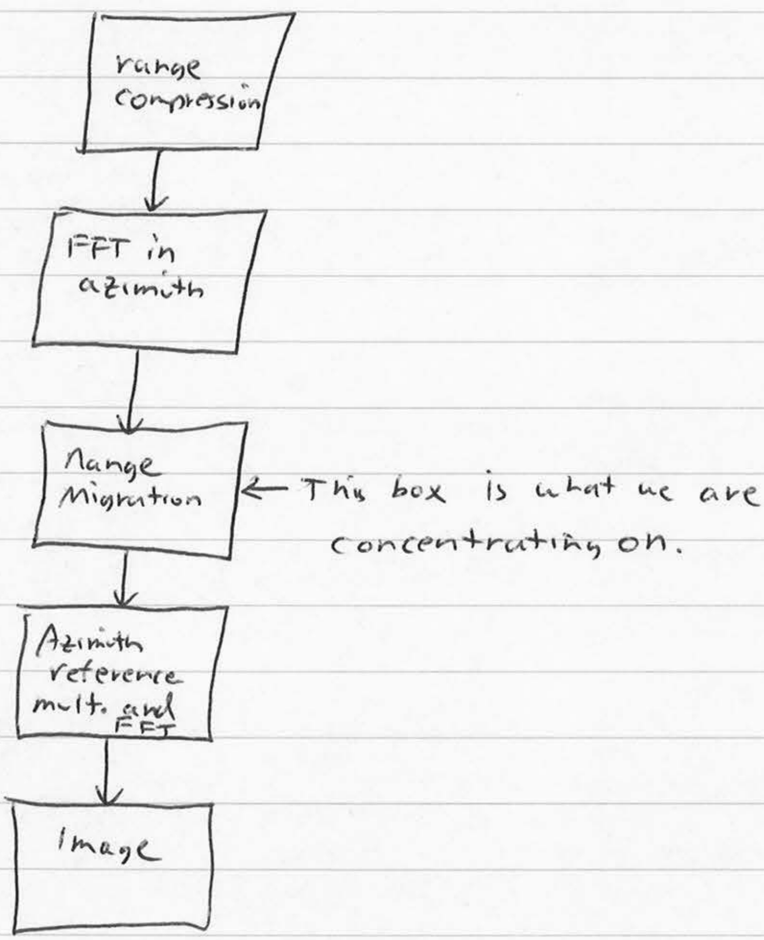
GP265 / EE355

Handout #27

Implementation issues in range migration

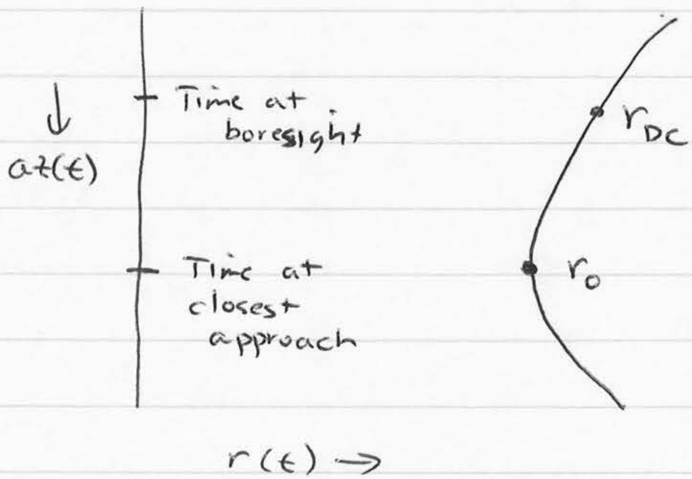
Let's assume we decide to process data using the interpolation method for range migration. We will go through design of a range migration routine for our SAR processor. Our flow diagram looks like



Here we have to decide a major issue for our processor: Just how we interpolate determines where the output pixels appear in the radar image. This means a great deal in terms of how we interpret our resulting image.

Deskewing

As we have found before, typically we process data at some centroid offset from the exact side-looking direction. Let's plot the range history first:



Two coordinate systems naturally present themselves: one referenced to the azimuth time and r_0 corresponding to closest approach, and one referenced to the azimuth time and r_{DC} for the antenna boresight direction. Since the (boresight time, r_{DC}) coordinate system is dependent on our choice of Doppler centroid while the (closest approach, r_0) system is independent of processing parameters, the former is called a <u>skewed</u> coordinate system while the latter is <u>deskewed</u>. It doesn't really matter which we use as long as we are consistent with our processing code and our analysis approach.

Some argue deskewing leads to more cartographically accurate images. This is largely a philosophical issue as a further processing step to convert from slant range to ground range is always necessary for mapping.

A range migration routine

So, what might a range migration interpolation routine look like? Assume we start off with a range compressed patch, which we have already transformed in the azimuth direction.

We want to trace out a different migration path for each range bin. Hence we begin by creating a loop over the range bins:

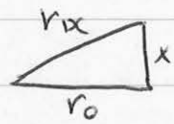
do $i = 1$, nbins

$r = r_0 + (i-1) \cdot \delta r$ ← get range to line

(Note here as usual $\delta r = \frac{c}{2 f_s}$)

f_{DC} = <some function of range, perhaps a constant>

$r_{DC} = r\left(1 + \frac{f_{DC}^2}{8v^2}\right)$ ← from construction



$f_R = \frac{-2v^2}{\lambda r_{DC}}$

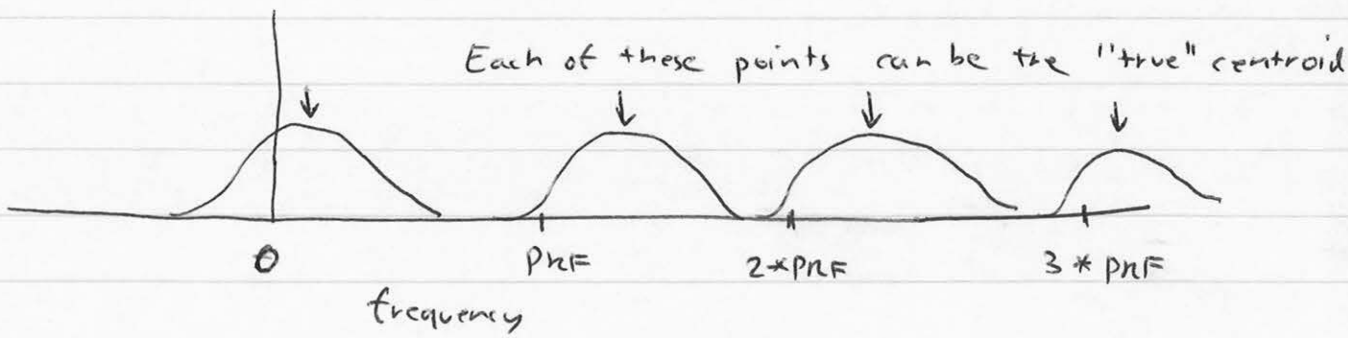
where we compute the range at Doppler centroid r_{DC} and the chirp rate f_R as a function of range.

Now, we need to find the range to the object at each available Doppler bin, this is determine the range history as a function of azimuth frequency. Hence, we need to loop over frequency bins:

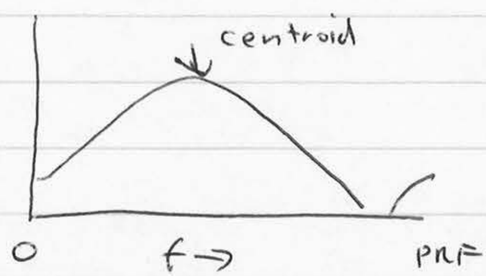
do $j = 1$, nazfft ← j is azimuth bin # after transform

$freq = \frac{(j-1)}{nazfft} \cdot prf$

Here is a tricky part: we want to process on the correct "ambiguity". Recall that our azimuth spectrum is sampled at the prf and hence repeats in the frequency domain with that period:



The azimuth spectrum will be identical for each assumption of fdc. Here we must go out on a limb and say while we don't know f_{DC} exactly, we probably do know it to the precision of a prf, and can thus properly calculate r_{DC} from the centroid measurement.



The centroid estimation gives f_{DC} to the "ambiguity" of a prf only!

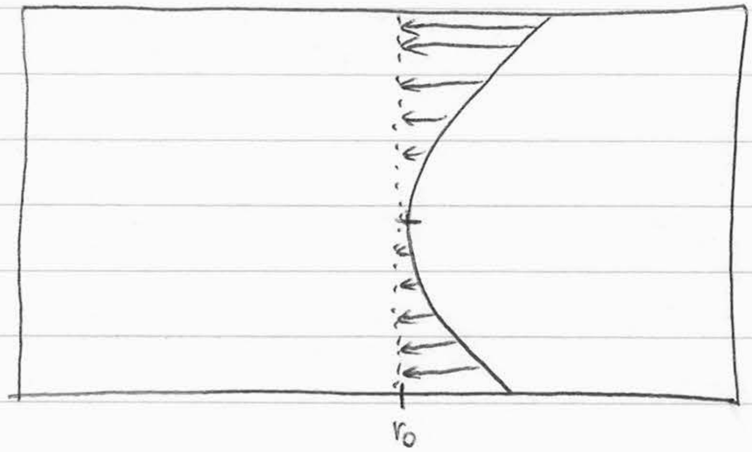
The following code ensures that we ~~estim~~ use the proper frequency for each bin, and ensure that the frequency used is within $\frac{1}{2}$ prf of the true centroid.

$$n = \text{nint}\left(\frac{freq - f_{DC}}{prf}\right)$$

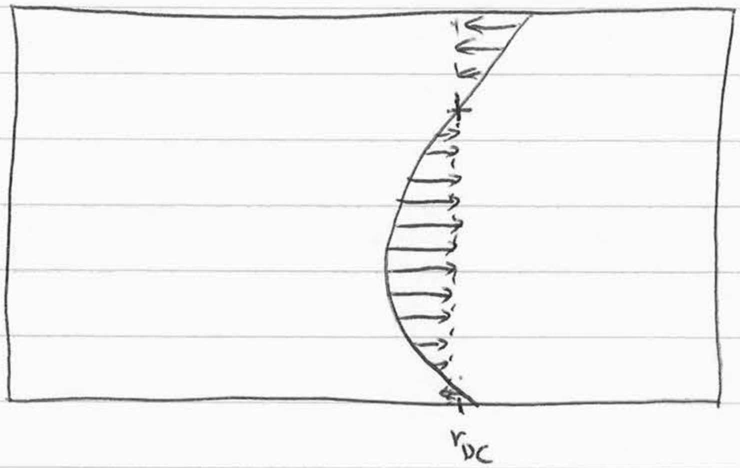
"nint" is nearest integer

$$freq = freq - n * prf$$

Now, we need to decide about deskewing. Will we resample points to lie along the r_o bin or the r_{DC} bin?



Deskewing



Not Deskewing

We need to ~~tran~~ shift the points by the difference of the true range history and the desired reference line in either case.

So, what is our range history as a function of frequency?

$$r^2(t) = r_o^2 + x^2$$

$$= r_o^2 + \left(\frac{f \cdot \lambda \cdot r(t)}{2v}\right)^2$$

~~$r^2(t) = r_o$~~

$$r^2(t)\left(1-\left(\frac{f\lambda}{2v}\right)^2\right) = r_o^2$$

$$r^2(t) = r_o^2\left(1+\left(\frac{f\lambda}{2v}\right)^2\right)$$

$$r(t) = r_o\left(1+\frac{f^2\lambda^2}{8v^2}\right)$$

So, for the deskew case, the offset $\Delta(f)$ is

$$\Delta(f) = \frac{f^2\lambda^2 r_o}{8v^2}$$

In the non-deskewed case

$$\Delta(f) = r_o\left(1+\frac{f^2\lambda^2}{8v^2}\right) - r_o\left(1+\frac{f_{DC}^2\lambda^2}{8v^2}\right)$$

$$= \left(f^2 - f_{DC}^2\right)\frac{\lambda^2 r_o}{8v^2}$$

We simply pick one of the above, and then interpolate:

$$V_{out} = \sum_{k=-3}^{4} v\left(i + \mathrm{nint}\,\frac{\Delta(f)}{\Delta r} + k,\ j\right)\mathrm{sinc}\left(k + \mathrm{frac}\left(\frac{\Delta(f)}{\Delta r}\right)\right)$$

where frac( ) is the fractional part.

Also $v(i,j)$ is the transformed array.

Now that we have resampled the array to straighten out our range histories, we can multiply our columns by the ~~the~~ conjugate of the transformed reference functions and create the images.

However, we still have the issue of deskewing in the azimuth direction (we have already corrected for the range part of the deskew correction). So, we must pay attention to the along-track position of the output pixel.

In other words, if we want to reference our output to the non-deskewed ("natural") position, we may need to apply a linear phase correction in the definition of the reference function. This simply means being careful in the matched filter definition: do we apply a reference with zero time at the point of closest approach or at the boresight time?