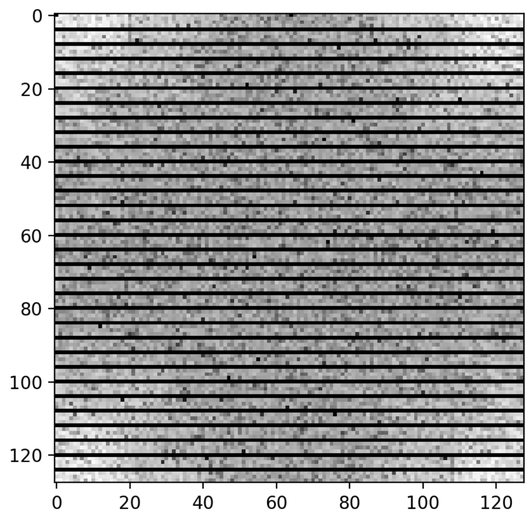
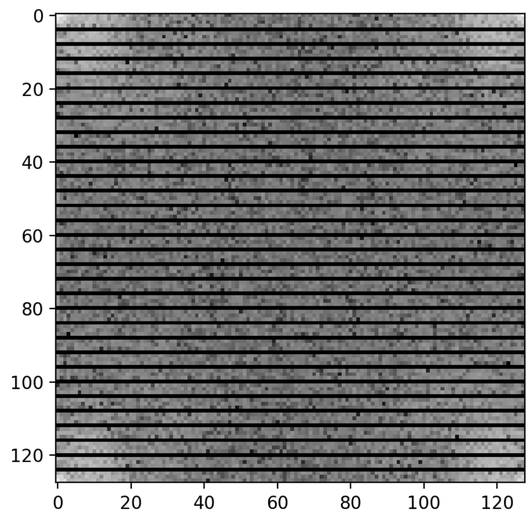
Data: real part (re) and imag part (im) of image's k-space received by the MRI.
Black spots are where the signal is lost.

re:

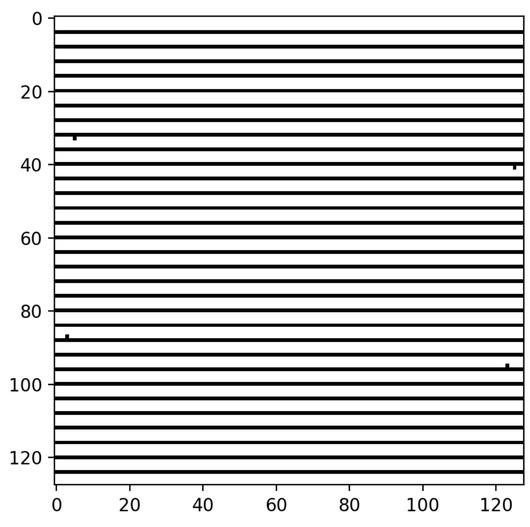


im:



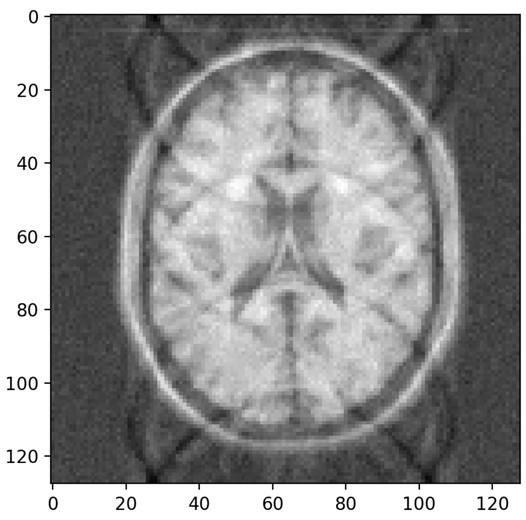
Apply a threshold: signal = abs(re) > 0.5 to get a matrix of where the signal is received.
signal[i][j] = 1 if there is signal in this spot, 0 otherwise

signal:



Naively reconstruct the image by taking inverse FFT, we get the naive image.

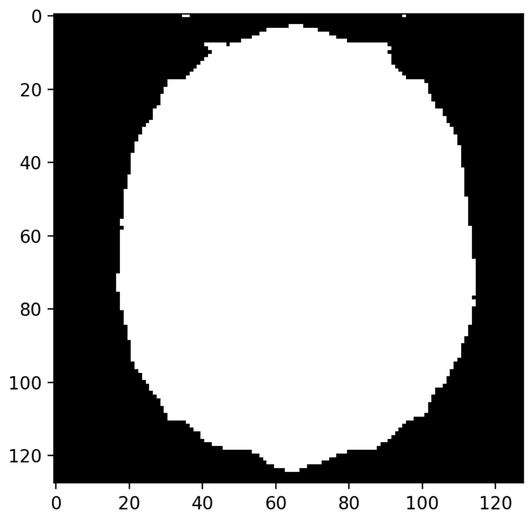
naive image



median filter
threshold

median = normalize(ndimage.median_filter(img, 10))
head = median > 0.1

head



Determine the box constraint

x_lb and x_ub
- Inside head: pixel >= 0
- Outside head: -4 <= pixel <= 4

Optimization problem

```
variables:
  x[128][128] = 0


constants:
  im[128][128] = Dataset("im.h5", "im")
  re[128][128] = Dataset("re.h5", "re")
  signal[128][128] = Dataset("signal.h5", "signal")
  xLowerBound[128][128] = Dataset("x_lb.h5", "x_lb")
  xUpperBound[128][128] = Dataset("x_ub.h5", "x_ub")

constraints:
  x ⩾ xLowerBound, x ⩽ xUpperBound

let:
  smootherX = rotate (0, 1) x + rotate (0, -1) x - 2 *. x
  smootherY = rotate (1, 0) x + rotate (-1, 0) x - 2 *. x
  regularization = norm2square smootherX + norm2square smootherY

minimize:
  norm2square ((signal +: 0) * (ft x - (re +: im))) + 3000 *. regularization
```

Run (read Readme.md)

Result: