The Radon Transform

Aside from the slope-intercept form of the equation of a line , $y = mx + b$, we can also express a line by its normal representation, given by

$$x\ \cos\quad \theta + y\ \cos\quad \theta = \rho. \quad (1)$$

Here ρ is the length of the shortest ray from the origin to the line and θ is the angle formed by that ray with respect to the x-axis as shown below in Figure 1.

Let $f(x, y)$ describe a 2d object(note: it can be a slice of a 3D object!). The project of the object along a line given by equation (1) is given by

$$g(\rho, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\delta(x \cos\theta + y \sin\theta - \rho)dxdy \qquad (2)$$

Eqtn 2 is the definition of the Radon transform. A Radon transform of a function $f(x, y)$ is expressed in notation as $g(\rho,\ \theta)\ =\ \Re[f(x, y)]$.

When the Radon transform $g(\rho,\ \theta)$ is plotted with ρ and θ in Cartesian coordinates, the image is called a sinogram.

The Matlab code below creates a circle, computes its Radon transform, and displays it.

```
%% Make a circle
      x = linspace(-1,1, 100); y = x;
      [X,Y] = meshgrid(x,y);
      R = sqrt(X.^2 + Y.^2);
      C = zeros(size(X));
      C = R < 0.5;
      figure (1);
      subplot(1,2,1);
      imshow(C);

%% Make a rectangle
      C = zeros(size(X));
      C(20:80,40:60)=1.0;
      figure (1);
      subplot(1,2,1);
      imshow(C);

%% Compute Radon Transform
      theta = 0:180;
      [R,p] = radon(C,theta);

%% Display sinogram
      subplot(1,2,2);
      iptsetpref('ImshowAxesVisible','on')
      imshow(R,[],'Xdata',theta,'Ydata',p,'InitialMagnification','fi
      t')
      xlabel('\theta (degrees)')
      ylabel('\rho')
```

```
        colormap(gca,hot), colorbar

%% plot cross section
        figure(2);
        plot(p,R(:,100));xlabel('\rho');ylabel('g( \rho )');
```

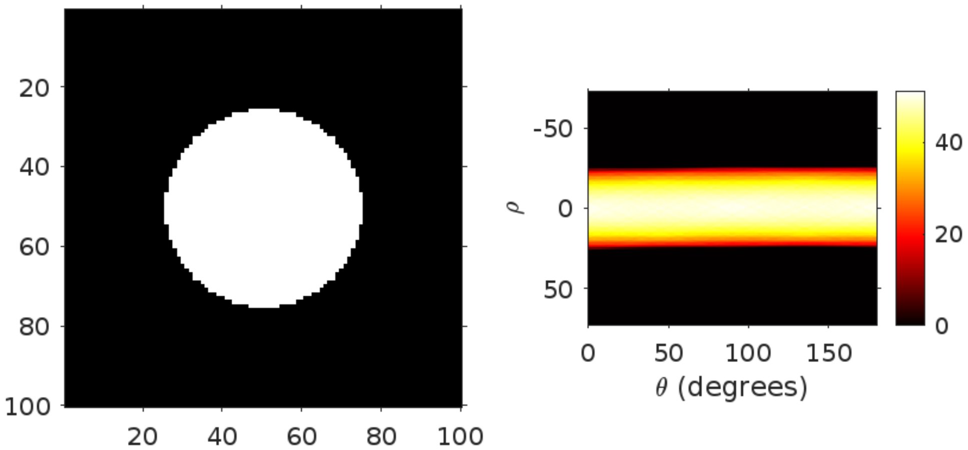

Figure 1. Circle image and its Radon Transform displayed as a sinogram.

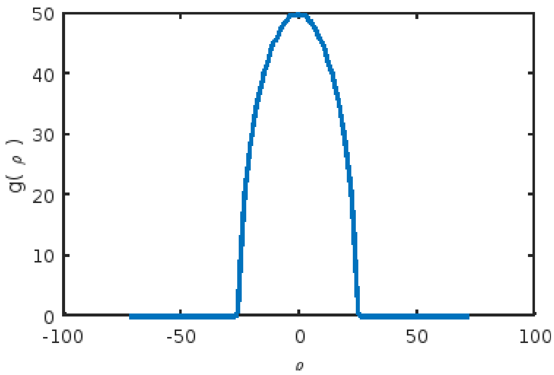

Figure 2. Cross section of sinogram at $\theta \;=\; 100^{o}$

Back-projection to Recover $f(x, y)$

To recover f(x,y), we can back-project the $g(\rho,\ \theta)$ for different angles. For each angle θ_k we replicate the $g(\rho, \theta_k)$ in a $f_{\theta_k}(x, y)$, that is

$$f_{\theta_k}(x, y) = g(\rho,\ \theta_k) = g(x\ \cos\theta_k + y\ \sin\quad \theta_k, \theta_k)\ .\ (3)$$

You can imagine this as the cross section in Figure 2, being replicated in (x,y) to form a 2D image.

In general at any angle

$$f_\theta(x, y) = g(x\ \cos\theta + y\ \sin\quad \theta, \theta).\quad (4)$$

The final image is formed by integrating over all the back-projected images,

$$f(x, y)\ =\ \int_0^\pi f_\theta(x, y) d\theta. \qquad (5)$$

In Matlab, the process discussed above is performed by the inverse radon transform. Here's the code:

```
%% Recover image
      Cp = iradon(R,theta,"linear","None");
      figure(3);
      imagesc(Cp); colormap("gray"); axis equal
```

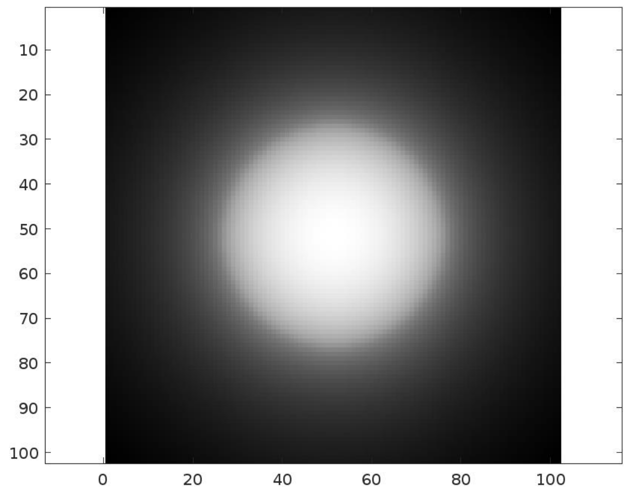

Figure 3. Inverse Radon Transform given the sinogram in Figure 1.

Notice that the recovered image appears blurry. In the next lesson, we will discuss how we can sharpen the image.

Activity 1

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

Reference

Gonzales and Woods, Digital Image Processing, 3rd ed., Chapter 5.11.3