VII Range Migration Algorithm
(For Fast 3-D SAR Imaging)

7.0 Review

1) FMCW, Stepped freq., Pulsed radars } generalized signal model.

2) 3-D SAR reconstruction using BP/TDC & MIMO radar.

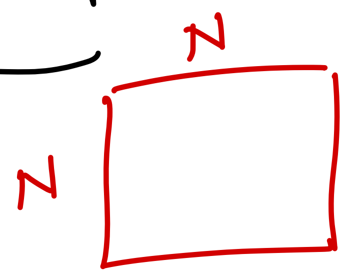
3) FMCW SAR Imaging using HFSS & MATLAB.

4) Fast 3-D SAR Imaging & use experimental data for real world 3-D Imaging.

7.1 Introduction / Motivation

looks like a Fourier Transform.

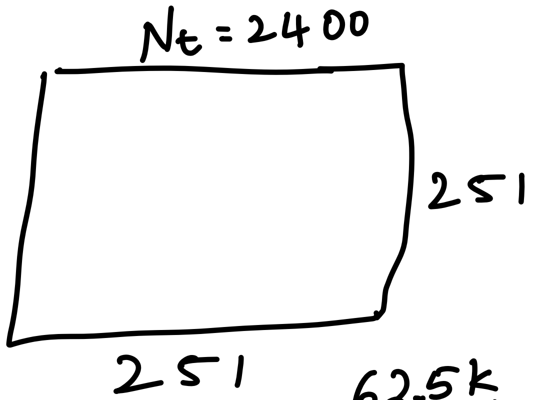
$$> \underbrace{\tilde{f}(x,y,k_t)}_{\underbrace{N \times N \times N_t}_{N^3}} = \underbrace{\sum_{K} \sum_{x'} \sum_{y'}}_{\underbrace{N \times N \times N_t}_{N^3}} \underbrace{s_b(x',y',k)\, e^{2jk\sqrt{(x-x')^2+(y-y')^2+z^2}}}$$



N time samples

$O(N^6)$

Eg: $251^4 \cdot 2400^2 \approx 10^{16}$



10^9 $\Rightarrow$ 10^7 secs $\approx$ 115 days.

62.5k pixels.

In MATLAB: $\approx$ 3500 days → 40 seconds.

1) Range Migration Algorithm.

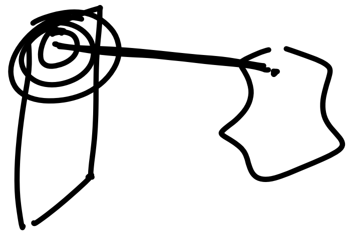
2) Frequency Wavenumber Algorithm (ω-k alg.)

3) Holographic reconstruction.

[Check the refs. in description].

7.2 Signal model.

$$S_b(x', y', k_t) = \iiint_{xyz} \frac{f(x,y,z)\, e^{-2jk_t \overbrace{|\vec{r}-\vec{r}'|}^{r}}}{16\pi^2 r^2}\, dx\,dy\,dz$$



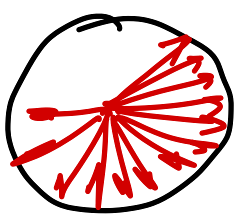
$$k_t = \frac{\omega_c + \beta t}{c} \; ; \; \beta = \text{chirp rate} = \frac{BW}{T_p}$$

$$t \in \left[-\frac{T_p}{2}, \frac{T_p}{2}\right]$$

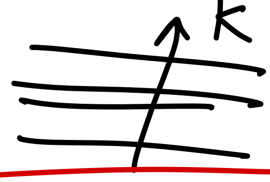
$$r = |\vec{r}-\vec{r}'| = \sqrt{(x-x')^2 + (y-y')^2 + (z-z')^2}, \quad z' \to 0$$

7.3 Holographic Reconstruction.

$$\frac{e^{-j2k_t r}}{r} = \frac{-j}{2\pi} \iint_{k_x' \, k_y'} \frac{e^{-jk_x'(x-x') - jk_y'(y-y') - jk_z z}}{k_z}\, dk_x' dk_y'$$



Weyl identity.



$$\vec{k}' = k_x'\hat{x} + k_y'\hat{y} + k_z\hat{z}$$

$$k_z = \sqrt{4k_t^2 - k_x'^2 - k_y'^2}$$

$$S_b(x',y',k_t) = \cancelto{1}{\frac{-j}{2\pi}} \iiint_{x\,y\,z} \frac{f(x,y,z)}{\cancelto{1}{16\pi^2}\,\cancelto{1}{y}} \iint_{k_x'\,k_y'} \frac{e^{-jk_x'(x-x') - jk_y'(y-y') - jk_z z}}{\cancelto{1}{k_z}} \, dk_x' \, dk_y' \, dx \, dy \, dz$$

$$= \underbrace{\iint_{k_x'\,k_y'} e^{jk_x'x' + jk_y'y'} \left[\underbrace{\iiint_{x\,y\,z} f(x,y,z)\, e^{-jk_x'x - jk_y'y - jk_z z} \, dx\,dy\,dz}_{FT(f(x,y,z))} \right] dk_x'\,dk_y'}_{FT^{-1}\ \text{of}\ \{FT(f(x,y,z))\}}$$

Assumption

Coord systems (x,y) is the same as (x',y').

$(k_x', k_y' \Leftrightarrow k_x, k_y)$

$$S_b(x',y',k_t) = FT_{2D}^{-1}\{FT_{3D}\{f(x,y,z)\}\}.$$

$$f(x,y,z) = FT_{3D}^{-1}\{\underbrace{FT_{2D}\{S_b(x',y',k_t)\}}\}$$

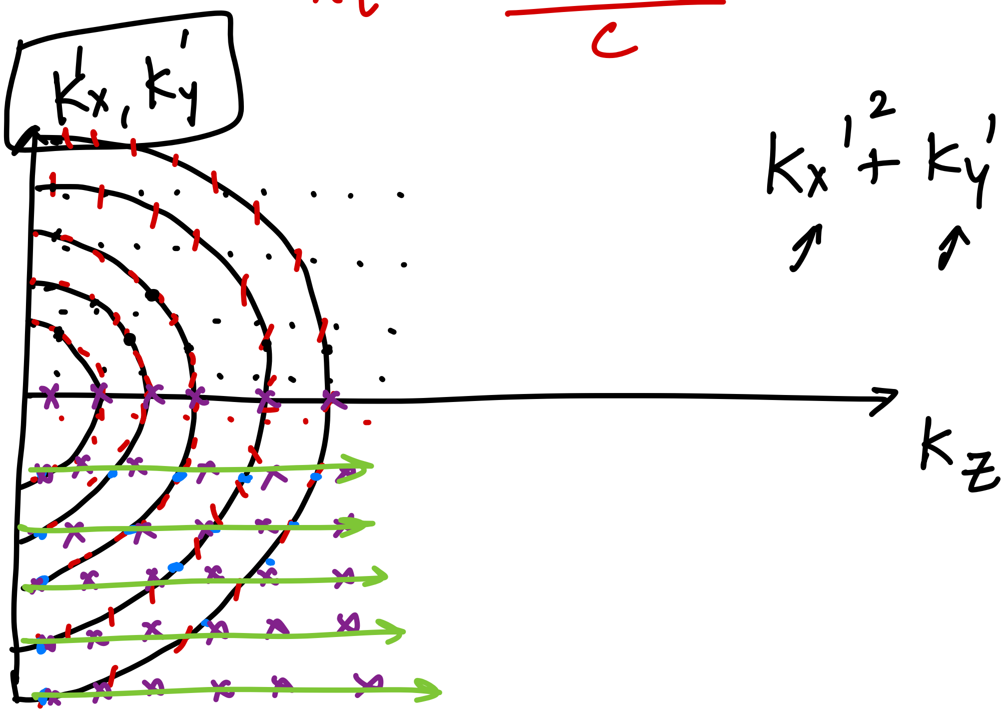
$$= FT_{3D}^{-1}\{FFT_{2D}\{S_b(x',y',k_t)\}\}$$

$$= FT_{3D}^{-1}\{\underline{\underline{S_B}}\underbrace{(k_x',k_y',k_t)}\}$$

$$\longrightarrow k_z = \sqrt{4k_t^2 - k_x'^2 - k_y'^2}$$

K_t is uniformly sampled because t is uniformly sampled.

$$K_t' = \frac{\omega_c + \beta t}{c}$$



$${k_x'}^2 + {k_y'}^2 + k_z^2 = 4k_t^2$$

Interpolate $S_B(k_x, k_y, K_t) \longrightarrow S_B(k_x, k_y, k_z)$

and then take 3-D IFFT.

Interpolation in K_z is called Stolt interpolation

$S_b(x', y', K_t)$

$\downarrow$ 2D FFT $\quad O(N^3 (\log N)^2)$

$S_B(k_x', k_y', K_t)$

$\downarrow$ Stolt interp. $\quad O(N^3)$

$S_B(K_x, k_y, k_z)$

$\downarrow$ 3D IFFT $\quad O(N^3 (\log N)^3)$

$f(x, y, z)$

$O(N^3 (\log N)^3)$

$251 \times 251 \times 2400$

$\times \log_2(251) \times \log_2(251)$ 40 secs.

$\times \log_2(2400) \approx 100$ sec

compared to 100-1000 days

7.4 Range Migration ($\omega - k$)

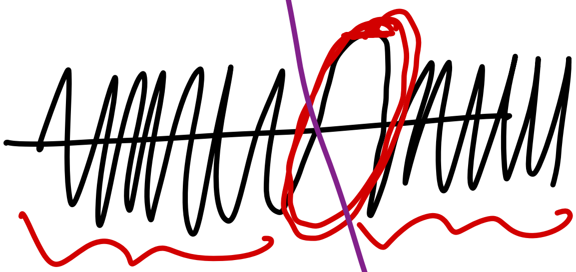
$$S_b(x',y',k_t) = \iiint_{x\,y\,z} \frac{f(x,y,z)\, e^{-2jk_t r}}{16\pi^2 r^2}\, dx\,dy\,dz.$$

2D FT on both sides w.r.t x', y'.

Recall $r = \sqrt{(x-x')^2 + (y-y')^2 + z^2}$.

$$S_B(k_x',k_y',k_t) = \iiint_{xyz} f(x,y,z) \underbrace{\iint_{x'y'} \frac{e^{-2jk_t r}}{16\pi^2 r^2}\, e^{-jk_x'x' - jk_y'y'}\, dx'dy'}_{FT}\, dx\,dy\,dz$$

$$\iint_{x'y'} \frac{e^{-2jk_t r}}{16\pi^2 r^2}\, e^{-jk_x'x' - jk_y'y'}\, dx'dy'$$



Method of stationary phase (MOSP)

$$= \frac{\pi j}{k_t z}\, e^{-jz \underbrace{\sqrt{4k_t^2 - k_x'^2 - k_y'^2}}_{k_z}}\, e^{-jk_x'x - jk_y'y}$$

$$= \frac{\pi j}{k_t z}\, e^{-jk_z z\, j k_x'x - jk_y'y}$$

$$S_B(k_x', k_y', k_t) = \underbrace{\iiint_{xyz} f(x,y,z)\,\frac{\pi j}{k_t z}\, e^{-jk_x'x - jk_y'y - jk_z z}\, dx\,dy\,dz}$$

$$(x,y) \Longleftrightarrow (x',y') \Rightarrow (k_x, k_y) = (k_x, k_y)$$

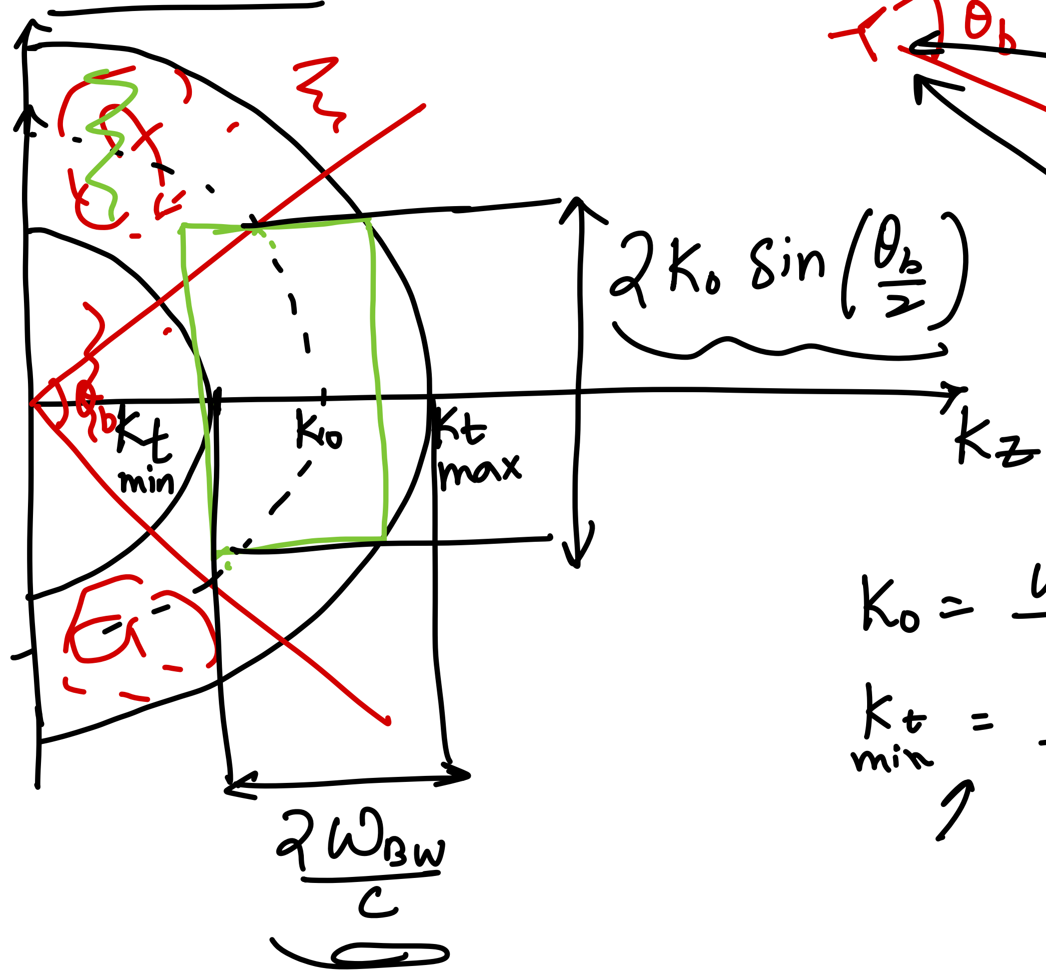
3D FT of $\dfrac{f(x,y,z)\cdot\pi j}{k_t z}$

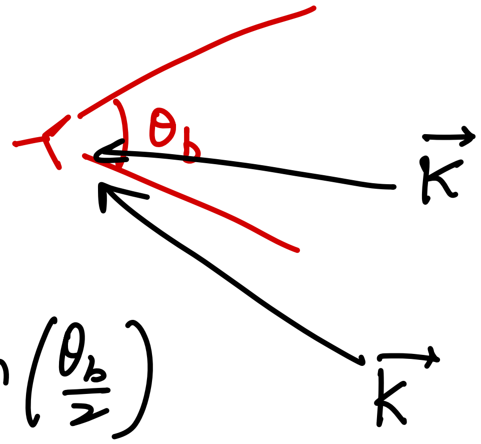
$$S_B(k_x', k_y', k_t) = FT_{3D}\left\{ f(x,y,z)\,\frac{\pi j}{k_t z} \right\}$$

$$f(x,y,z) = \frac{\cancel{k_t z}}{\pi j}\; \underbrace{FT_{3D}^{-1}}\left\{ \underbrace{FT_{2D}\left\{ S_b(x', y', k_t) \right\}} \right\}$$

1

7.5 Resolutions.





$$K_0 = \frac{\omega_c}{c}$$

$$k_{t\,min} = \frac{\omega_{min}}{c};\quad k_{t\,max} = \frac{\omega_{max}}{c}$$

$$\underline{\delta_{x,y}} = \frac{2\pi}{2k_0 \sin\left(\frac{\theta_b}{2}\right)} \approx \frac{\lambda_c}{4\sin\left(\frac{\theta_b}{2}\right)} \rightarrow \text{SAR}$$

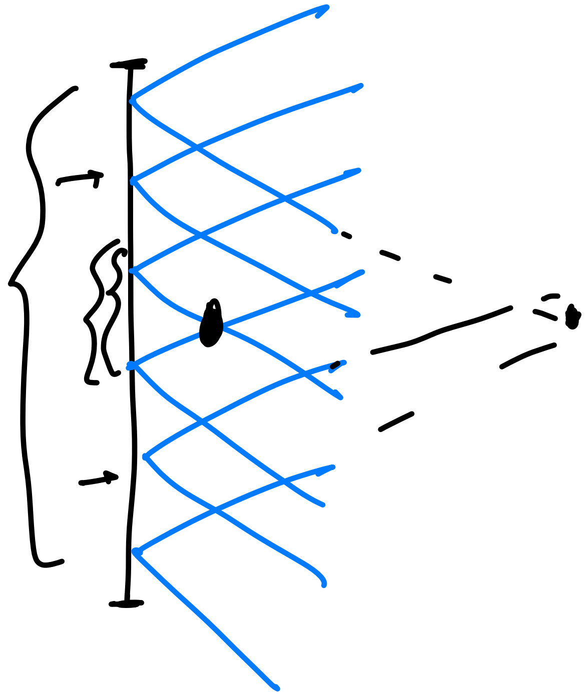
$$\delta_z = \frac{2\pi}{2\frac{\omega_{BW}}{c}} = \frac{c}{2f_{BW}} \longrightarrow \underline{\text{FMCW}}$$

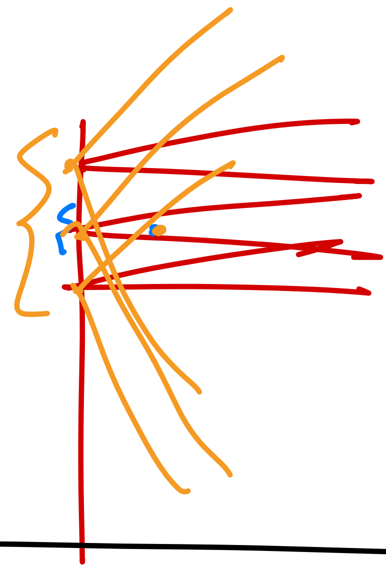
$\delta_{x,y}$ is independent of array size.

$\delta_{x,y}$ ↓

$\delta_{x,y} \propto \theta_b$

$\delta_{x,y} \propto \dfrac{1}{\theta_b}$





Range Migration Algorithm

Omega-K Algorithm

Holographic Reconstruction.

1,000,000 times FASTER!