EE369C: Assignment 3 Solutions

1. We first need to compute the kernel. The optimum β for the Kaiser-Bessel kernel is 11.44, using the expressions from the Beatty paper. The result is plotted in Fig. 1, which was given in the assignment.

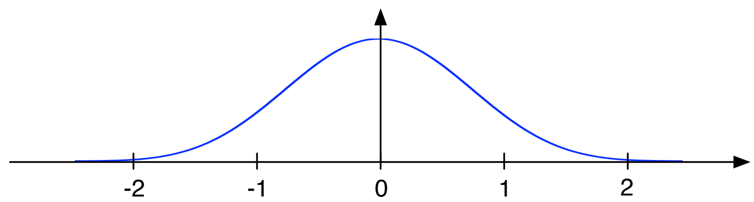

Figure 1: Optimum Kaiser-Bessel Kernel for a 2X grid, with a kernel width of 5 samples on the 2X grid.

I used nearest neighbor interpolation for the kernel, with an oversampling factor of 455. This results in a reasonable length array to look the kernel samples up from, and a simple implementation. A kernel oversampling factor of only 10 would be sufficient for linear interpolation, but the implementation would be more complex. A final alternative is simply to compute the kernel for each sample. This would be slower.

The only difficulty in the implementation is to make sure that the loops over the kernel samples has the proper ranges. It should go from $+/-$ half the kernel width on the 2X grid.

The reconstruction for the 2X FOV, and the central 1X FOV shown in Figs. 2. The cross section through the image is shown Fig.‘4, showing that the apodization has been accurately corrected.

2. In order to obtain a maximum reconstruction error of 10^{-3}, we need a kernel width of 6 samples. This is from the plot in Fig. 3 of the Beatty paper. The optimum β can then be computed as 10.995. The kernel oversampling factor for nearest neighbor interpolation, and 10^{-3} maximum

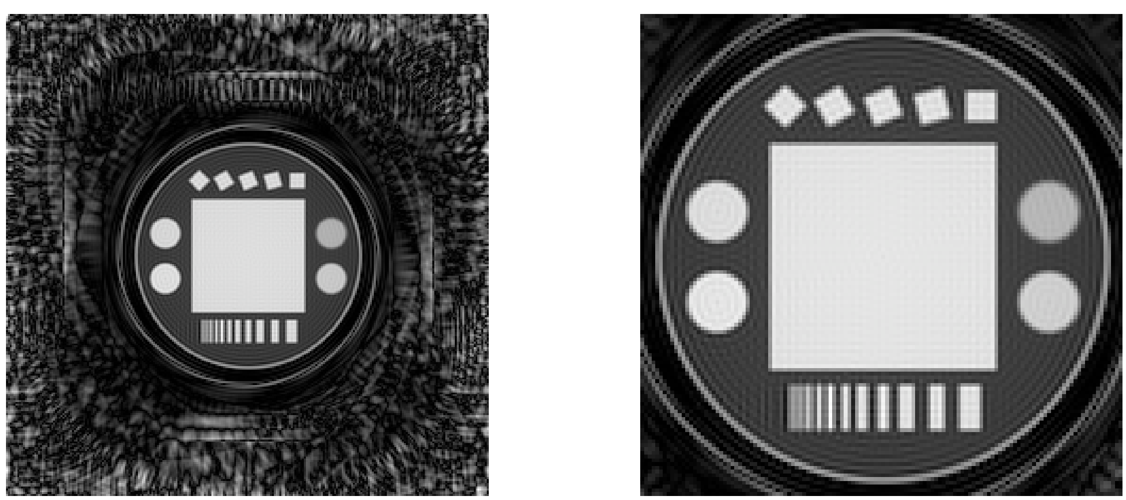

Figure 2: Reconstruction using the optimum Kaiser-Bessel kernel on a 2x grid, using a 5 sample kernel (on the 2X grid)

.

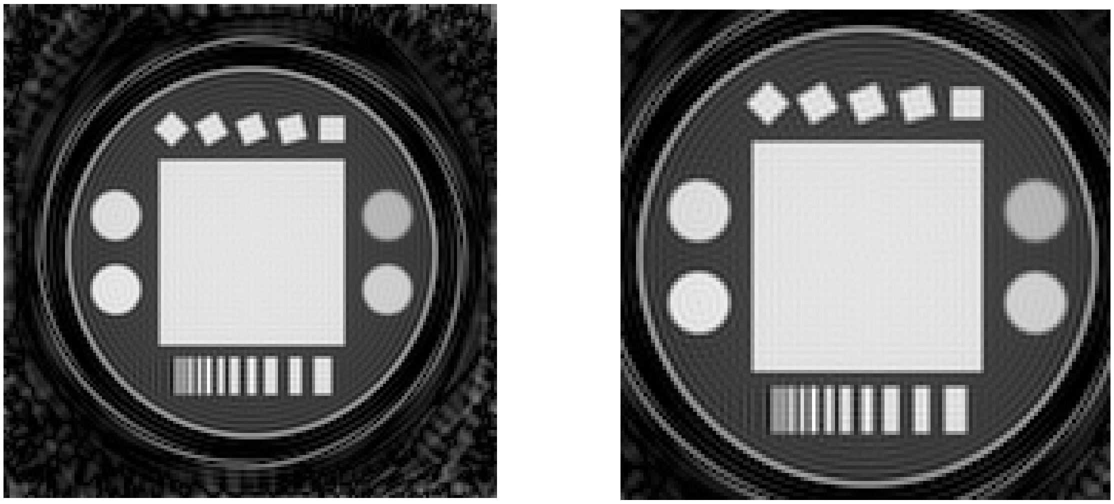

Figure 3: Reconstruction using the optimum Kaiser-Bessel kernel on a 1.25x grid, using a 6 sample kernel
.

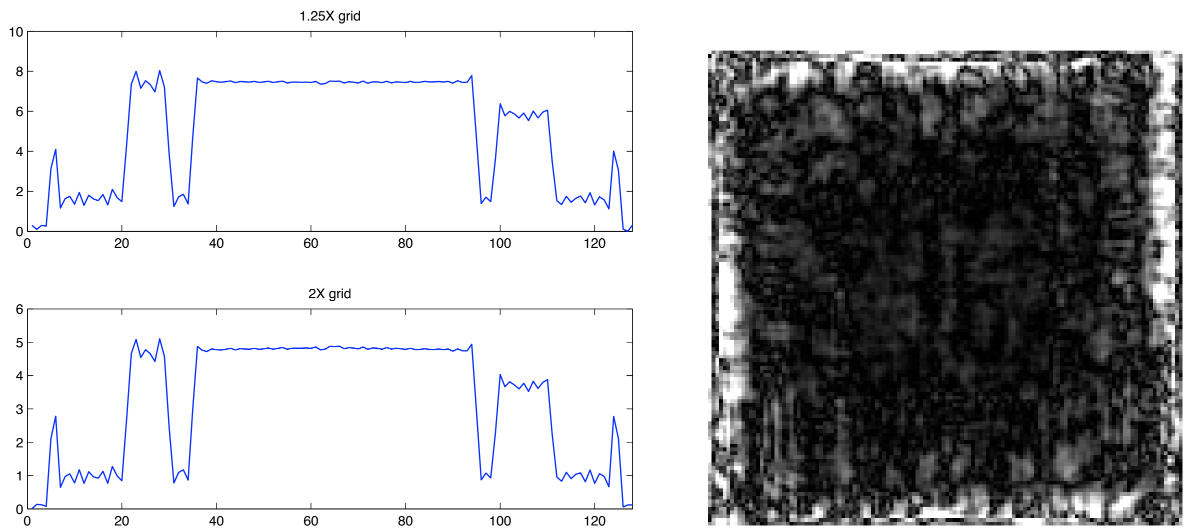

Figure 4: Cross sections through the central 1X FOV of the 2X and 1.25X reconstructions. The two plots are essentially identical. A difference image multiplied by 5000 is shown on the right.

error and 1.25X oversampling is 728. This still results in a reasonable length array, so I didn't go to linear interpolation, which would reduce the kernel oversampling to 16.

The reconstruction of the full 1.25X FOV and the central 1X FOV are shown in Fig. 3. A cross section is plotted in Fig. 4 showing the apodization correction is working properly, and that the profile is essentially identical to that of the 2X reconstruction. A difference image multiplied by 5000 is also shown. A constant scale factor of 0.66624 has been applied to the 1.25X reconstruction to correct for the different kernel areas.

3. The reconstruction using a post-compensation density correction is shown in Fig. 5, left. The reconstruction using pre-compensation by the Voronoi density estimate is also shown, center. Cross-section plots, right, show that the Voronoi density estimate is more accurate.

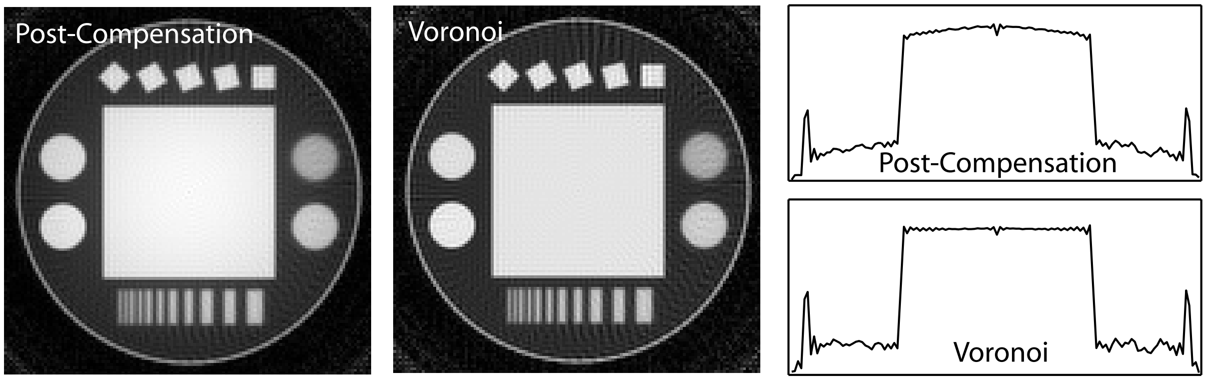

Figure 5: Reconstruction using post-compensation density estimate, left, and precompensation using the Voronoi density, center. Cross section plots show the Voronoi density is more accurate.

The key issue with computing the Voronoi density is what to do with the values at the edges of k-space, where the areas become large or infinite. The infinite values must be suppressed to get any reconstruction at all. There are also be unreasonably large values. If these are not suppressed, the result is very large structured noise. The question is, what value to assign to these edge samples in k-space. One approach is to enclose the trajectory in a circle. Another approach is to zero out the samples that are infinite or large, but this wastes hard-earned data, and reduces resolution. A third approach is to artificially extend the trajectory. A fourth approach, taken here, is to limit the Voronoi areas based on knowledge of the trajectory.

For the `rt_spiral.mat` data the trajectory of question 1, the trajectory has reached the constant velocity regime. The samples are then spaced by $1/FOV$ along the trajectory, and between spirals. We can then limit the Voronoi area to $(1/128)^2$, since k-space is scaled to go from $\pm$ 0.5 normalized cycles/cm. For the `var_dens.mat` data of this question, the gradient amplitude, and hence the k-space velocity, is still increasing at the end of the trajectory, so the samples are closer than $1/FOV$ along the trajectory, but still $1/FOV$ between interleaves. What I've done here is simply use the density for the last fully enclosed sample, and carried that forward to the end of the waveform.