

---

## Table of Contents

Question 1 .....	1
Part (a) .....	1
Part (b) .....	1
Part (c) .....	3
Part (d) .....	8
Part (e) .....	8

## Question 1

```
phantom_image = phantom(128);
```

### Part (a)

myIntegration(image, t, theta, deltaS); As we can see that the value of delta s set to 0.5 is good enough as the integration value doesnt change much. But the time will increase linearly. For the most accurate and still reasonable performance a low value such as 0.01 or lower is favorable.

```
sampleT = 5;
sampleTheta = 75;
i1 = myIntegration(phantom_image, sampleT, sampleTheta, 2);
i2 = myIntegration(phantom_image, sampleT, sampleTheta, 1);
i3 = myIntegration(phantom_image, sampleT, sampleTheta, 0.5);
i4 = myIntegration(phantom_image, sampleT, sampleTheta, 0.1);
i5 = myIntegration(phantom_image, sampleT, sampleTheta, 0.05);
i6 = myIntegration(phantom_image, sampleT, sampleTheta, 0.01);
i7 = myIntegration(phantom_image, sampleT, sampleTheta, 0.001);
display([i1, i2, i3, i4, i5, i6, i7]);

ans =

    13.0060    12.9055    12.9230    12.9274    12.9262    12.9266    12.9266
```

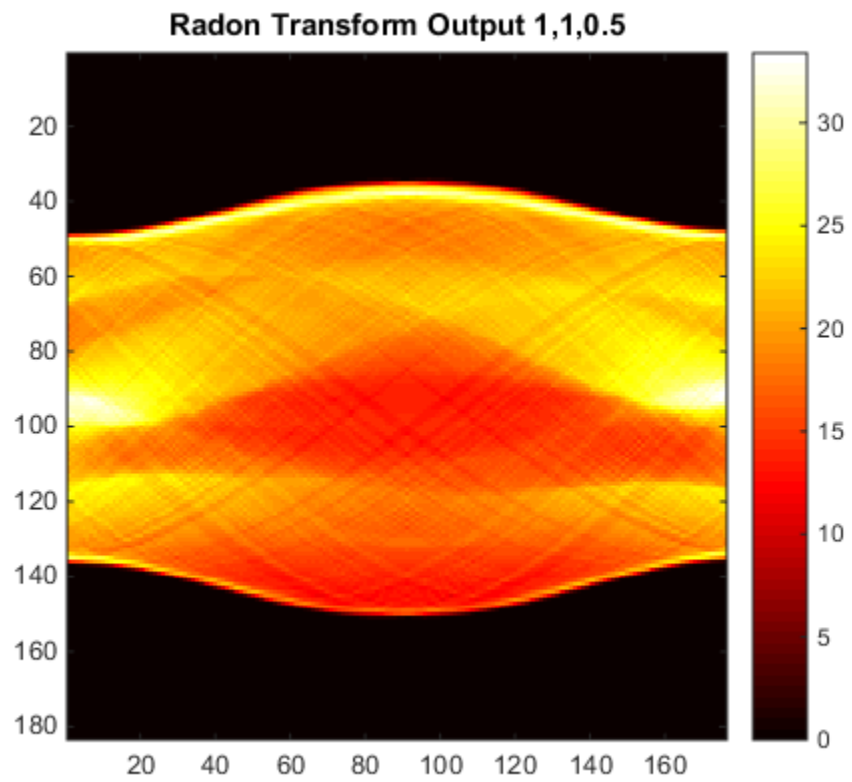
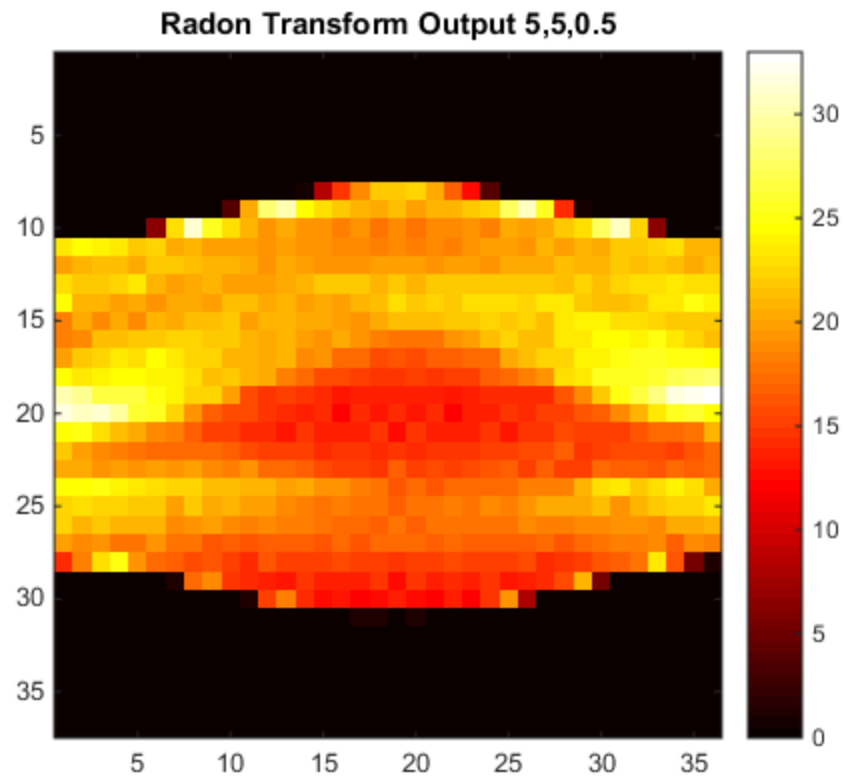
### Part (b)

```
myRadonTransform(image, deltaT, deltaTheta, deltaS);
```

```
% deltaT = 5, deltaTheta = 5, deltaS = 0.5
R = myRadonTransform(phantom_image, 5, 5, 0.5);
showImage(R, 'Radon Transform Output 5,5,0.5');

% deltaT = 1, deltaTheta = 1, deltaS = 0.5
R = myRadonTransform(phantom_image, 1, 1, 0.5);
showImage(R, 'Radon Transform Output 1,1,0.5');
```

```
Starting parallel pool (parpool) using the 'local' profile ... connected to 4 work
```



---

## Part (c)

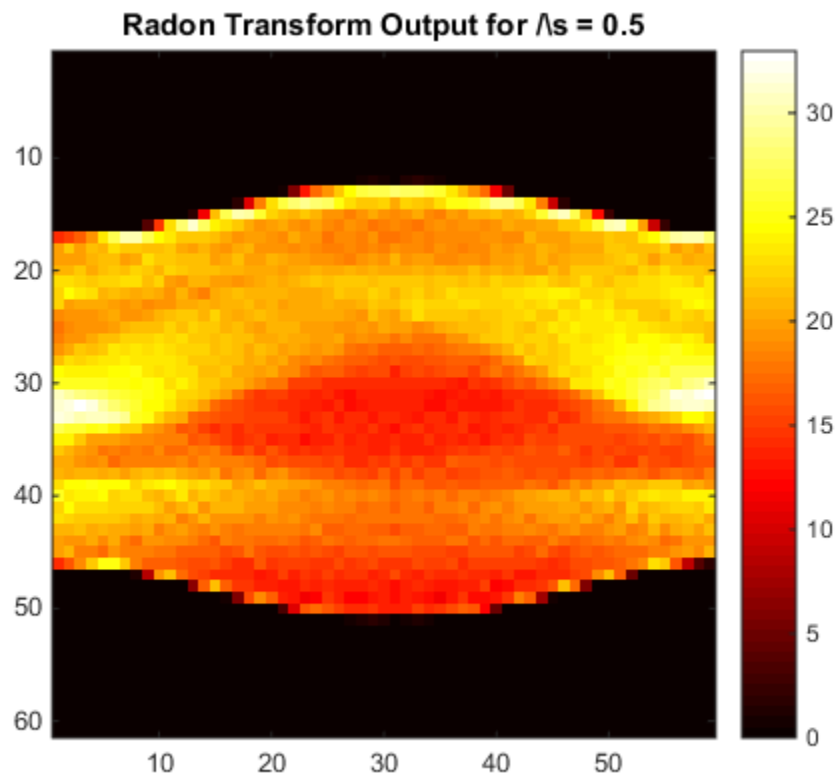
The 1D and Image appear the smoothest for  $\Delta s = 0.5$ . This is because the calculation of the integral is finer and more accurate. And the result is smoother as it is closer to the original value.

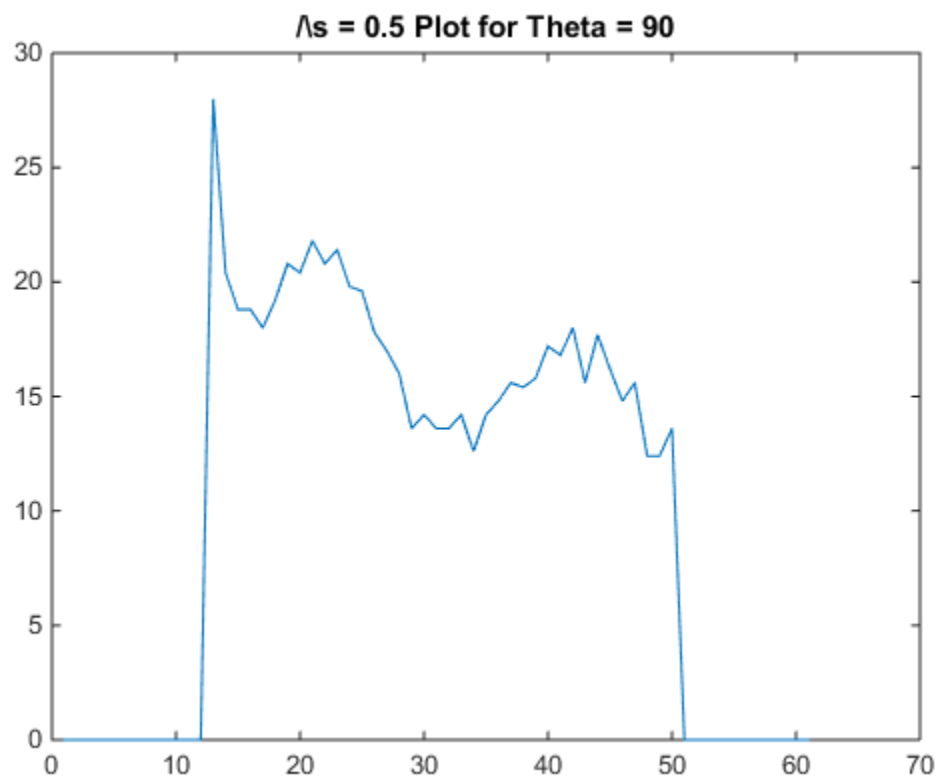
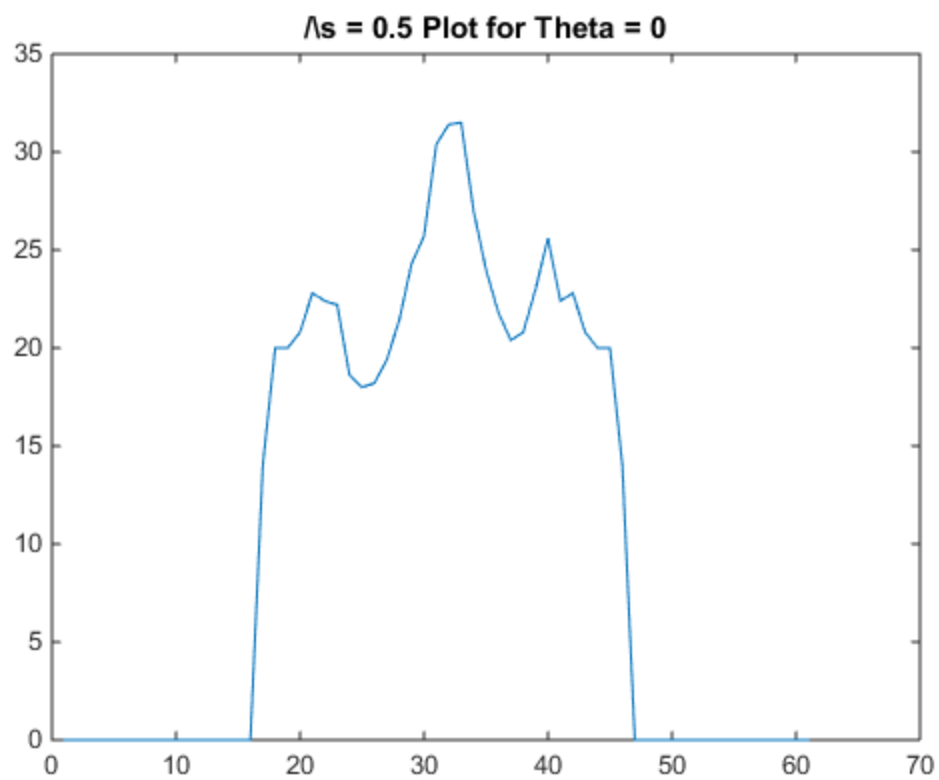
```
delta_t = 3;
delta_theta = 3;

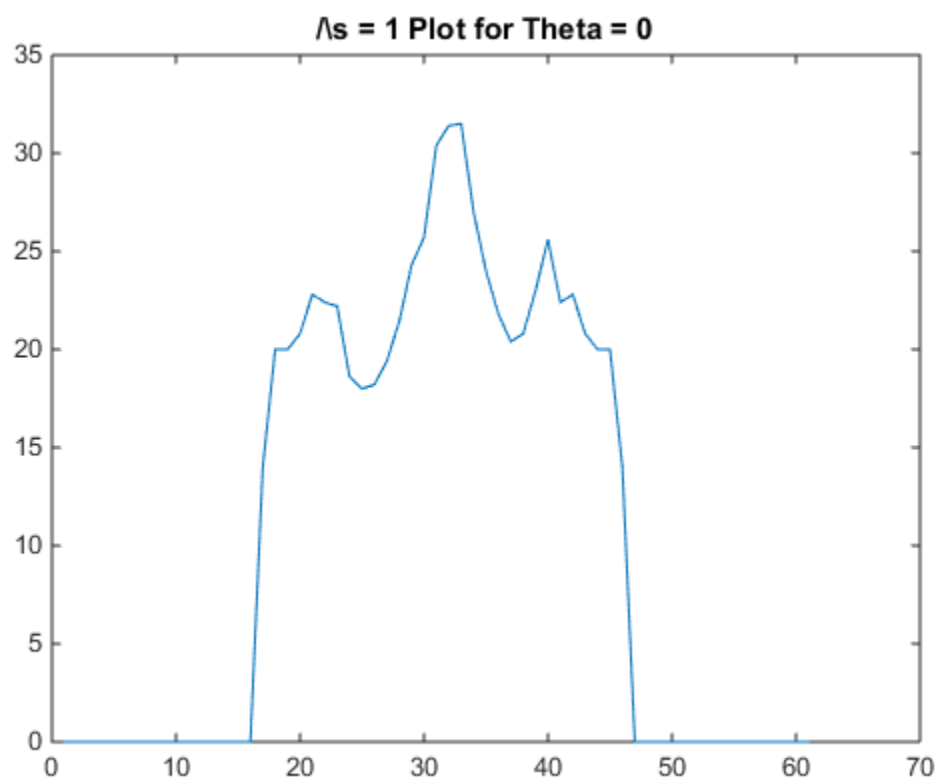
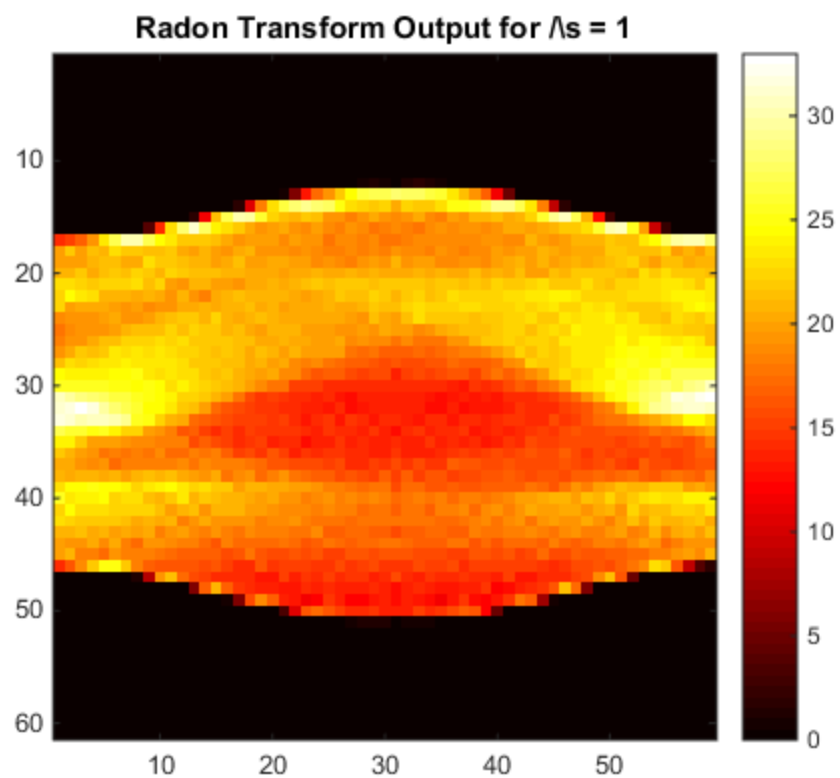
R = myRadonTransform(phantom_image, delta_t, delta_theta, 0.5);
showImage(R, 'Radon Transform Output for  $\Delta s = 0.5$ ');
plotForPartC(R, 0, ' $\Delta s = 0.5$  Plot for Theta = 0', delta_theta);
plotForPartC(R, 90, ' $\Delta s = 0.5$  Plot for Theta = 90', delta_theta);

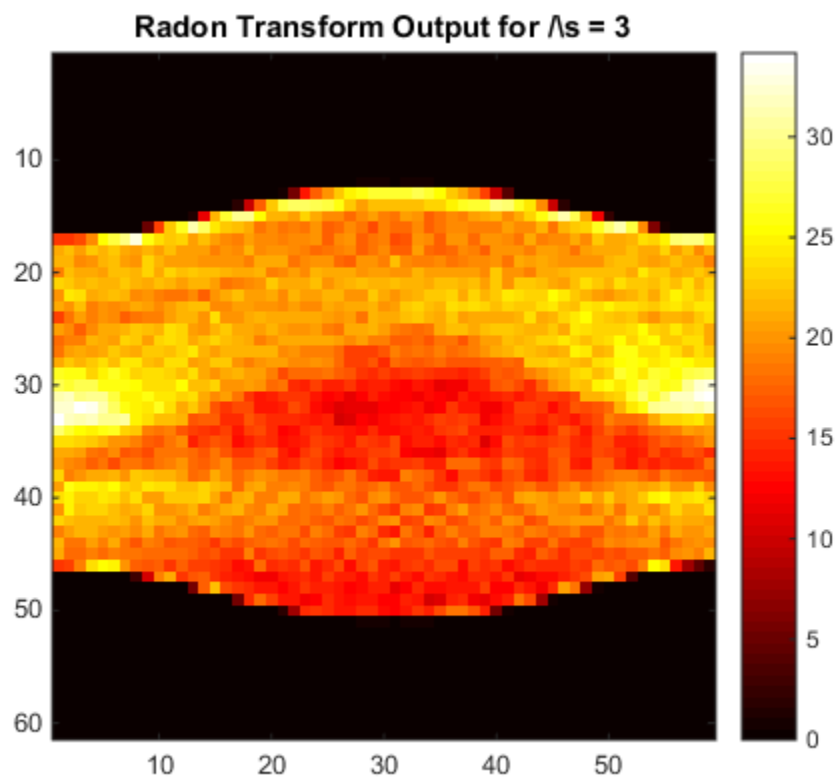
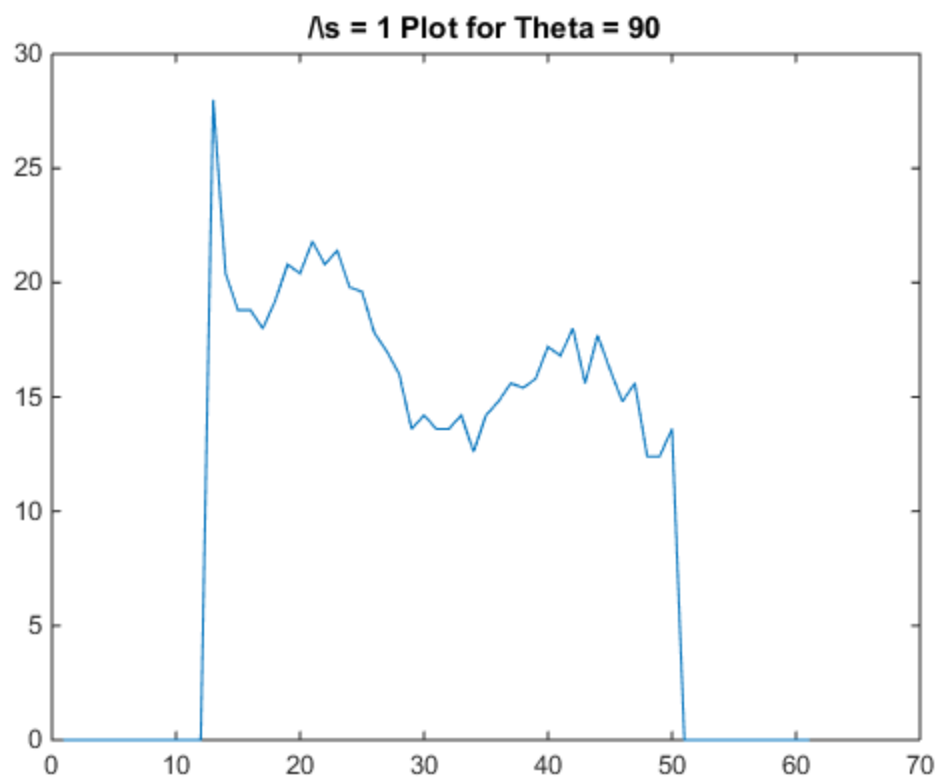
R = myRadonTransform(phantom_image, delta_t, delta_theta, 1);
showImage(R, 'Radon Transform Output for  $\Delta s = 1$ ');
plotForPartC(R, 0, ' $\Delta s = 1$  Plot for Theta = 0', delta_theta);
plotForPartC(R, 90, ' $\Delta s = 1$  Plot for Theta = 90', delta_theta);

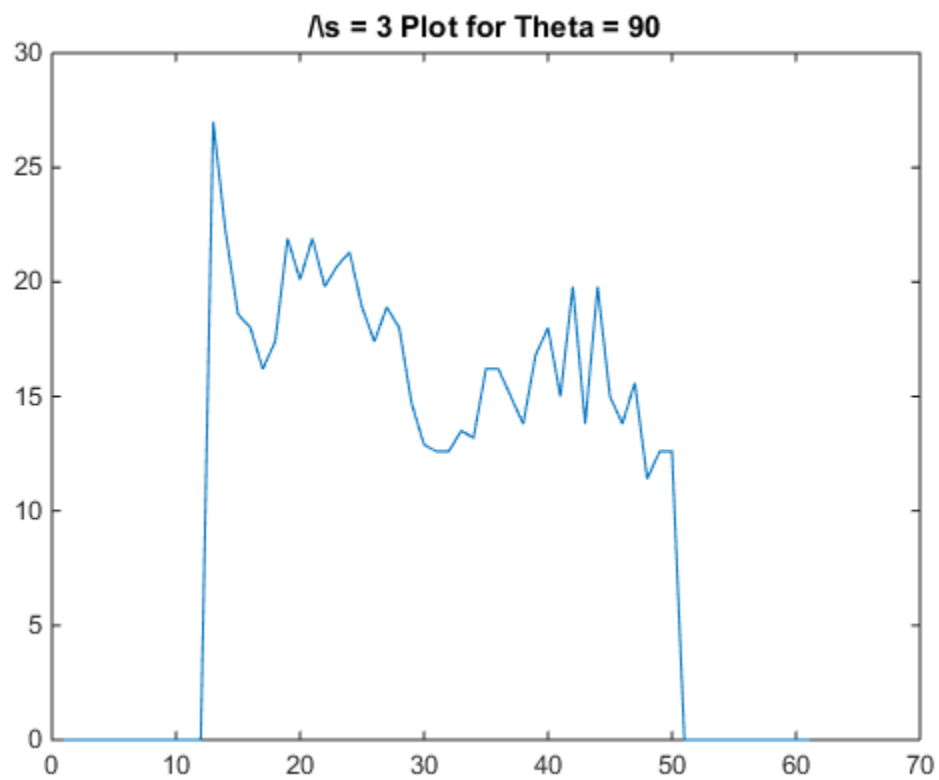
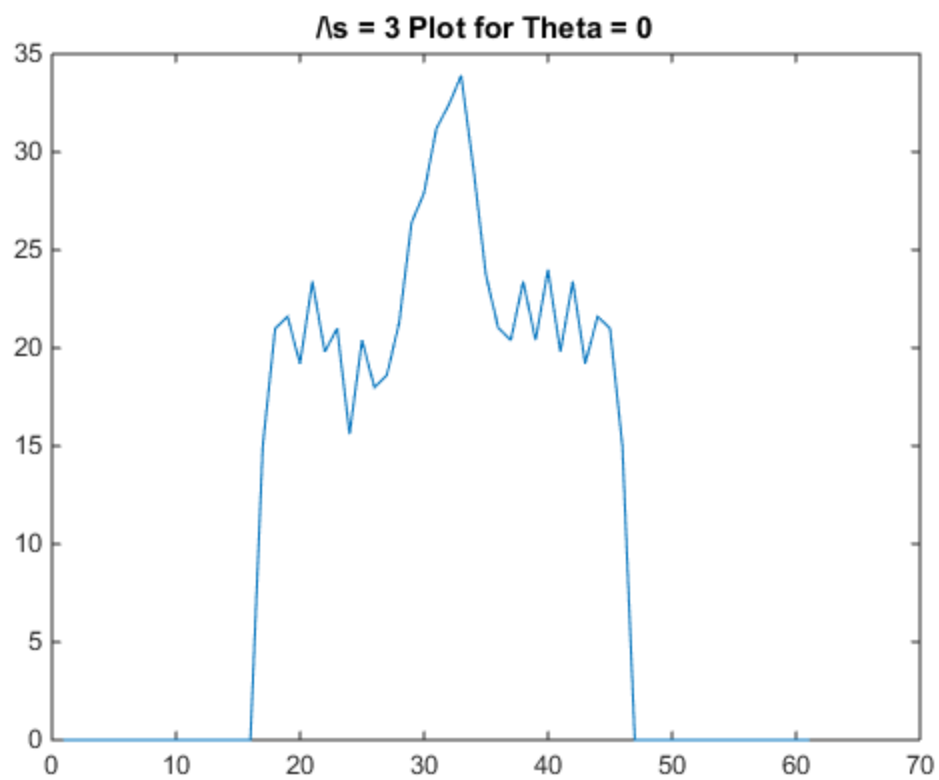
R = myRadonTransform(phantom_image, delta_t, delta_theta, 3);
showImage(R, 'Radon Transform Output for  $\Delta s = 3$ ');
plotForPartC(R, 0, ' $\Delta s = 3$  Plot for Theta = 0', delta_theta);
plotForPartC(R, 90, ' $\Delta s = 3$  Plot for Theta = 90', delta_theta);
```











---

## Part (d)

Value of  $\Delta t$  and  $\Delta \theta$  should be small. As large values make the image very boxy and unclear. However too small values will make the code very slow. As we saw in the first part a value of 1 is enough to resolve the values as close to one pixel apart. Also, we saw that it resembles very close the true radon transform. Hence a  $\Delta t = 1$  is good enough and  $\Delta \theta = 1$  is also needed to resolve all pixel far away.

## Part (e)

For this case a pixel grid which has enough resolution to detect objects of importance and their boundaries should be chosen. For instance, most internal arteries need to be well resolved and so do some alveolar ducts. Hence a pixel size corresponding to 0.1mm or so will work. Lower can be done to improve resolution.

A  $\Delta s$  of 0.5pixel width will be able to resolve most of the corner cases well as we have previously seen. Ofcourse doing 0.01 or so will improve reconstruction upto a limit.

For  $\Delta s \leq 1$ , the benefit of reducing will fade after a point. (Seen in part(a)). Hence after a point, no change will be visible, the computation time will increase with no benefit. Some floating point artifact might even corrupt the result For  $\Delta s > 1$ , the image will be very blurry and not accurate. It will be unable to clearly resolve the various parts of the body.

*Published with MATLAB® R2014b*