Note for Hough and Radon transform

I investigate two transform in image recognition to distinguish a line in picture.
1) Hough transform
2) Radon transform

Hough transform:

How it works:

Hough transform is better with binary images like 0 and 1 image. In a binary image with few non-zeros pixels, Hough transform maps the individual pixels from the image domain into a shape in the parameter domain. If we have one dot in the image, it will show a straight line in parameter space. We assume there are several pixels in image and they are appear in same line for simple analysis , in the parameter space it will appear a intersection of several lines after Hough transform. If there's a intersection , we make accumulator of pixels in parameter space +1. Then we do this process to all pixels of image, and statics the points in each pixel, the max value of accumulator is standing for a line in image.

Algorithm:

```
for  x  in  N:                                   // assume we have N*M pixels in image I
    for y in M:
        if I(i,j)  not equal 0:
            for  θ  in D:            // assume in parameter space (θ,ρ) ,we have D length for θ
                ρ  = x* cosθ  +  y *  sinθ
                accumulator[ θ,ρ ] +=1       //accumulator has same shape as parameter space.
```

In this algorithm , The compute complexity is $\approx O(M*N*D)$

I have tried the hough transform to search line in re-bin data. This compute is slowly. As hough only calculate the binary value image, we need to set a threshold to our data.

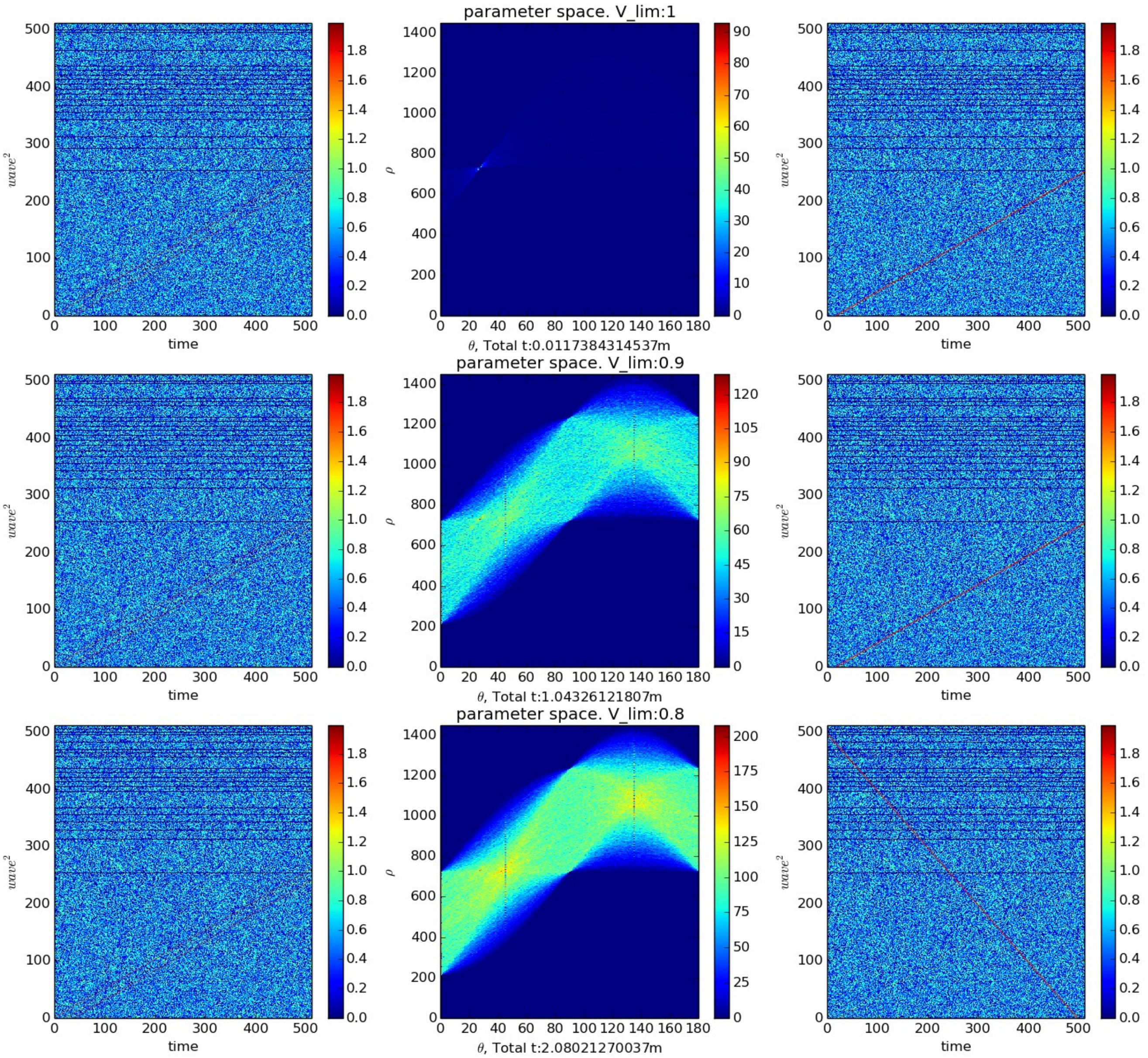
$$if\ I(x,y)>threshold, I(x,y)=1$$
$$if\ I(x,y)<threshold, I(x,y)=0$$

In the figure 1 shows the threshold with 1 ,0.9,0.8 separately.

Hough space is discretized to cover the expected range of target object geometries with the resolution appropriate to the application. The resolution must be sufficient to describe target objects with sufficient resolution, but excessively fine resolution is undesirable as it results in a prohibitively large Hough space.

Radon Transform:

(cited from :The Radon Transform - Theory and Implementation. Toft peter Aundal's thesis , published in 1996)



1.2 Defining the (p, τ) Radon Transform

The Radon transform can be defined in different ways. The definition used, e.g., within seismics [19] is perhaps the easiest to comprehend. The Radon transform $\check{g}(p, \tau)$ of a (continuous) two-dimensional function $g(x, y)$ is found by stacking or integrating values of g along slanted lines. The location of the line is determined from the line parameters; slope p and line offset τ.

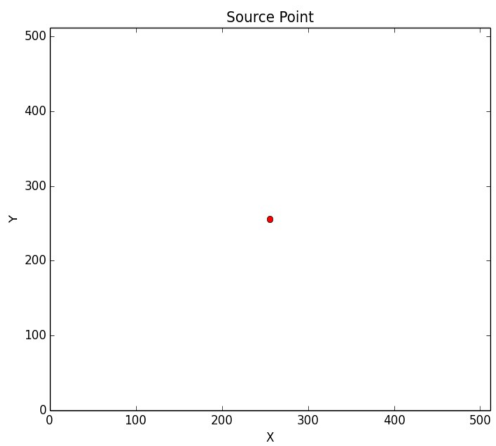
$$\check{g}(p, \tau) = \int_{-\infty}^{\infty} g(x, px + \tau)\ dx \tag{1.1}$$

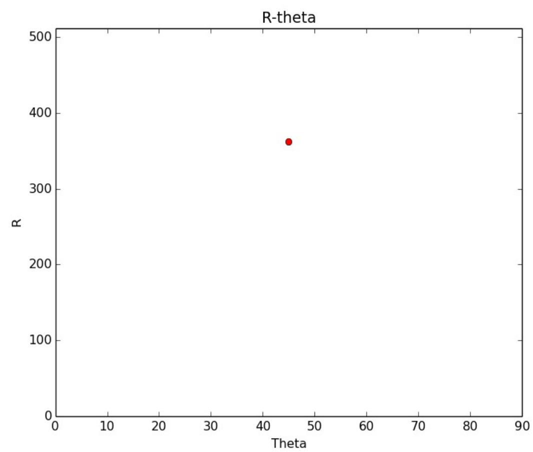
The Radon transform has almost same algorithm as Hough, the difference is it doesn't has a accumulator but accumulate I(x,y) instead.

As Hough transform is not quite suite to our process , We'll ignore it and take consideration about Radon transform only. For easier understanding , We show what it is that point source and line source after our polar-coordinates transform and Radon transform. For Radon transform ,there are two kinds of expressions in parameter space. The common one which people often use is (θ, ρ) , θ stand for angle corresponding to the angular orientation of the line. ρ stand for the shortest distance from the origin of the coordinates system. Another set of parameter is (p, τ) , p is the slope of the source, τ is the offset of line in image. Certainly , The two sets of parameter are same.

Point source (x*,y*) or (p*, τ *) or (θ *, ρ *)

polar-coordinates transform:





Radon transform:

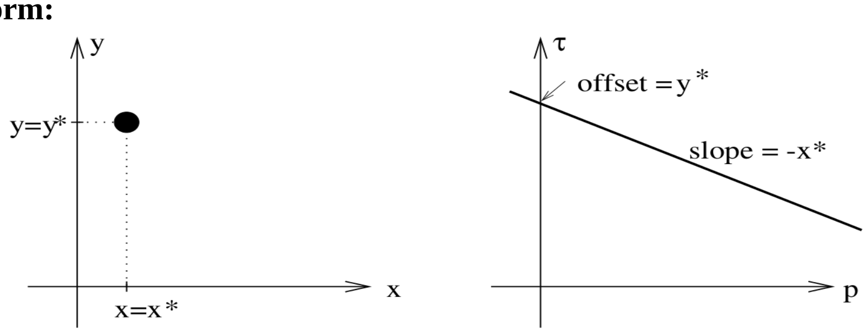


Figure 1.3 Left: A two dimensional function that only is non-zero in the point $(x, y) = (x^*, y^*)$. Right: The corresponding Radon transform (slant stacking result). Only when the Radon domain parameters matches the parameters of the line a non-zero result is found.

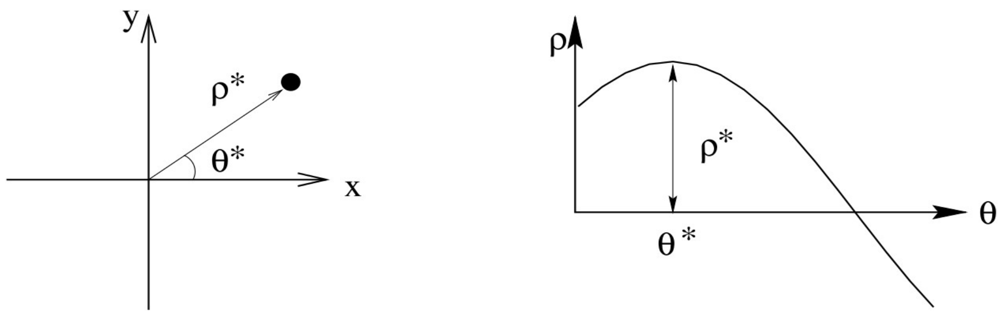
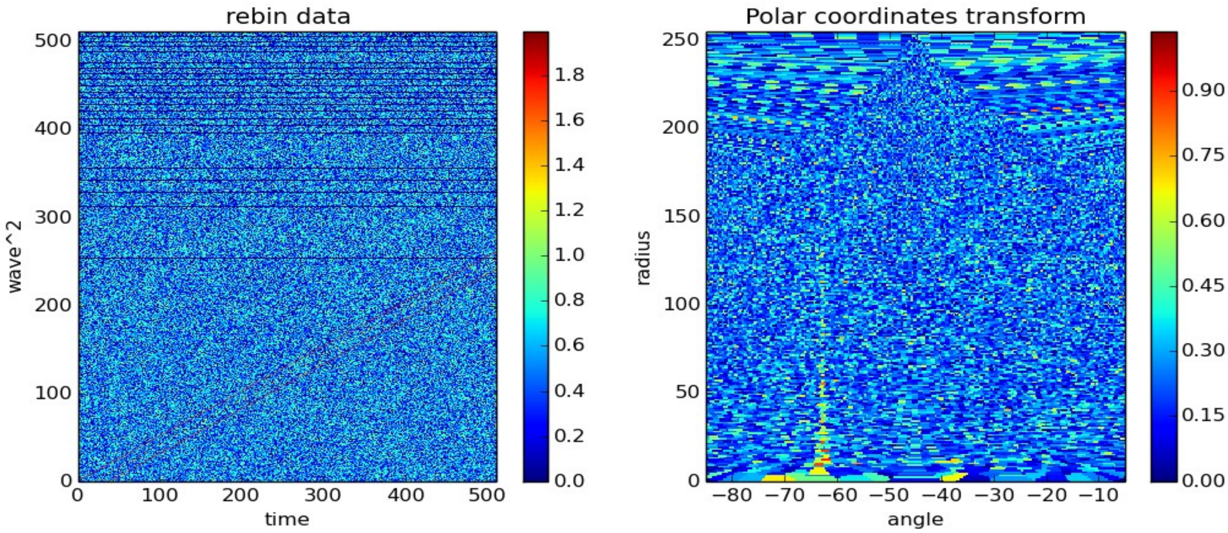


Figure 2.2 To the left is shown a point source, and to the right is shown the corresponding normal Radon transform.

Line Source (p^*, τ^*) or (θ^*, ρ^*),

ploar-coordinates transform:



Radon transform:

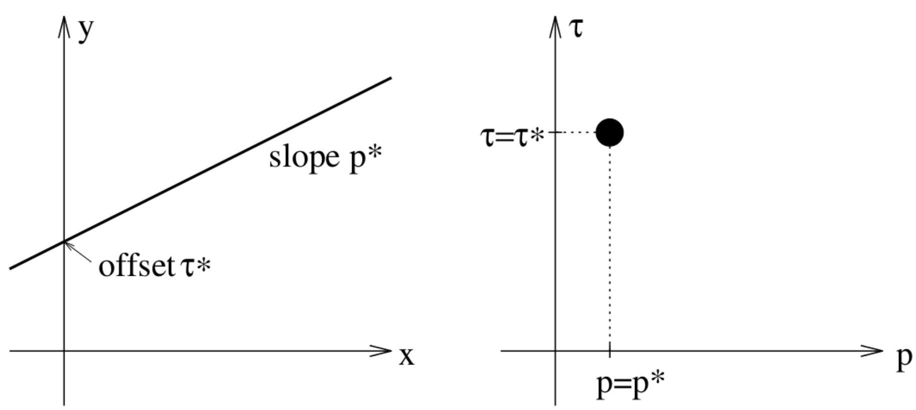


Figure 1.4 Left: A two dimensional function that is only non-zero when on the line. Right: The corresponding Radon transform (slant stacking result). When the Radon domain parameters matches the parameters of the line, a peak is found positioned at the parameters of the line in the image. The finite terms in the parameter domain are here ignored for sake of clarity.

Conclusion:

In the original re-bin map, It worth to do a 2DFFT, as 2D-FFT only have a complexity as O(N^2 logN) assuming the image has a square shape as N*N. For Radon transform ,it's O(N*2*D) where D is the parameter's dimension.

After 2D-FFT , signal will appear in the center of image. In this case, Radon will only has 1 parameter θ to consider , because the ρ is equal to 0. So that we can find the algorithm for this case:

```
if tau = 0:
    for i in N:
        for θ in D:                //D is the length of theta in parameter space
            Value[ θ ] += I [i ,i*tan( θ )]
```

We can see the complexity is O(N*D)
In our algorithm, after polar-coordinates transform, We will sum along the radius for each θ . The compute cost look like the same.

In implementation, Radon transform facing the same problem as us. That's How to make a decision about parameter resolution. There's a mathematical analysis about the resolution choosing in Tott's thesis.
(page 8-14)

Interestingly , 1.31,1.32 and 1.34,1.35 (page 11) got same result of constrain on parameter step. And 1.31 1.32 are deducting from Shannon sampling information theory, and 1.34,1.35 are considered under pixel information conservation. That's similar with Yichao's suggestion.

As we have the constrain on δk from δDM , Then We can use the pixels conservation to calculate the δr . This process might keep the periodical information along the radius direction.

The Radon transform also has a relationship with 2D-FFT. 2D-FFT on g(x,y) is equal to using Radon transform first and then take a 1-D FFT on function. The following is the conduction:

$$F_{(1)}R[g(x,y)]=F_{(1)}R_f(\rho\cdot\theta)=F_{(2)}[g(x,y)]=F(u,v)$$

prof:

$$F(u,v)=\iiint g(x,y)\exp(-j2\pi s)\delta(s-ux-vy)dsdxdy$$

let s=m* p

$$F(u,v)=m\cdot\iiint g(x,y)\exp(-j2\pi mp)\delta(mp-ux-vy)dpdxdy$$

In Fourier space , let $u=m\cdot\cos(\theta),v=m\cdot\sin(\theta)$, and because $\delta(ax)=\frac{\delta(x)}{|x|}$

Then we get:

$$F(u,v)=\int \exp(-j2\pi mp)\iint g(x,y)\delta(p-xcos(\theta)-ysin(\theta))dpdxdy$$