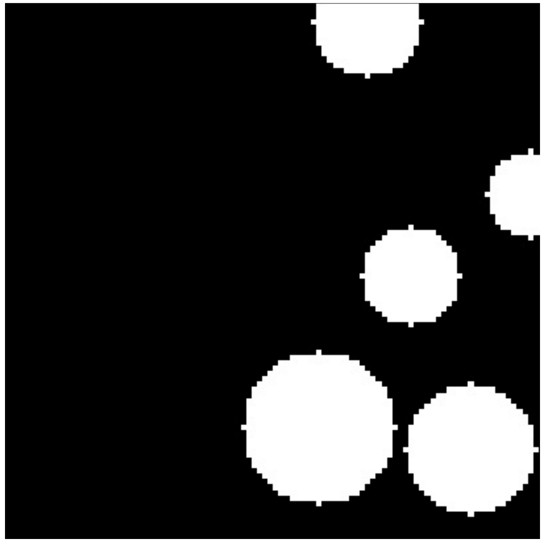
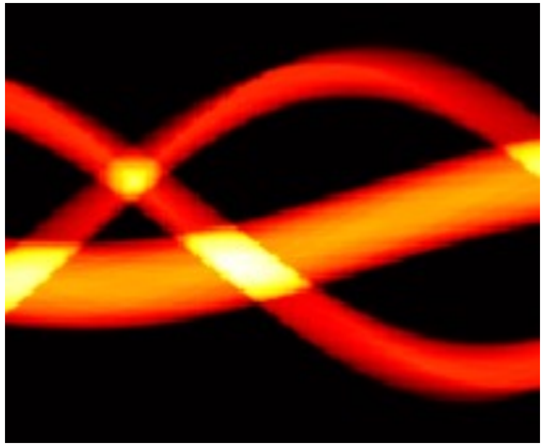
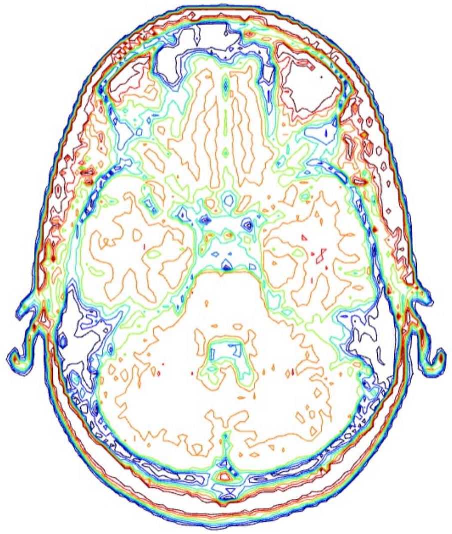
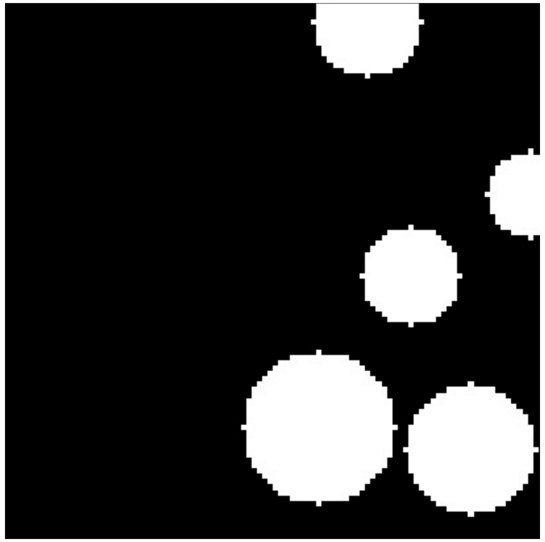


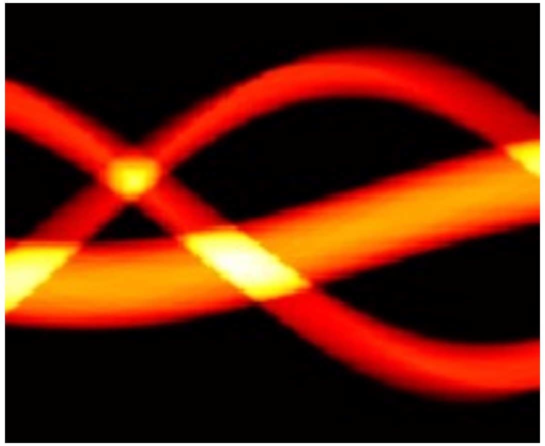



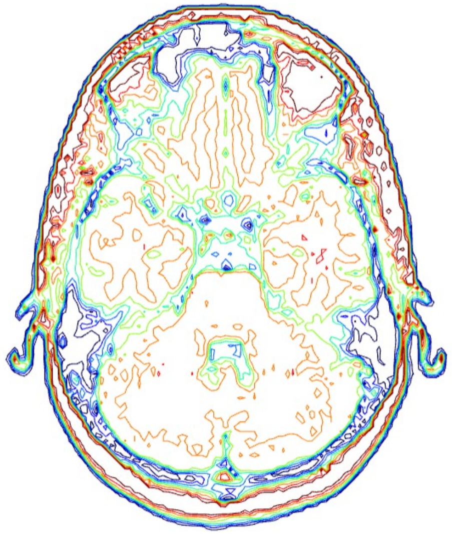
Radon Transform, Fourier Slice Theorem, Volume Visualization

Physics 305 Activity 6
2nd Sem AY 2022-2023

Mark Jeremy G. Narag
2014-64423
PhD Physics







All the codes and files in this activity are available on my Github:

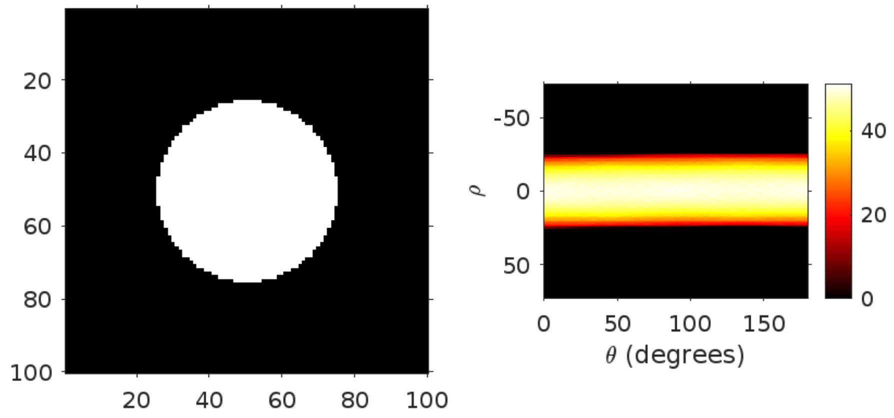
https://github.com/mgnarag/physics305_computational_imaging

Radon Transform

Radon transform is a mathematical technique used in image processing and medical imaging to analyze the internal structures of an object or region. Similar to Fourier transform, it converts an image to a different domain – in this case, the radon domain:

$$g(\rho,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - \rho)dxdy$$

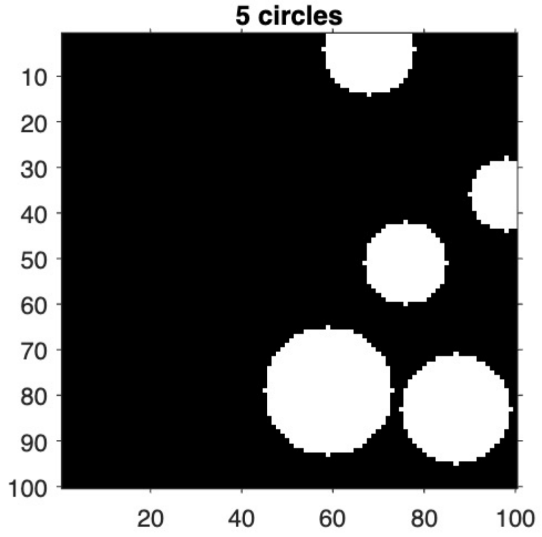
This domain represents the image as a set of line integrals which are projections of the object from various angles. So basically imagine shining a light to an object in different angle. This is why the sinogram* of the radon transform of a circle is rectangle

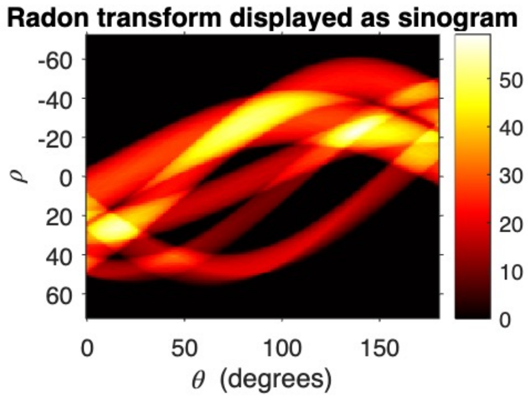


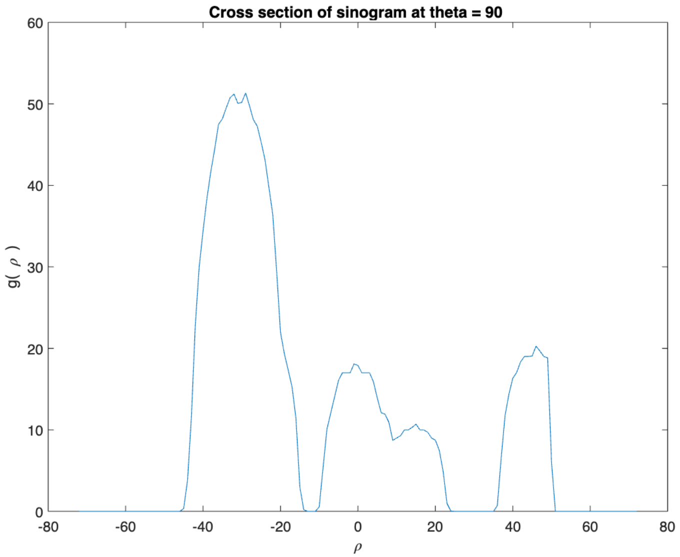
**sinogram*
– 2D representation of a cross-sectional slice of an object
– x-axis : projection angle
– y-axis : detector position or distance from the center of rotation

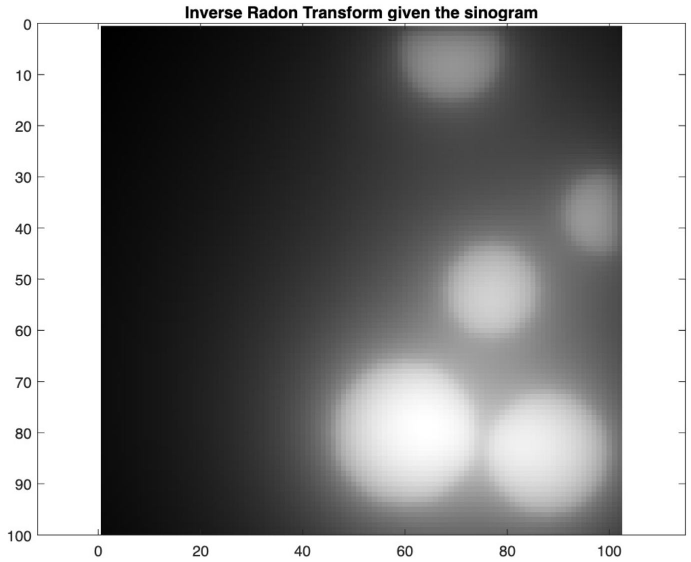
Radon Transform

1. Create an image of several circles with different sizes.
2. Take its Radon transform and recover the image using the inverse Radon transform.





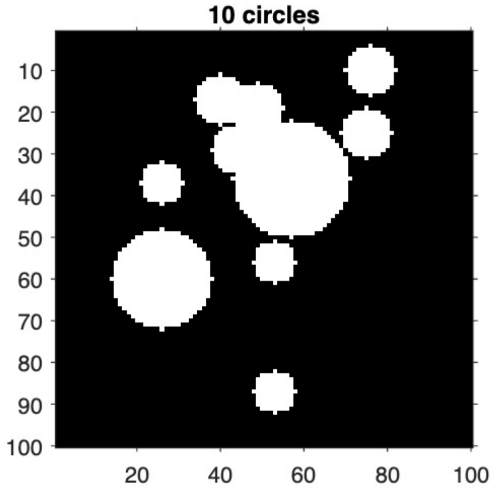


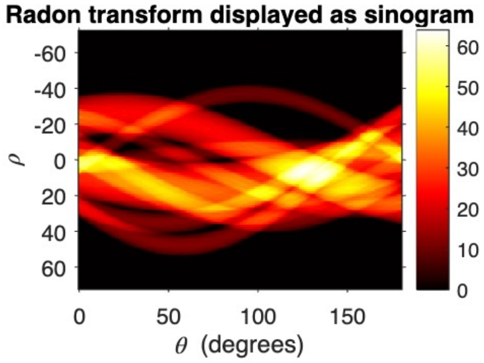


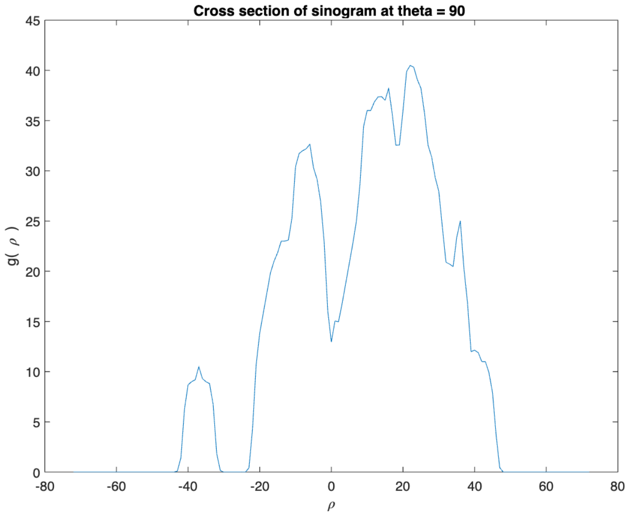
Here is an example of 5 random circles and its corresponding radon transform in sinogram, it's cross section, then the recovered image from inverse radon transform. We can see that the recovered image is blurry. This is because of the absence of filter in our inverse radon transform. We will resolve this later.

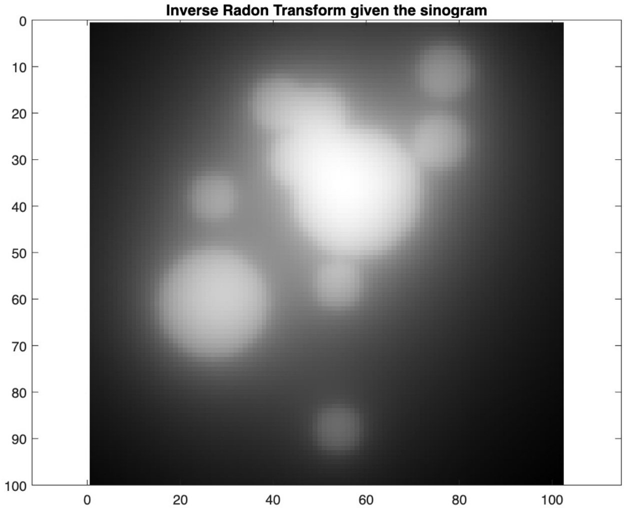
Radon Transform

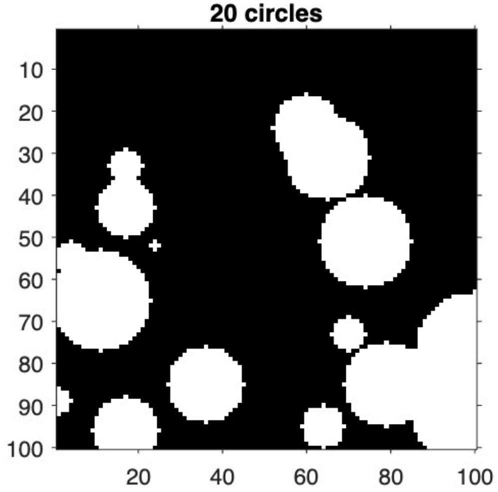
Here are more examples

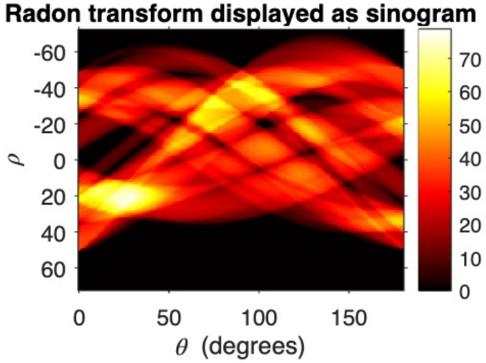


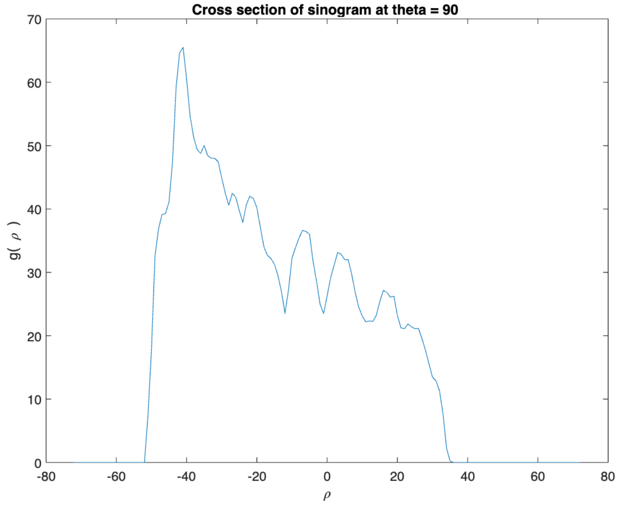


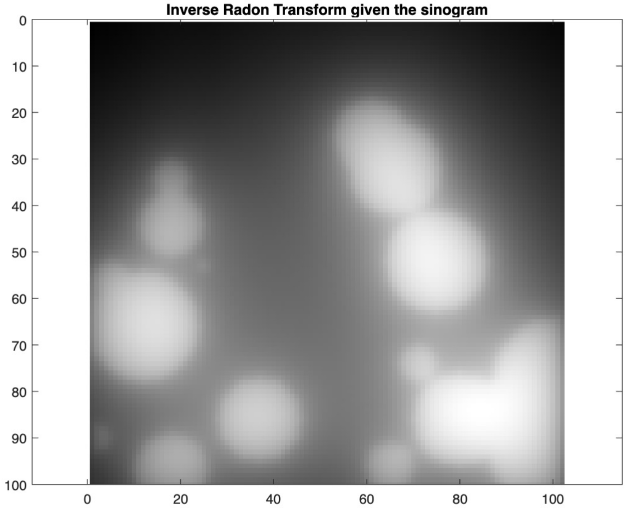






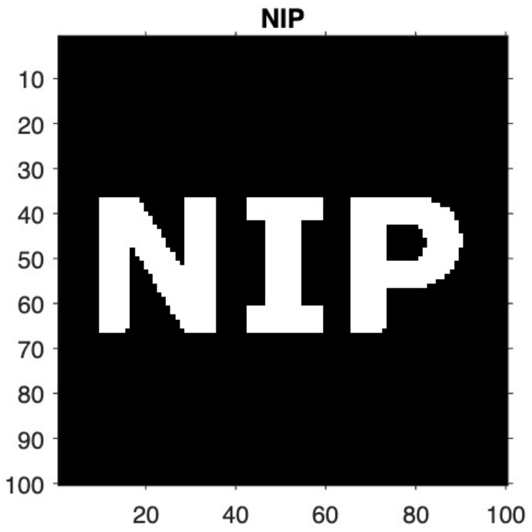


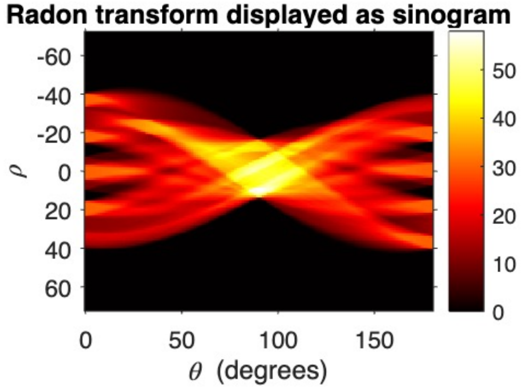


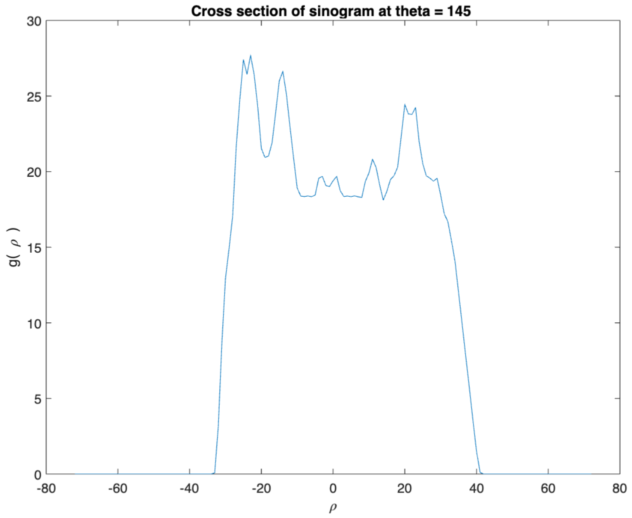


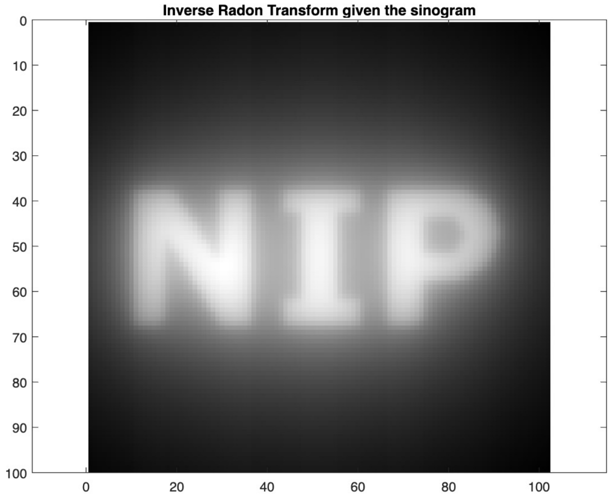
Radon Transform

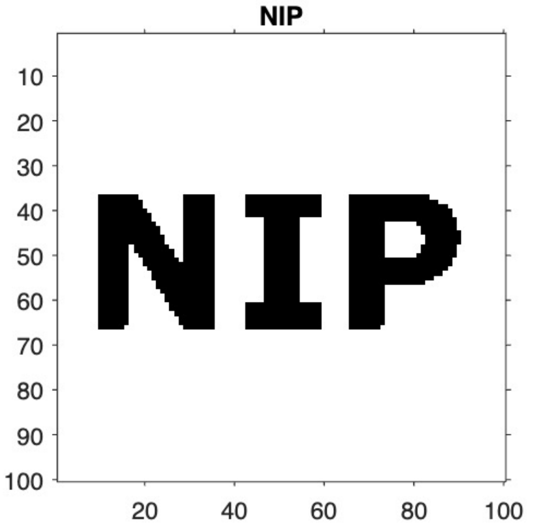
Let's try it on synthetic image. Let's say NIP letters both in black background and white background.

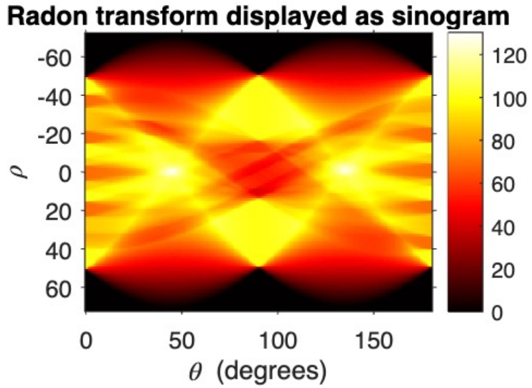


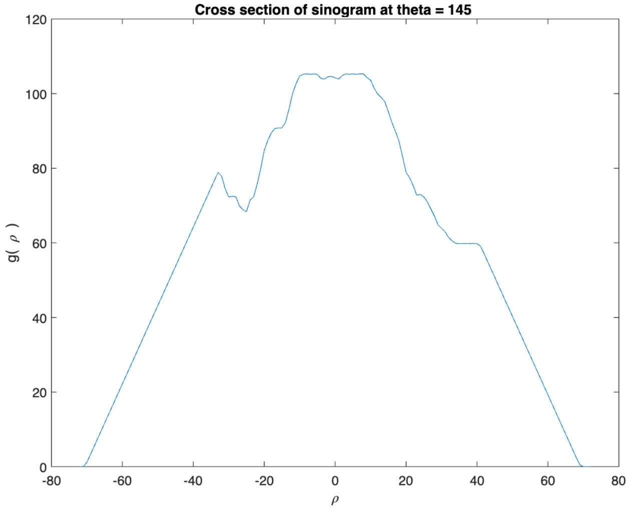


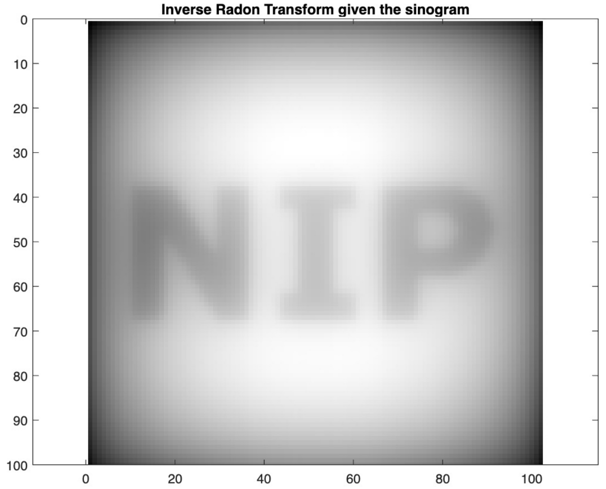








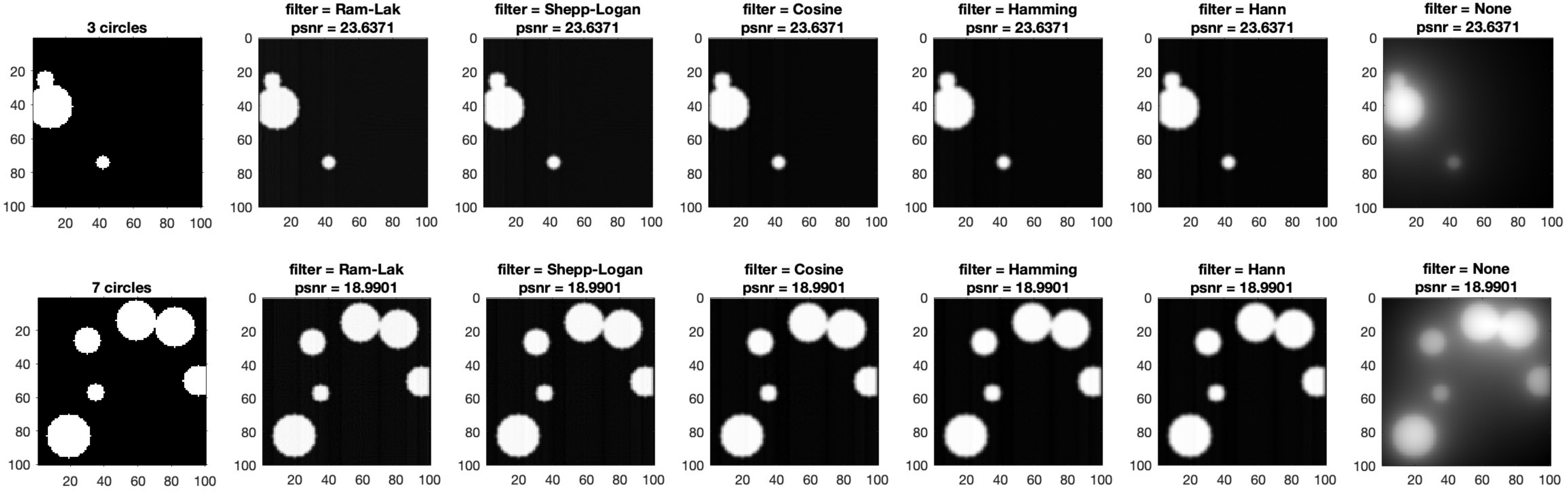




Fourier Slice Theorem

The filtered backprojection in the inverse Radon transform can be set by specifying a filter in the iradon argument.

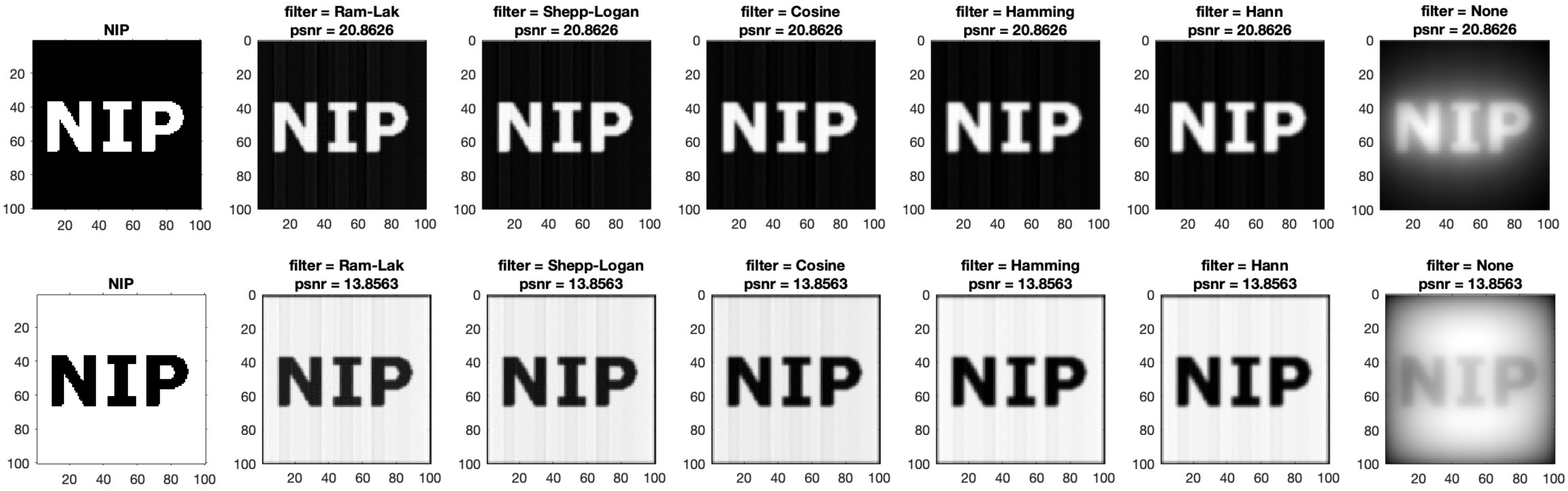
1. Get the inverse Radon transform of your synthetic images and compare the output for different filter functions.



We can see that the reconstruction is now better when we applied various filter functions. Moreover, PSNR score is actually the same for all filters.

Fourier Slice Theorem

Let’s try for the NIP letters. Here, we can see that the reconstruction from the white background produces artifacts – apparent vertical lines on the letters with vertical lines. The PSNR score is also low here.

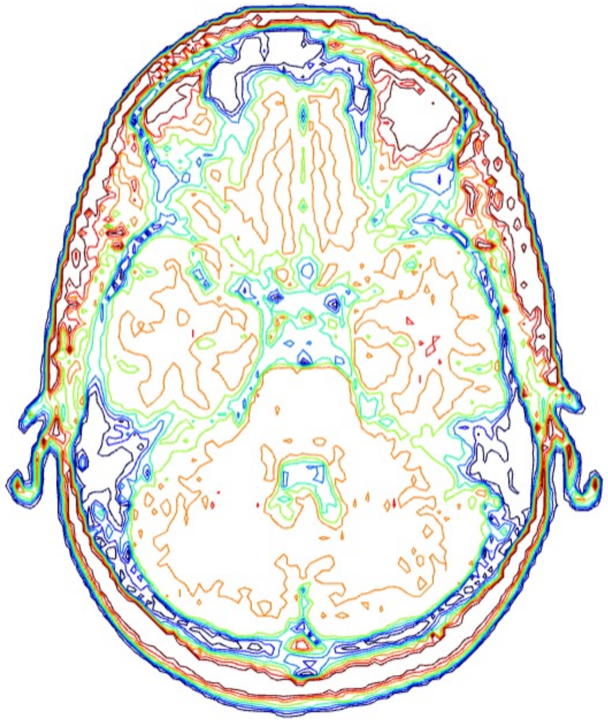


Volume Visualization

1. Follow the 3D visualization examples in the MATLAB link above. Show different views of the volume created.

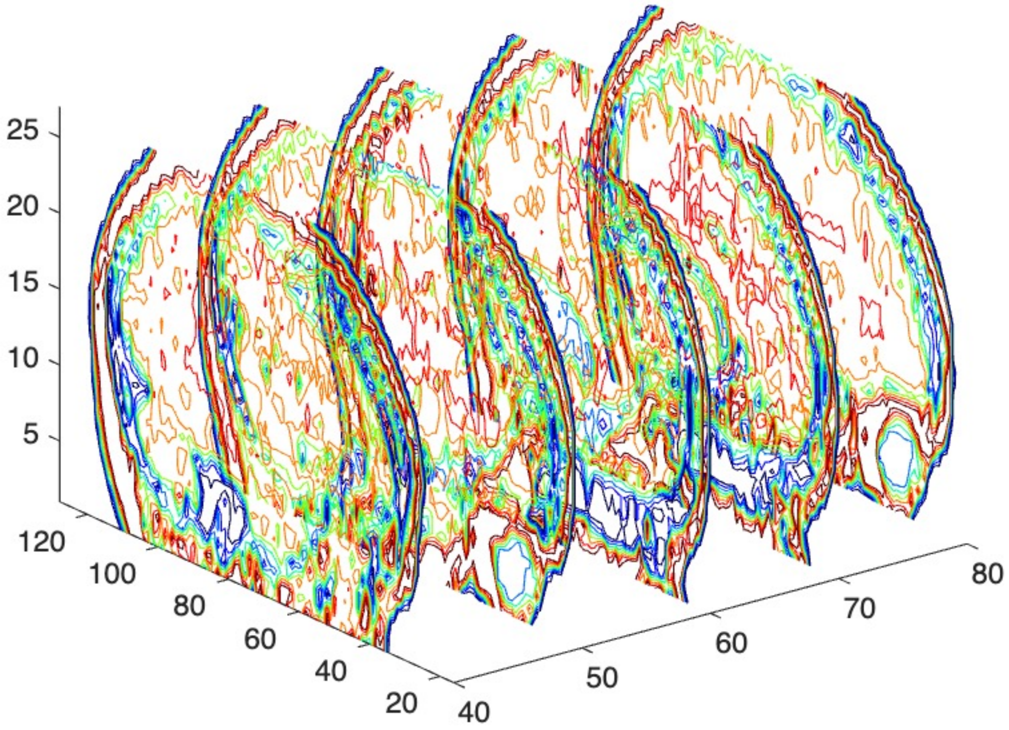
First, let's visualize the MRI image:

Then let's display the 2D Contour Slice

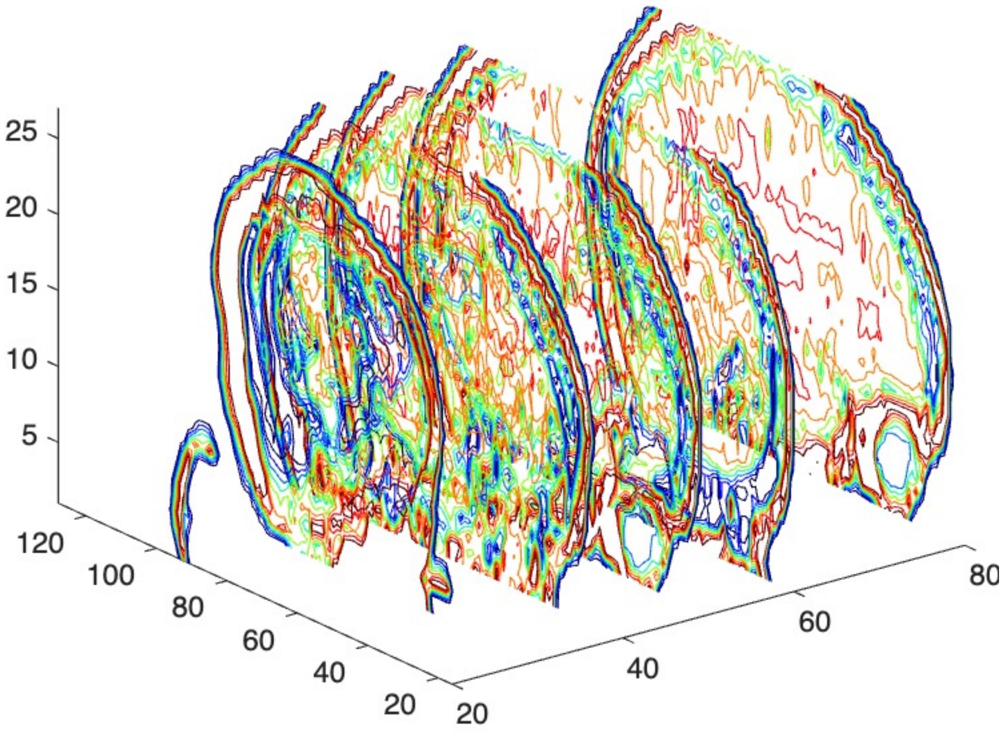


Volume Visualization

Next, let's view the volume created in different views by changing the parameters in contourslice(D,[Sx],[Sy],[Sx],8). Where Sx, Sy, and Sz are the slices in x,y, and z respectively. Let's check the slices in **x**



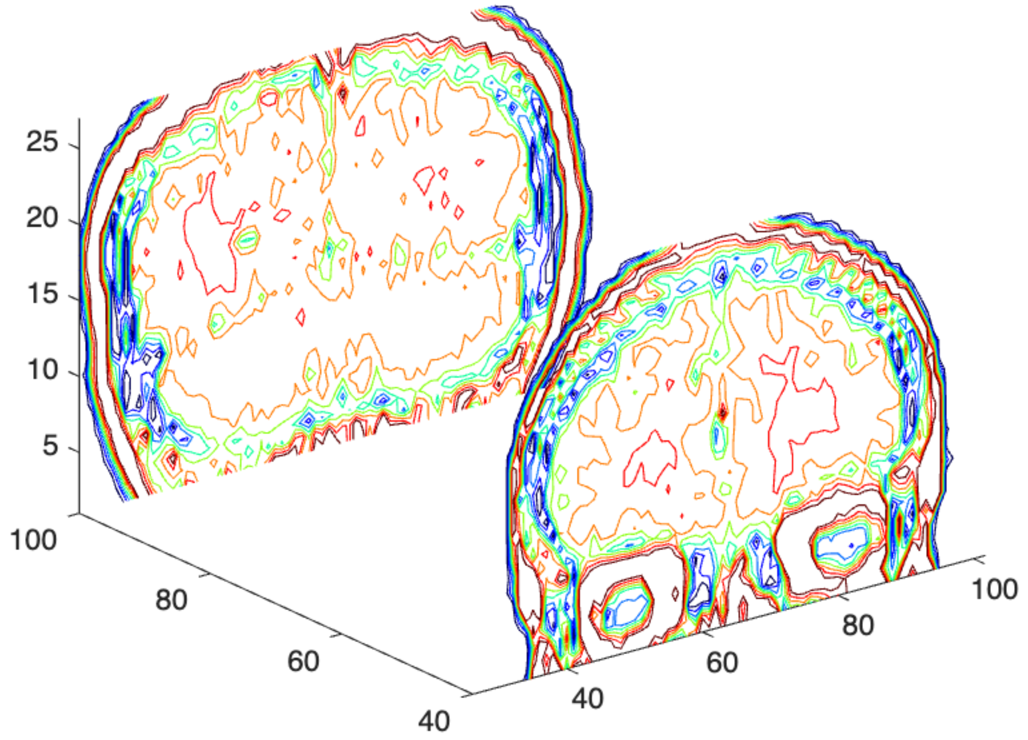

```
contourslice(D,[40 50 60 70 80],[],[],8);
```



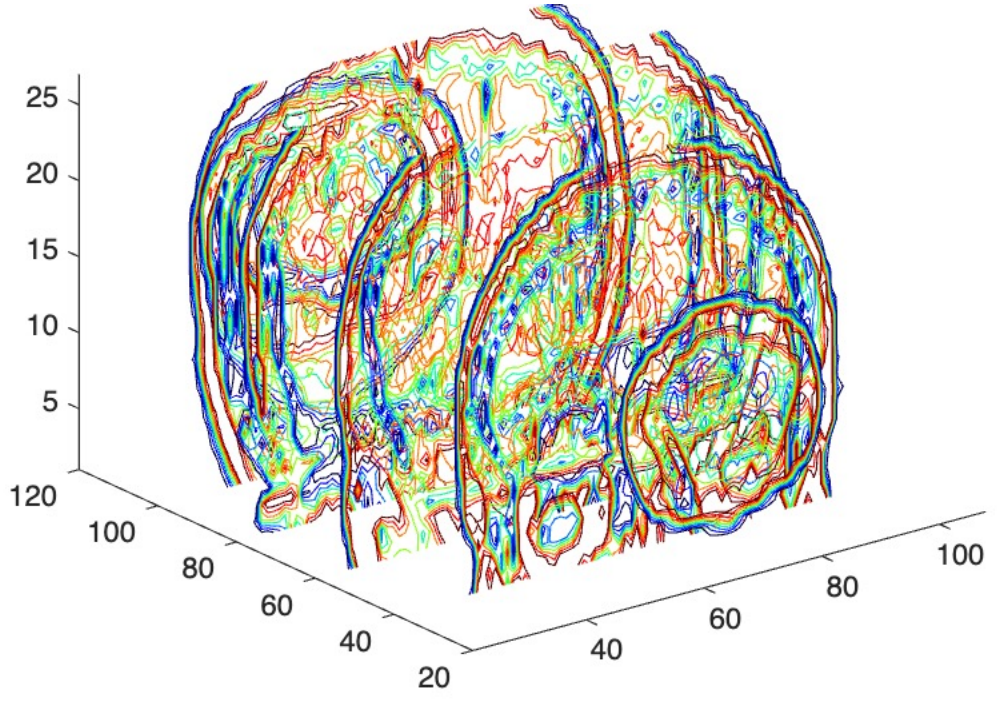

```
contourslice(D,[0 10 20 30 40 50 60 80],8);
```

Volume Visualization

Next, let's view the volume created in different views by changing the parameters in contourslice(D,[Sx],[Sy],[Sx],8). Where Sx, Sy, and Sz are the slices in x,y, and z respectively. Let's check the slices in **y**



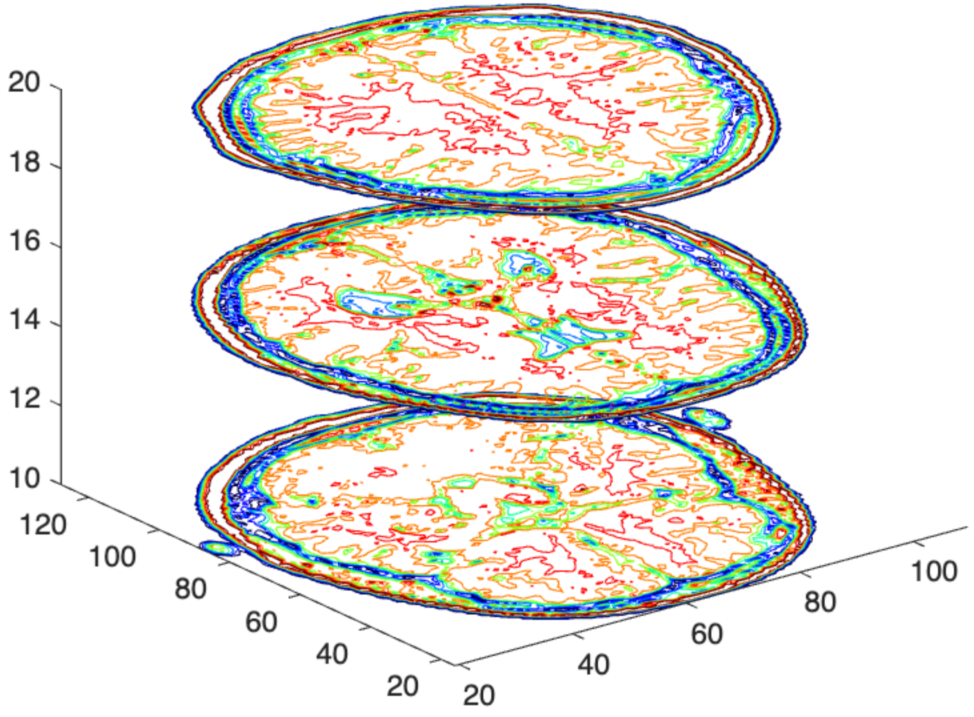
`contourslice(D,[],[40 100],[],8);`



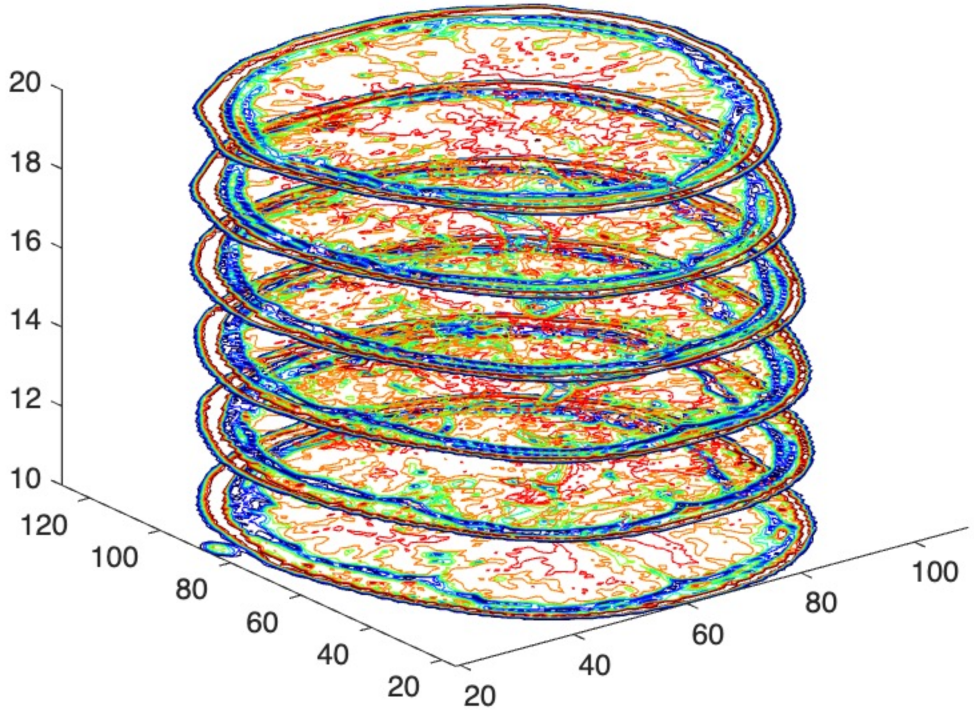
`contourslice(D,[],[20 40 60 80 100 120],[],8);`

Volume Visualization

Next, let's view the volume created in different views by changing the parameters in contourslice(D,[Sx],[Sy],[Sx],8). Where Sx, Sy, and Sz are the slices in x,y, and z respectively. Let's check the slices in **z**



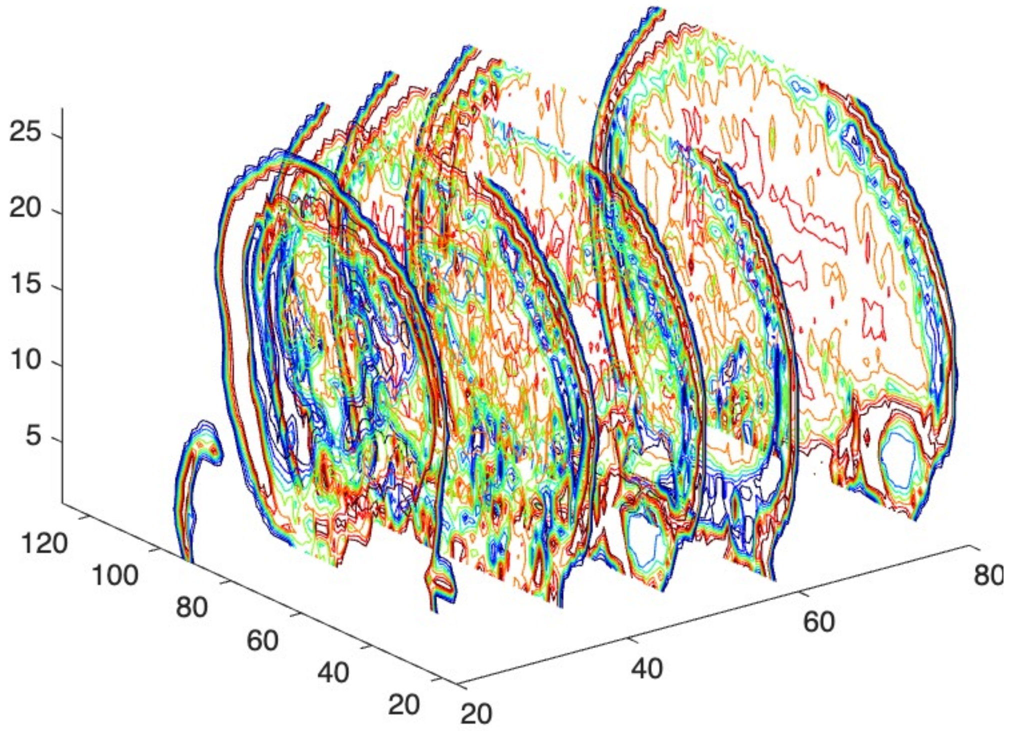
`contourslice(D,[],[],[10 15 20],8);`



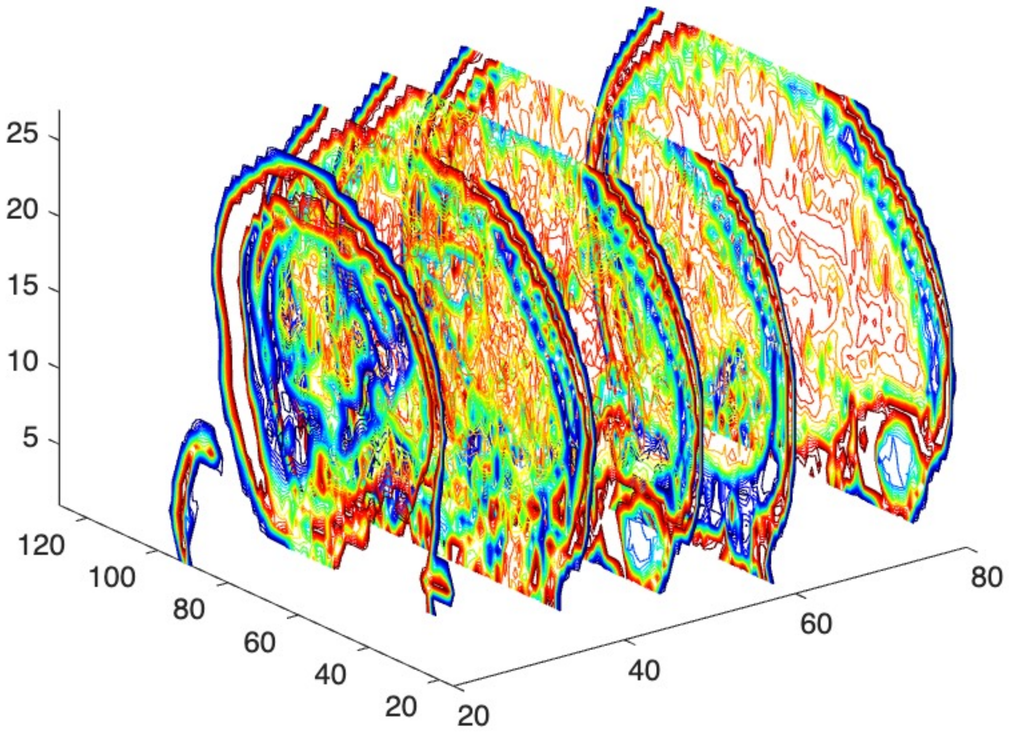
`contourslice(D,[],[],[10 12 14 16 18 20],8);`

Volume Visualization

We can also change the number of contour lines contourslice(D,[Sx],[Sy],[Sx],N).
Where N is the number of contour lines . Let's check the slices in **x** with N =8 and 20



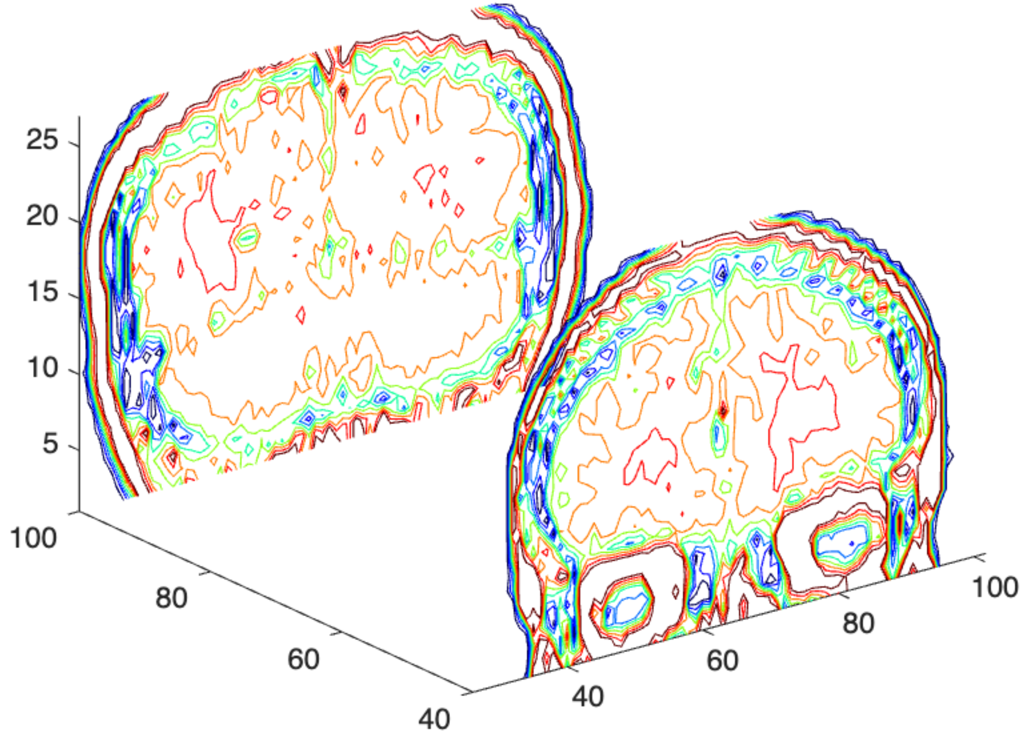
`contourslice(D,[0 10 20 30 40 50 60 80],8);`



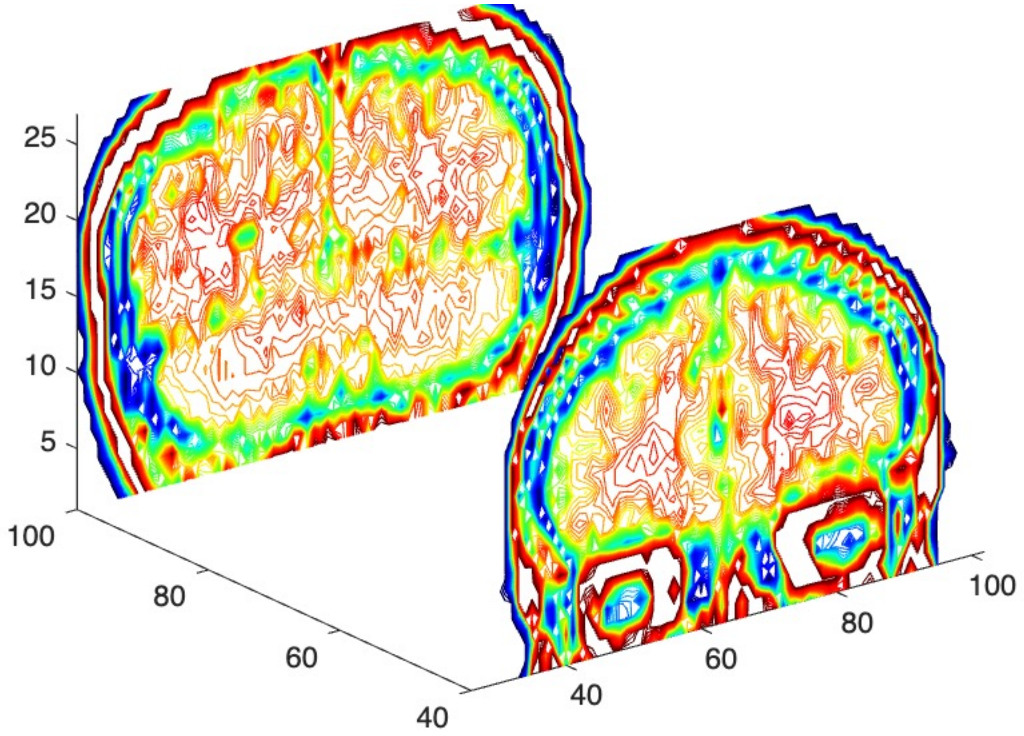
`contourslice(D,[0 10 20 30 40 50 60 80],20);`

Volume Visualization

We can also change the number of contour lines contourslice(D,[Sx],[Sy],[Sx],N).
Where N is the number of contour lines . Let's check the slices in **x** with N =8 and 50



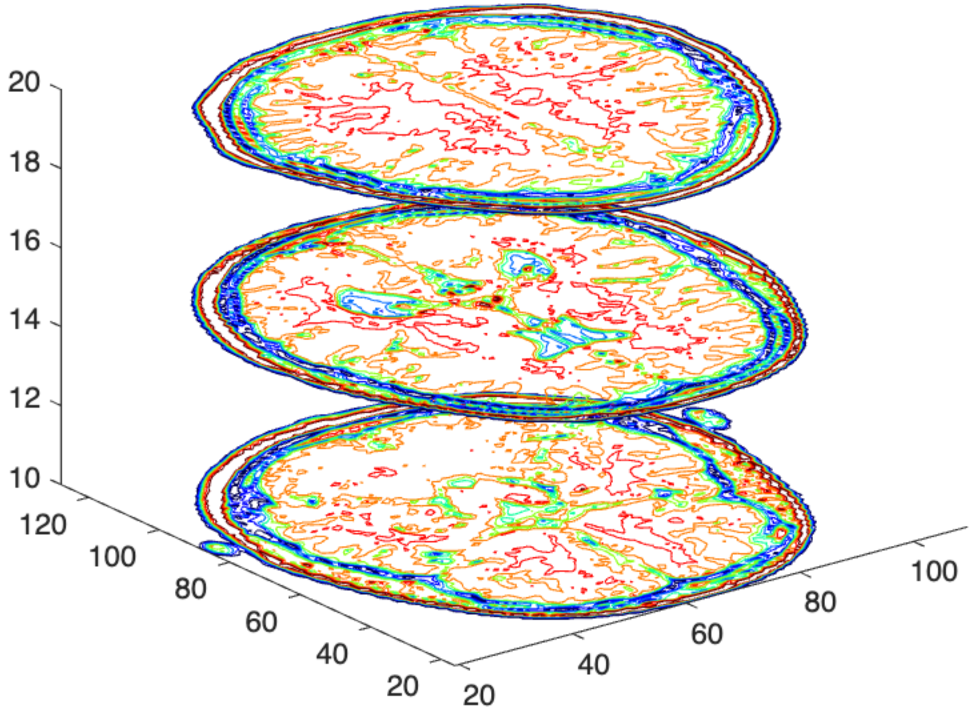
`contourslice(D,[],[40 100],[],8);`



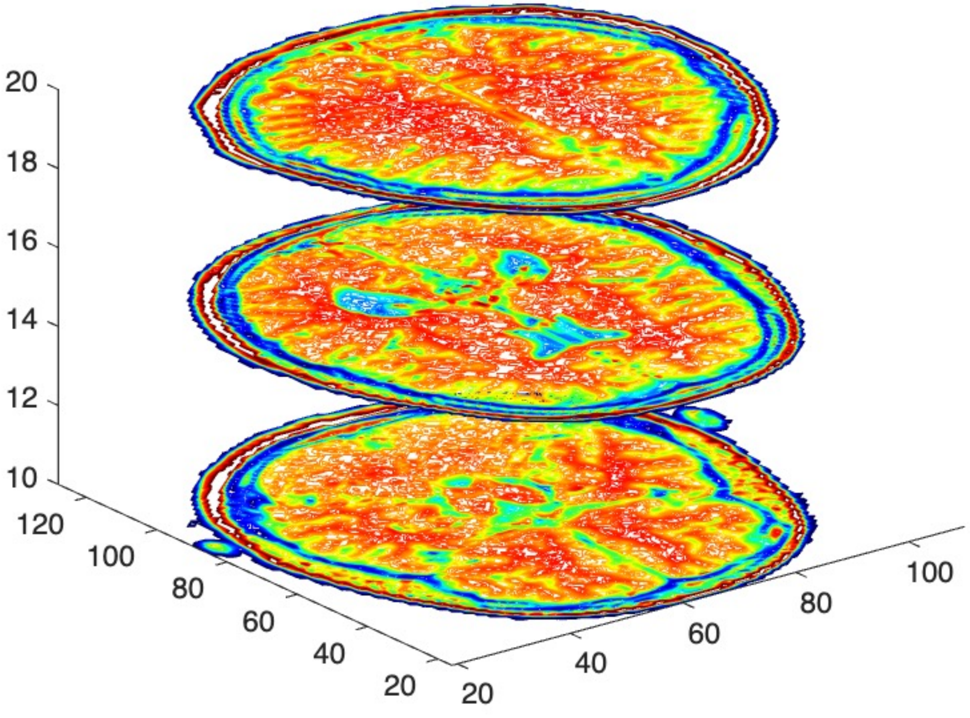
`contourslice(D,[],[40 100],[],50);`

Volume Visualization

We can also change the number of contour lines contourslice(D,[Sx],[Sy],[Sx],N).
Where N is the number of contour lines . Let's check the slices in **x** with N =8 and 75



`contourslice(D,[],[],[10 15 20],8);`



`contourslice(D,[],[40 100],[],75);`

Reflection

I love doing the MRI part! Unfortunately, I can't find other data I can do.

CRITERIA	QUALIFICATIONS	SCORE
Technical correctness	• Met all objectives. • Theory is discussed sufficiently. • Procedures and Results are complete. • Procedures and Results are verifiably correct. • Understood the lesson.	34
Quality of presentation	• All text and images are of good quality. • Code has sufficient comments and guides. • All plots are properly labeled and are visually understandable. • The report is clear.	30
Self-Reflection	• Explained the validity of results. • Discussed what went right or wrong in the activity. • Justified the self-score. • Acknowledged sources (e.g. persons consulted, references, etc.)	28
Initiative	• Experimented beyond what was required. **Somehow, added letters instead of just circles** • Made significant improvements to existing code. **No** • Analyzed limitations or potential of technique, etc. **Yes**	3

95/110