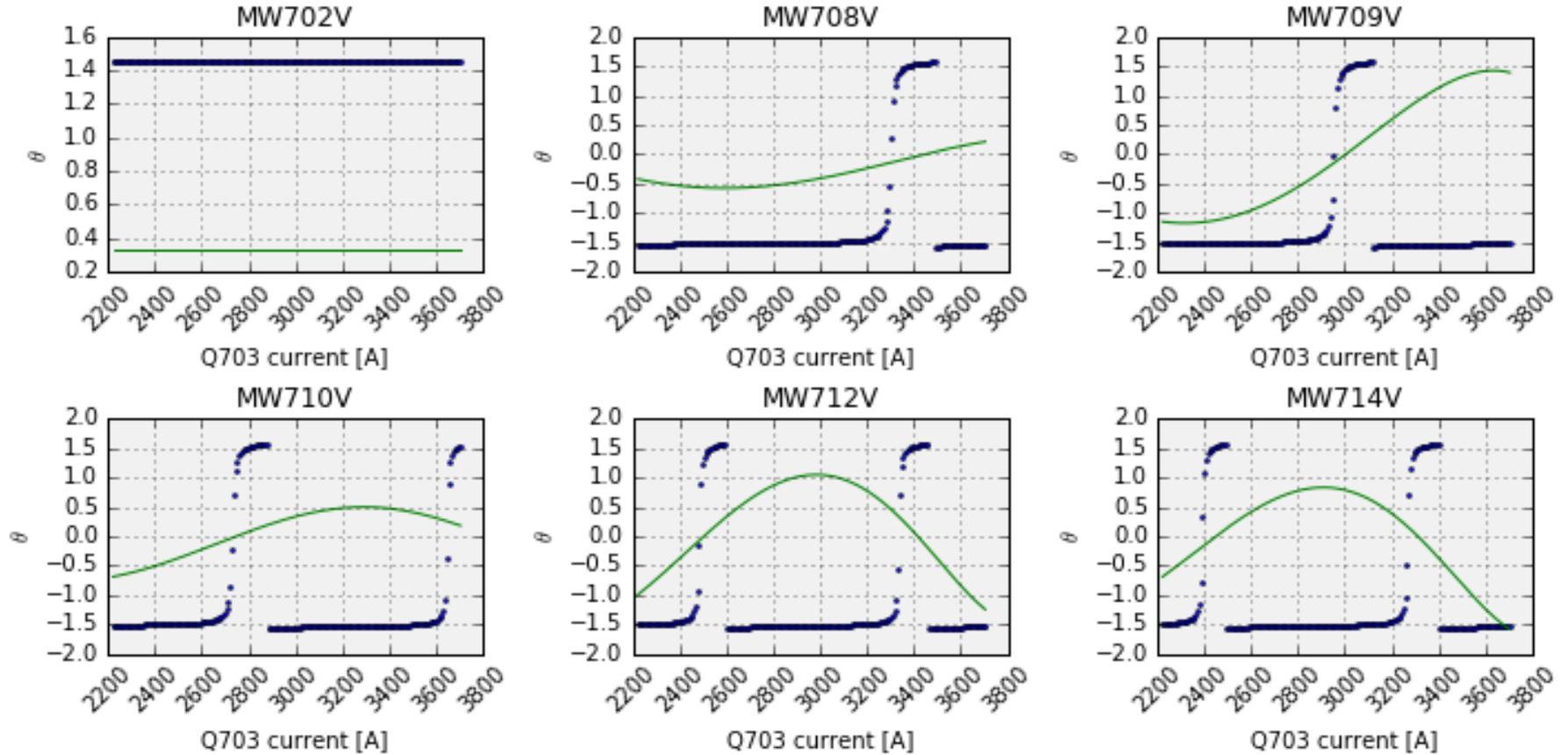
# P1 beamline

MAD simulations of the P1 beamline, while varying Q703 current, can show how the phase orientation angle (blue) changes with the dispersion (green). Ideally, we will choose a current range for the Q703 scan that provides wide phase angle sweep and minimal dispersion.



# P1 beamline

MAD simulations of the P1 beamline, while varying Q703 current, can show how the phase orientation angle (blue) changes with the dispersion (green). Ideally, we will choose a current range for the Q703 scan that provides wide phase angle sweep and minimal dispersion.

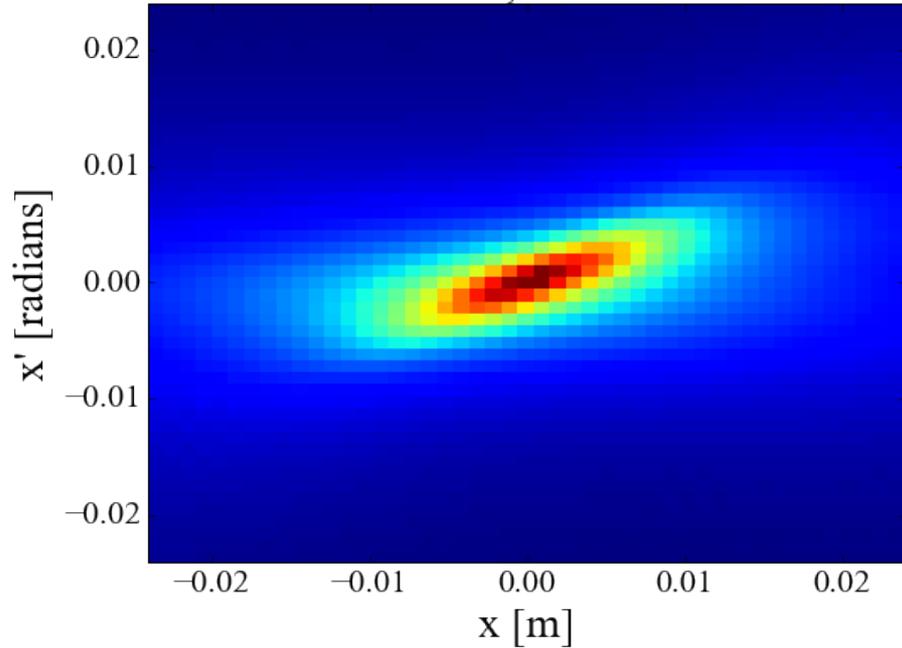


# Backup Slides

# FBP with no filter vs. Shepp-Logan for ellipse

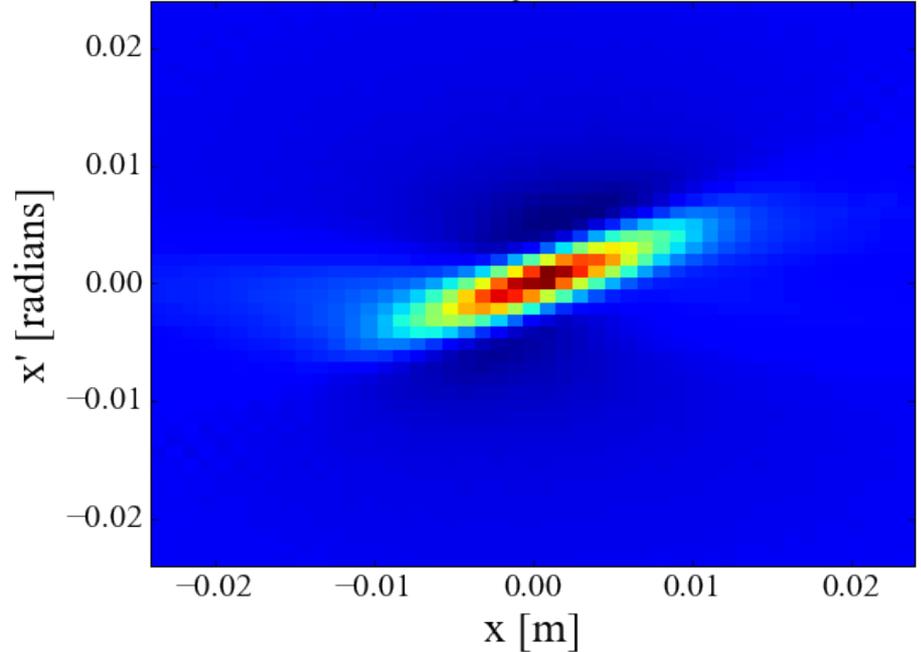
No filter

FBP reconstruction, RMS error = 0.0955



Shepp-Logan filter

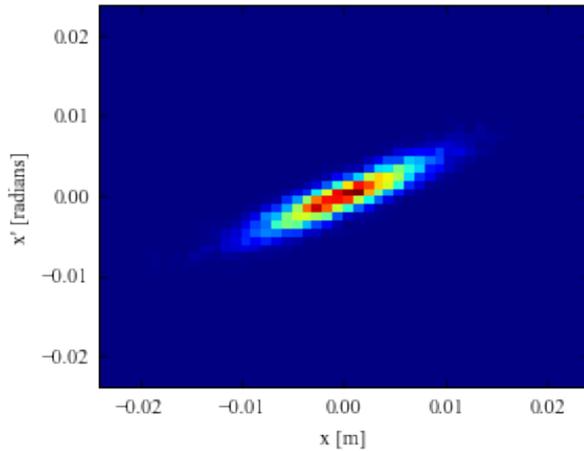
FBP reconstruction, RMS error = 0.0621



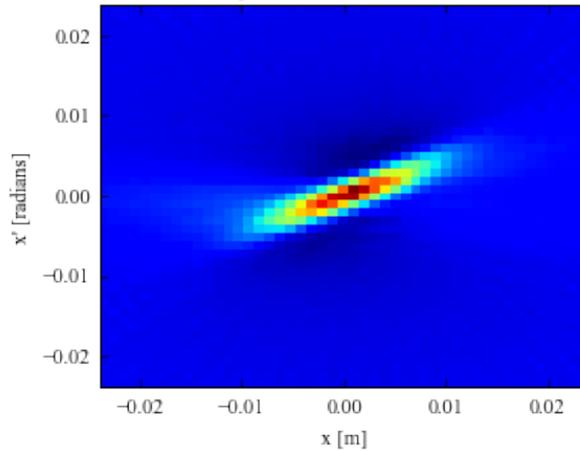
# Elliptical beam reconstruction

## FBP reconstructions

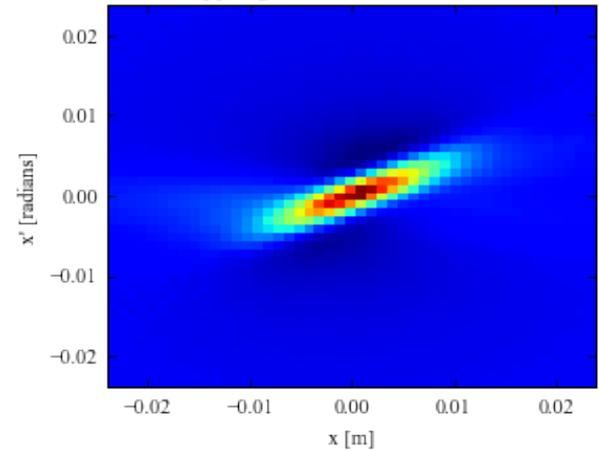
Initial Beam



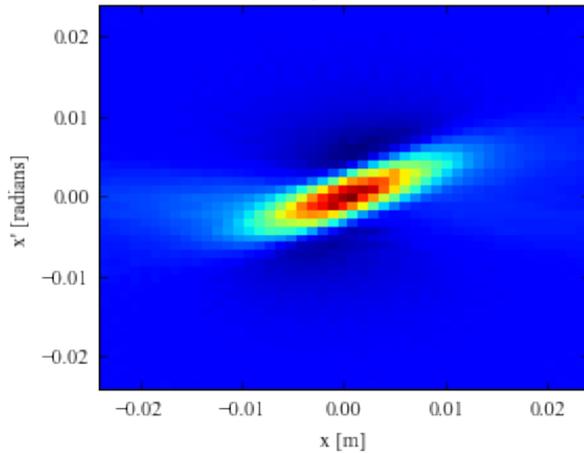
ramp filter, error = 0.068639



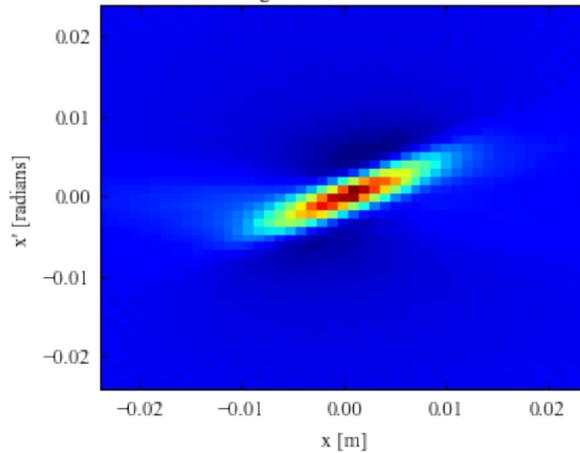
shepp-logan filter, error = 0.062114



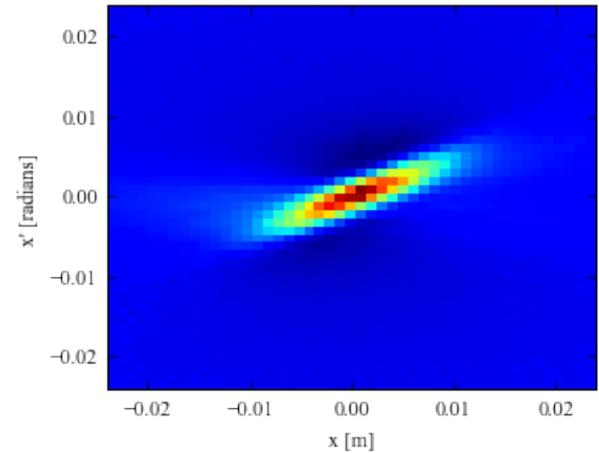
cosine filter, error = 0.059382



hamming filter, error = 0.065954

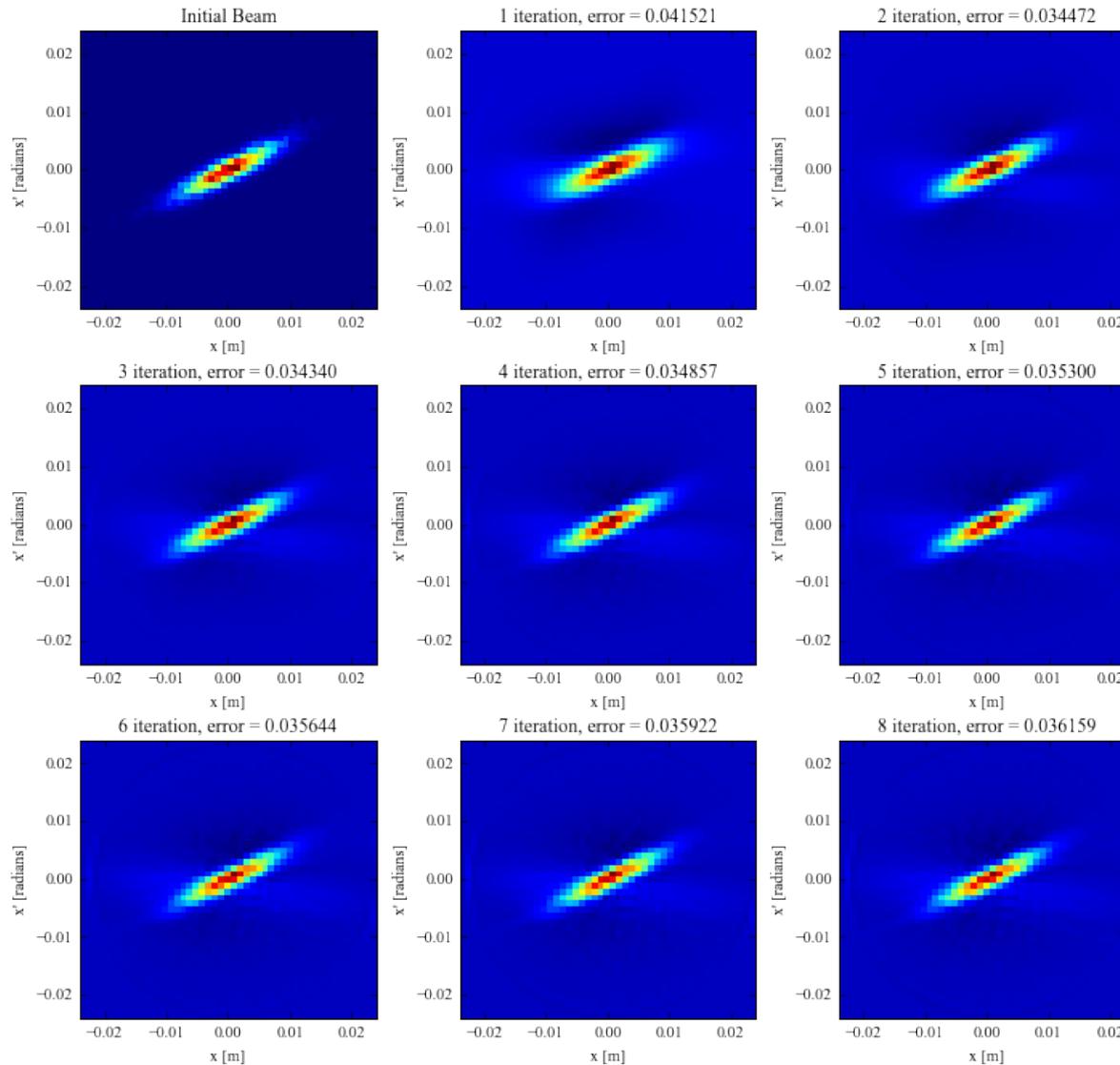


hann filter, error = 0.065740



# Elliptical beam reconstruction

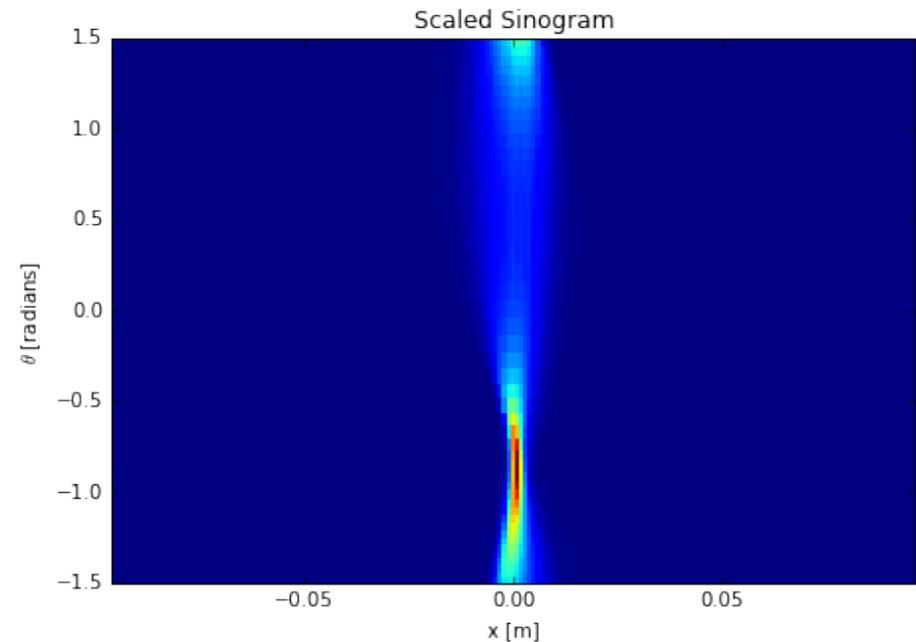
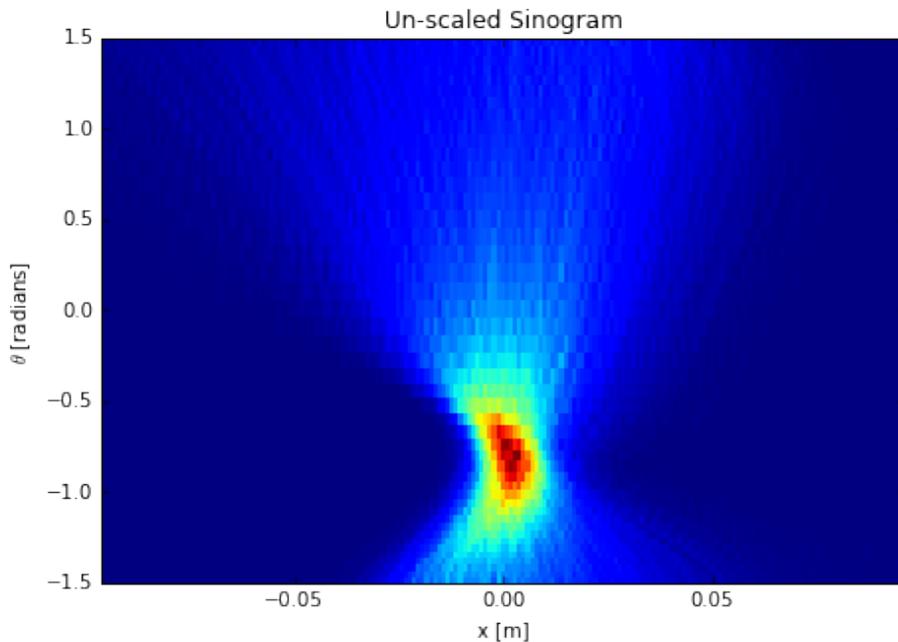
SART reconstructions, relax = 0.1



# Bigger multiwire

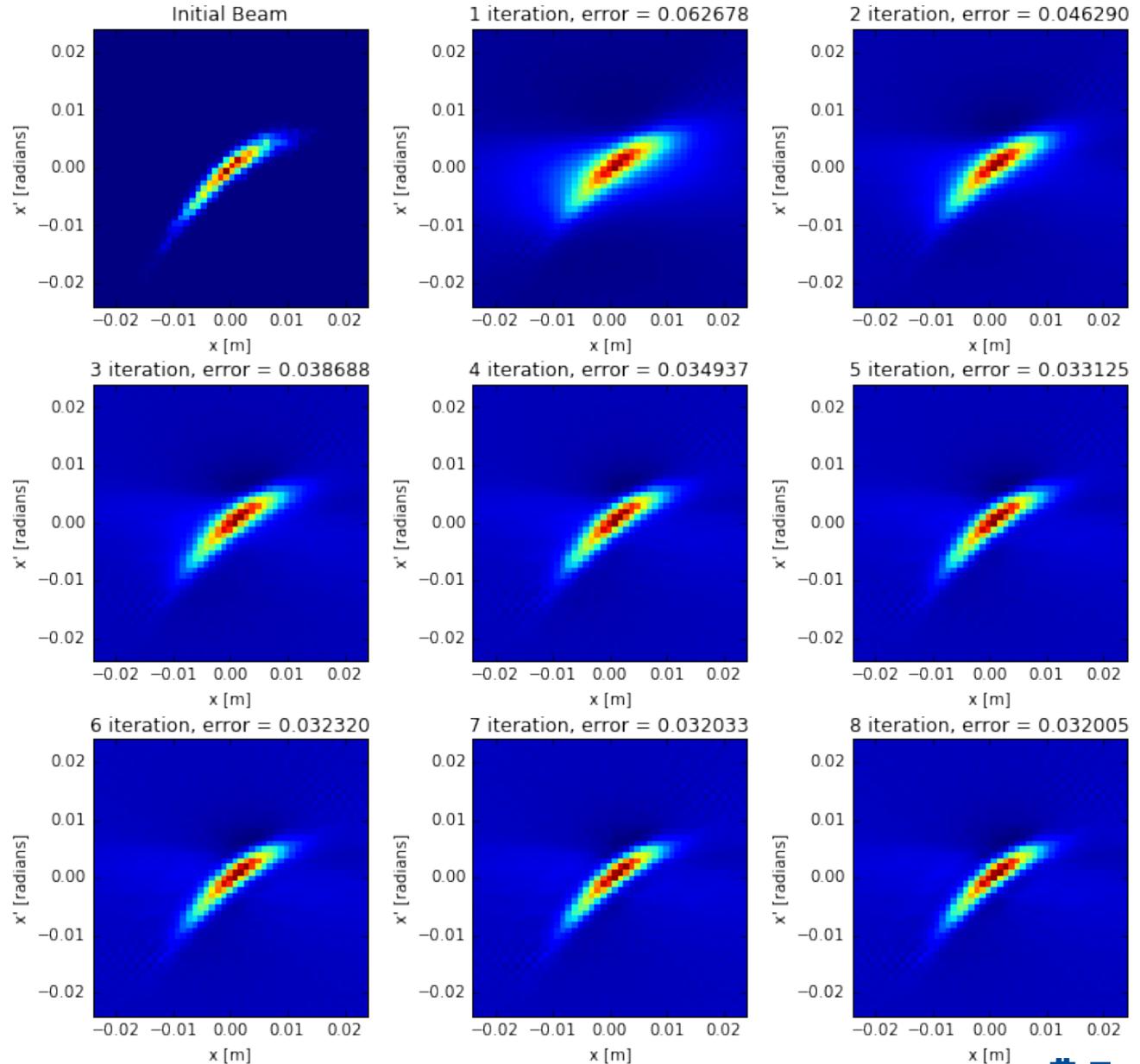
Outlying particles are reconstructed well by this method. One potential explanation is the aforementioned “scraping” of the beam tails due to the finite size of the multiwire detector. As we vary the quadrupoles, the beam may become larger than the multiwire, and information is lost.

To investigate, we redo the scan with the same initial distribution, but increase the size of the multiwire detector by a factor of 4 to eliminate any visible clipping in the sinogram.

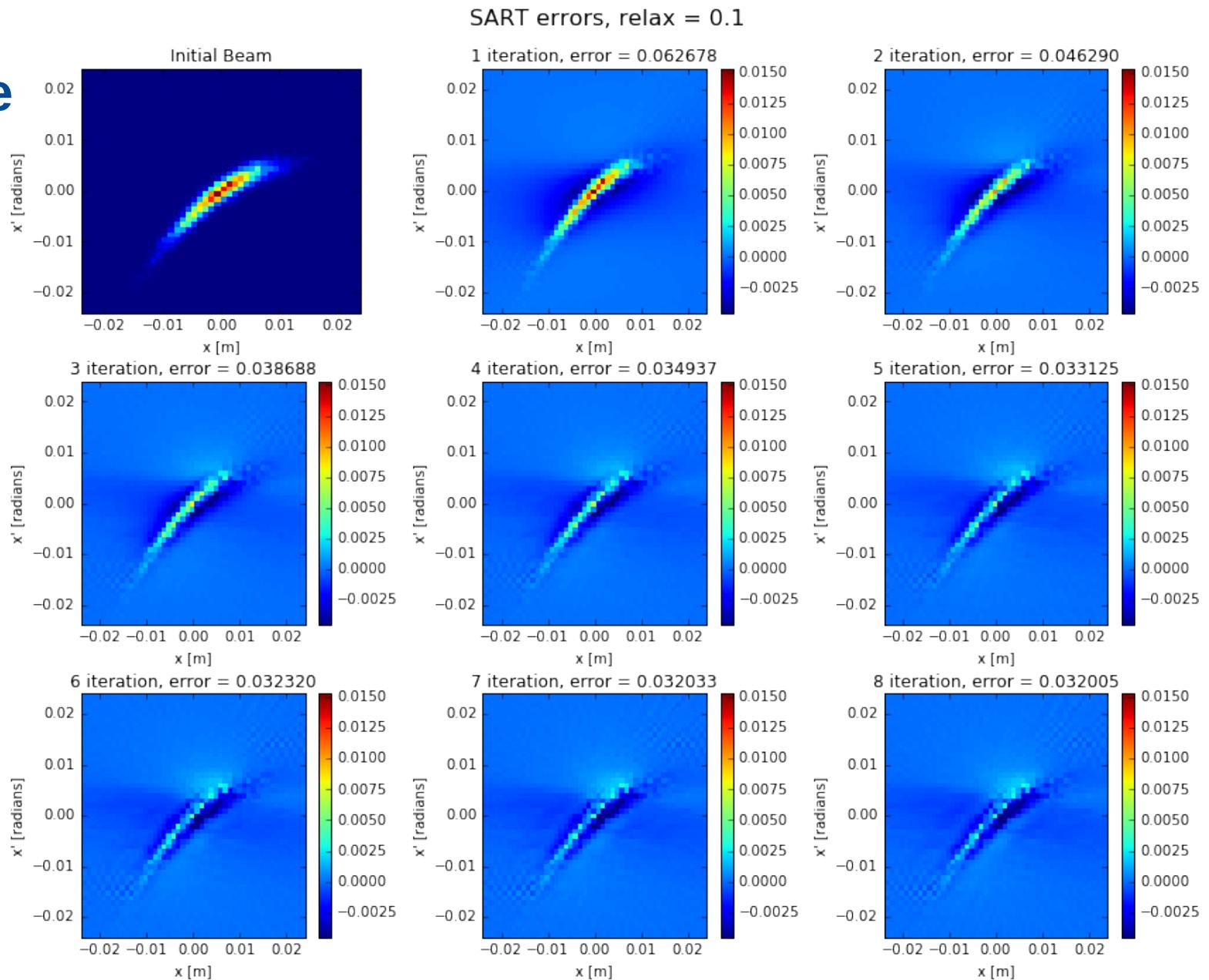


# Bigger multiwire SART

SART reconstructions, relax = 0.1



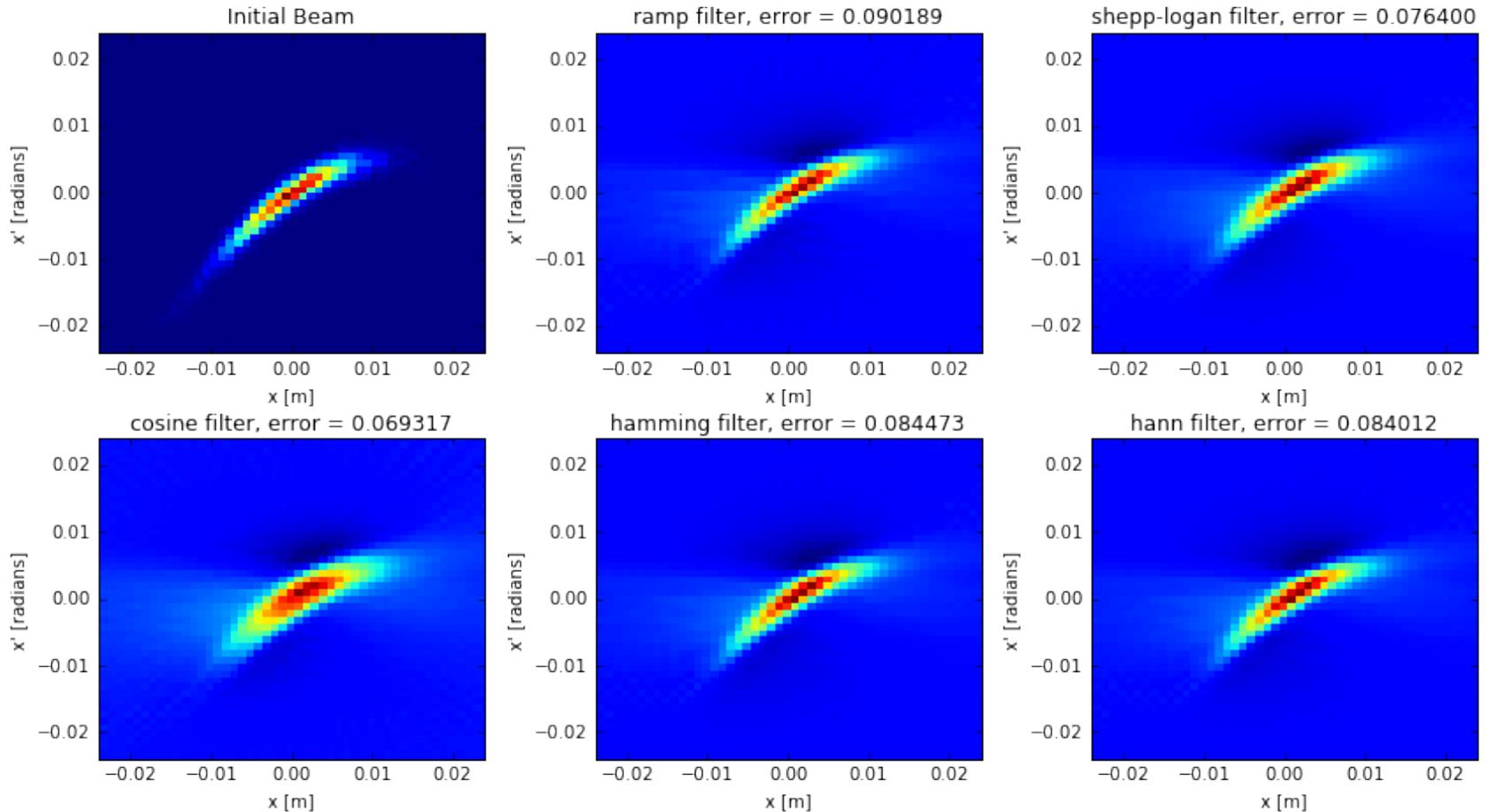
# Bigger multiwire SART errors



# Bigger multiwire FBP reconstruction

Tail reconstruction is not noticeably improved for the larger multiwire detector.

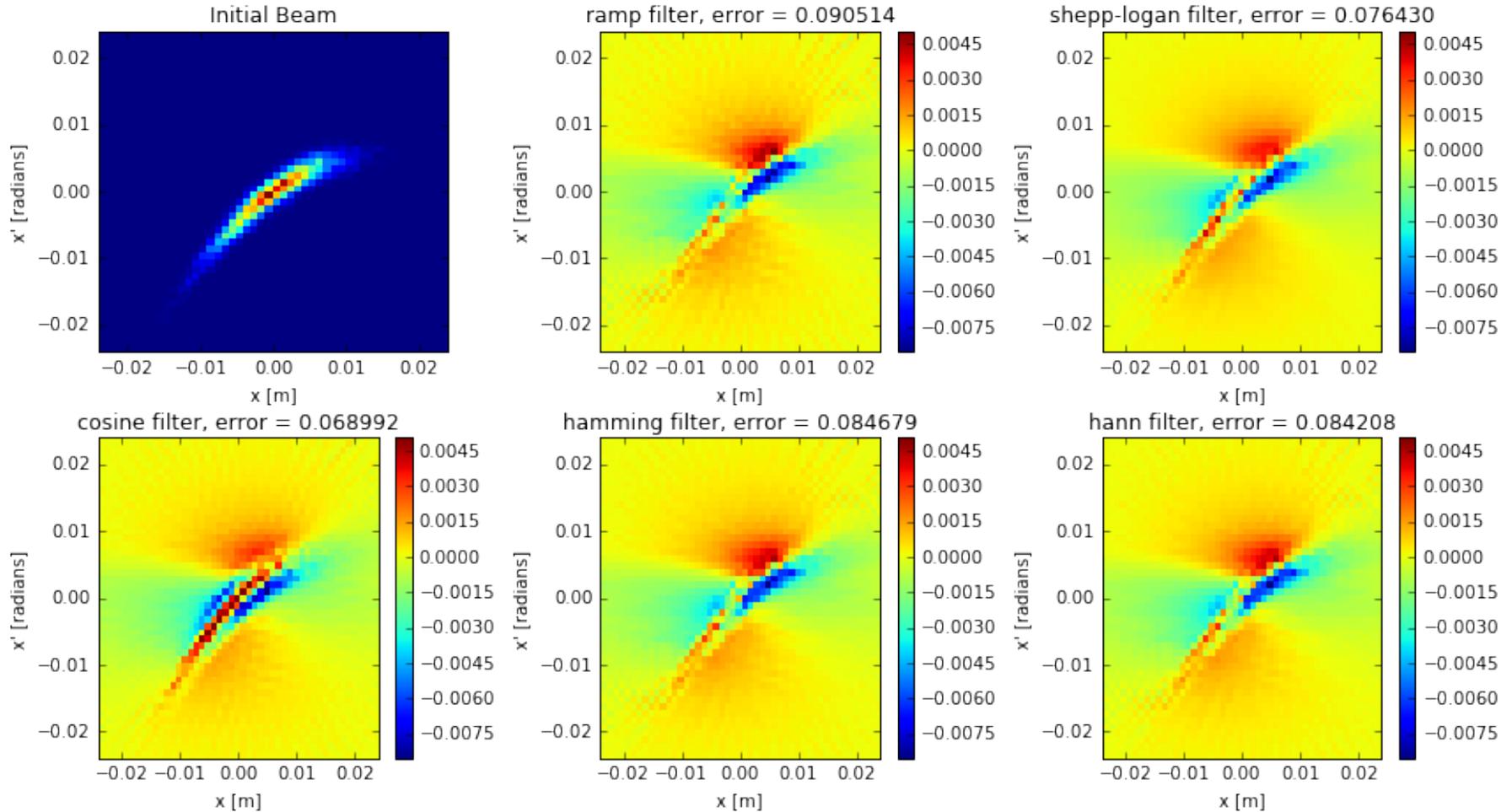
## FBP reconstructions



# Bigger multiwire FBP reconstruction

Tail reconstruction is not noticeably improved for the larger multiwire detector.

## FBP errors



# Bigger multiwire high-resolution FBP

SART reconstructions, relax = 0.1

