Improving Segmented Processing for Interferometric Synthetic Aperture Radar via Presumming

K. Clint Slatton

EE381K: Multidimensional Digital Signal Processing

Professor: Dr. Brian Evans

November 30, 1998

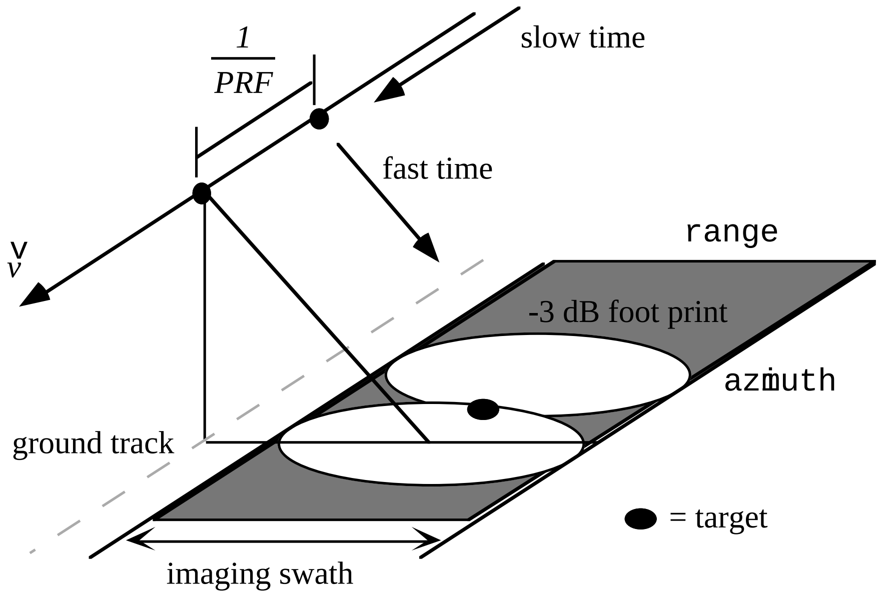
Outline

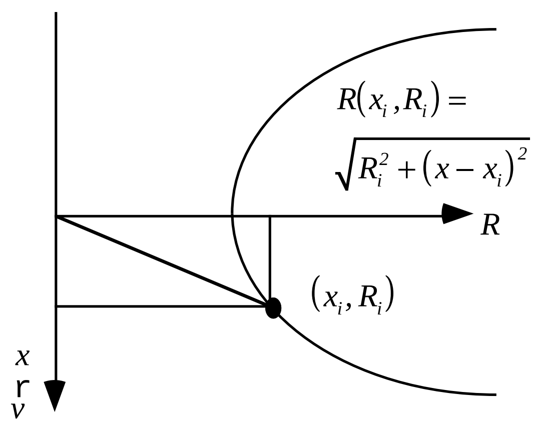
- Introduction
- Simulation results
- Processor architecture
- Implementation
- Results and conclusions to date

Introduction

- Interferometric Synthetic Aperture Radar (INSAR) data are needed to map Earth's topography
- Current approach
  - process the data in patches
    - keeps array sizes manageable and allows updating of motion and squint parameters in one scene
  - realign patches (deskew) during post-processing
- Periodic height errors of >1 m occur at the boundaries of these patches
  - unacceptable for mapping low-relief, flood-prone areas

SAR Imaging Geometry



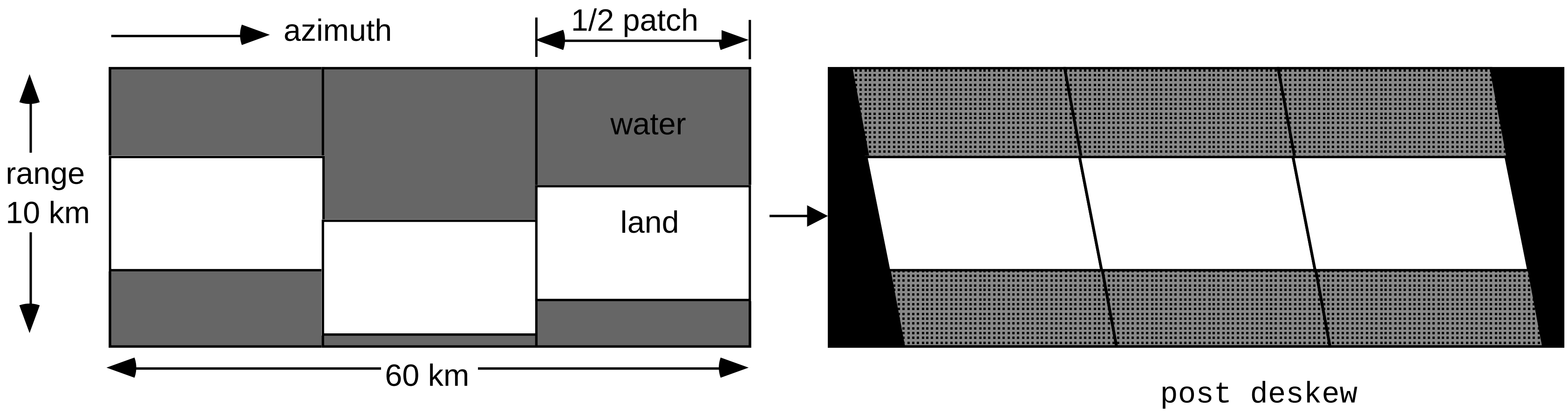


- Footprint of SAR beam pattern covers a swath as the SAR moves on a trajectory
- SAR emits pulses at the pulse repetition frequency (PRF) -> sampling frequency in azimuth
- Coherently sum reflected pulses to synthetically create linear antenna array

- Range to target varies
  - provides Doppler signature that determines proper phase offsets
- If SAR looking directly broadside, squint = 0

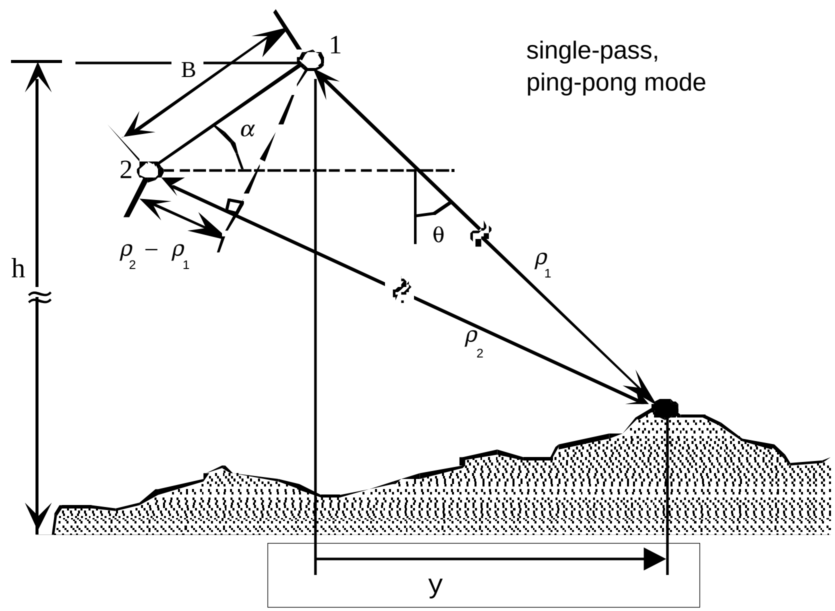
Deskewing

- Motion variations cause patches to be squinted differently
  - patches don't align after core processing
- Deskewing
  - non-zero squint -> zero Doppler frequency does not occur at closest approach
  - patches must be resampled using Doppler and range information
  - support region of deskewed data is a parallelogram
  - near-range pixels are shifted less than far-range pixels
  - data written out in half-patch sections to avoid data gaps
  - adequate for magnitude images, but not for INSAR phase images



INSAR Measurement

- Two antennas image the target area (assuming single pass mode) -> 2 complex images (C_1, C_2)
- Combine to get phase ϕ
- Phase is more sensitive to deskew than magnitude, so patch boundaries only a problem for INSAR



$C_1 = R_1 + jI_1$	$C_2 = R_2 + jI_2$
$A_1 = \sqrt{R_1^2 + I_1^2}$	$A_2 = \sqrt{R_2^2 + I_2^2}$
$\psi_1 = Tan^{-1}(I_1/R_1)$	$\psi_2 = Tan^{-1}(I_2/R_2)$

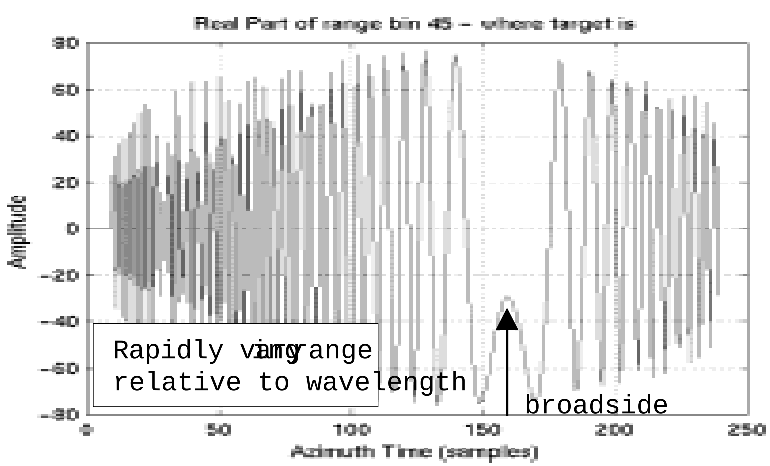
$$\angle C_1 C_2^* = \psi_1 - \psi_2 = \phi$$

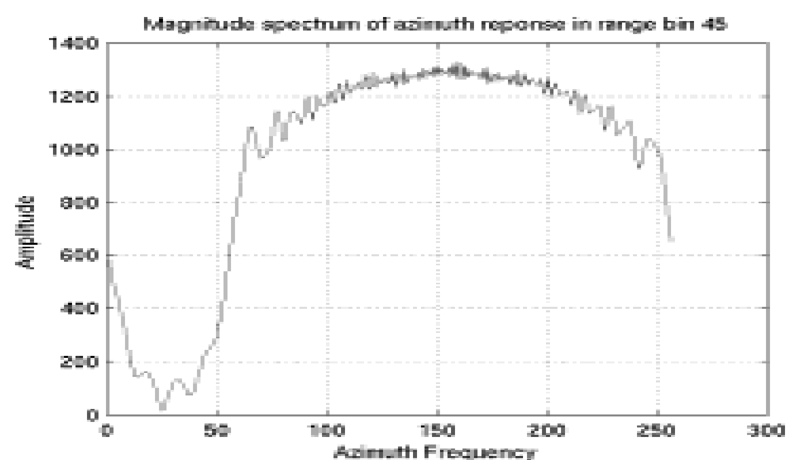
- Geometry relates ϕ to relative height z

$$sin(\theta - \alpha) = \frac{\rho_1^2 - \rho_2^2 + B^2}{2\rho_1 B} \qquad \theta = \alpha - Sin^{-1}\left(\frac{\lambda\phi}{2\pi 2B}\right) \qquad z = h - \rho_1 cos\,\theta$$

Point Target Simulation

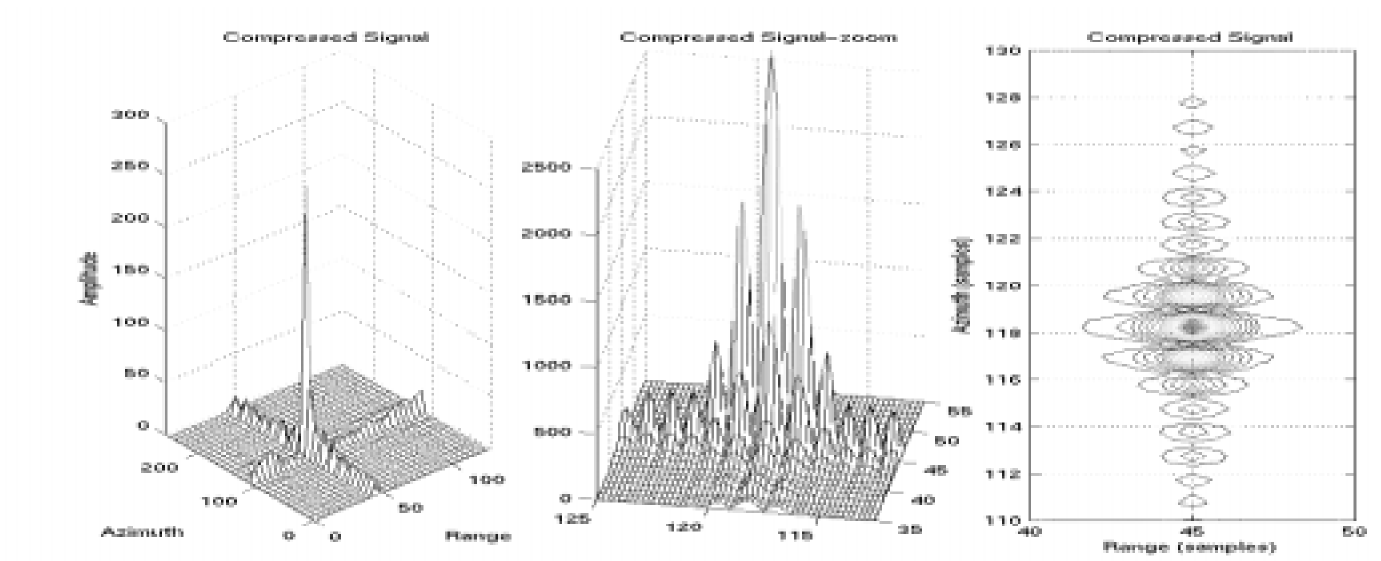
- Azimuth response
  - as SAR moves past target, many returned chirp pulses are collected
  - the return samples corresponding to a given target will consist of samples from these pulses
    - delayed according to the changing range to target
    - result is a new chirp, orthogonal to the range chirp in the data space
  - phase of azimuth spectrum varies rapidly
- Presumming
  - low pass filter, then downsample azimuth response
  - reduces patch boundaries by -> slowly varying phase





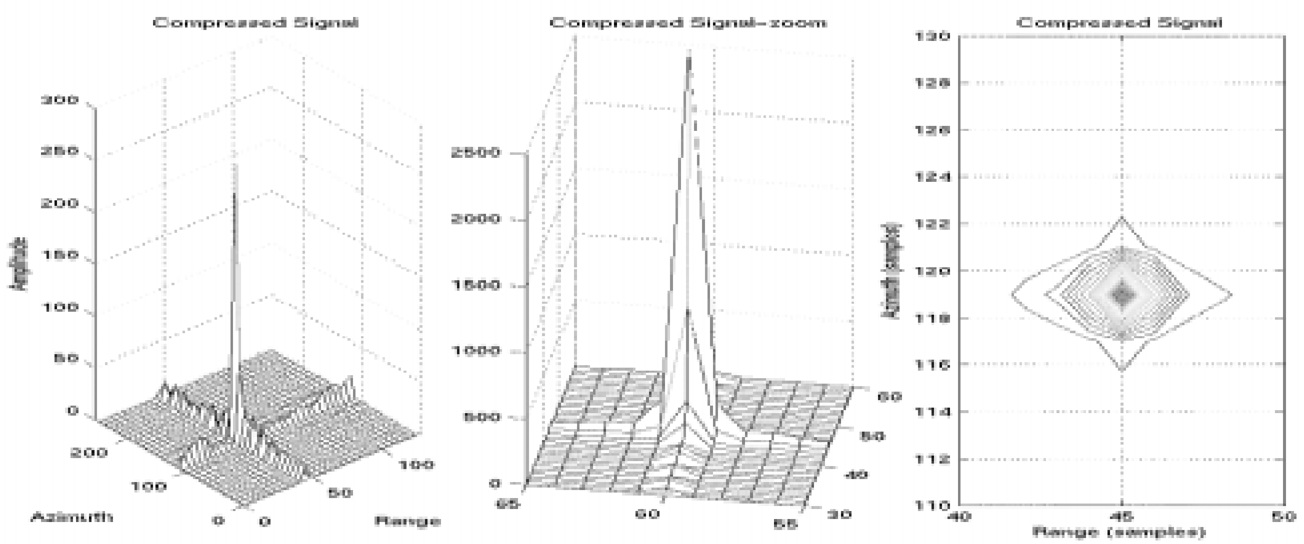
Point Target Simulation: Nominal

- After nominal azimuth compression
  - target is resolved, but significant sidelobes remain in azimuth direction
    - even after filtering azimuth reference function with a sidelobe reduction filter (kaiser)
  - PRF oversamples in azimuth relative to final posted resolution, so downsampling is acceptable



Point Target Simulation: Presummed

- Low pass filter and downsample the azimuth response
  - widens main lobe, reduces sidelobes, makes phase vary slowly
- Downsampling factor restricted to be integer
  - factor of 8 used in simulation to highlight effects
  - azimuth reference function not presummed -> defined for new azimuth response length
- Low pass filter for simulation was kaiser window for simplicity
  - β=3, 128 taps for point-wise multiplication



Nominal JPLIP Architecture: deskew

- Integer deskew program called immediately after core processing
  - done in spatial domain
  - for each patch
    - do loop over azimuth lines successively reads in 1-D arrays in range from 2-D pre-deskew image array
    - file pointer for this read is integer number of record lengths (uniform)
    - 1-D arrays reassembled into intermediate 2-D array
      - azimuth index in 2-D array depends on range bin via a do loop over range samples and Doppler (squint) values for current patch (non-uniform)
    - do loop over azimuth lines successively writes 1-D arrays from intermediate array, with azimuth index reset to start at 1
    - write 1-D arrays to 2-D post-deskew image array with file pointer equal to integer multiples of record length

JPLIP Architecture: Implement Presumming

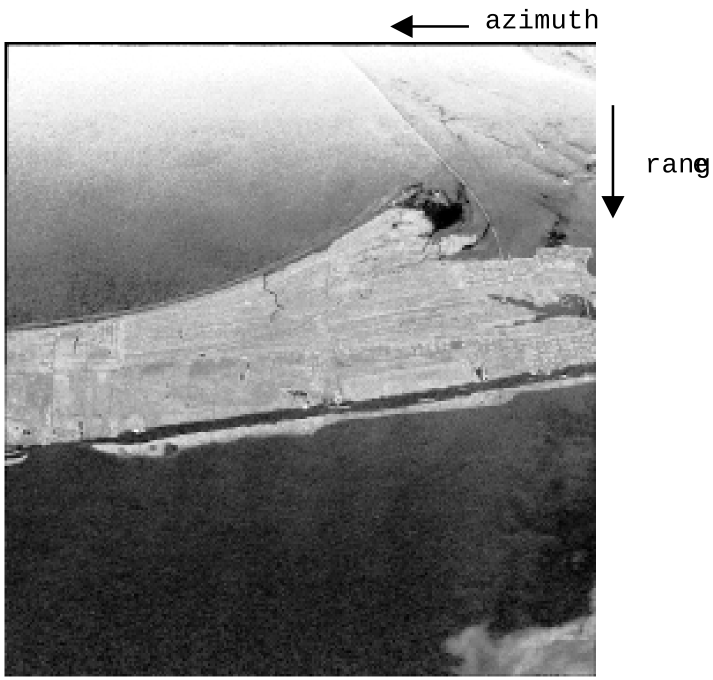
- Core processor
  - range compression
  - estimate Doppler frequency
    - calculate arrays for Doppler frequency as function of range
  - azimuth compression
    - inside this subroutine is where presumming is implemented
    - low pass filter (anti-aliasing filter)
      - in frequency-domain multiply azimuth response with 11-tap Parks-McClellan FIR filter
      - multiply with DFT{azimuth reference} and take inverse DFT
    - downsample azimuth response by D
      - take every D^{th} sample of spatial-domain azimuth-compressed signal
      - restricted to be an integer (D = 2)
    - write out pre-deskewed image array
- Deskew
  - unchanged

FIR: Finite Impulse Response

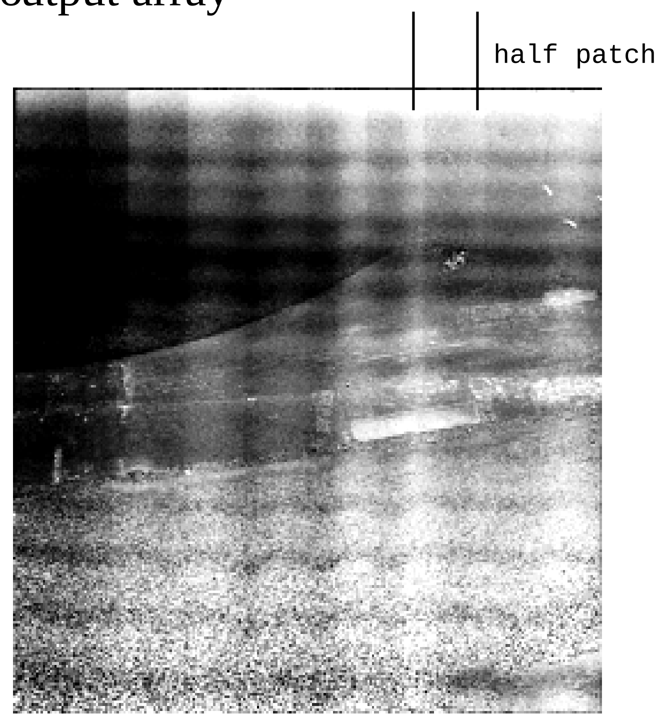
DFT: Discrete Fourier Transform

JPLIP Results (Nominal Case)

- Processed subset of SAR data acquired over Texas using the JPLIP processor
  - best data set to examine since large area of open water allows patch boundaries to be observed with no obscuring topographic signal
  - 10 km x 10 km scene took roughly 3 hours to process on HP-9000
  - both images are 1296 x 960, 32 bit (4 byte) floating point data = roughly 5 Mb
- Patch discontinuities in the topographic image are severe
  - occur where the patches are written to the output array

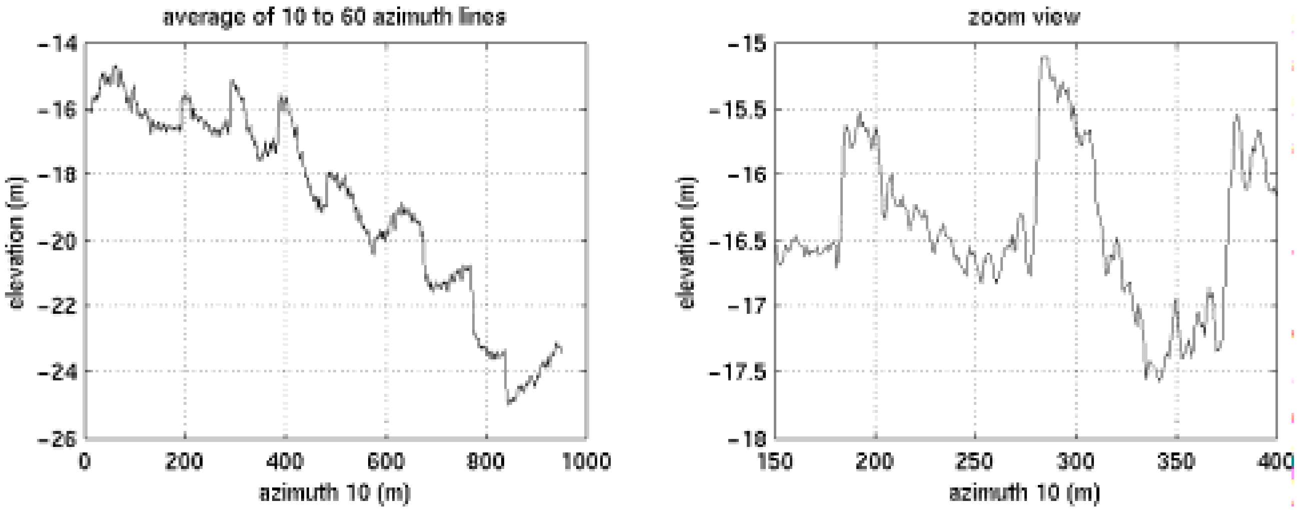


magnitude image



topographic image

Conclusions



- Extracted data transects show patch discontinuities clearly
- Presumming will improve azimuth response
  - sidelobes reduced
  - slowly-varying phase leads to better estimates of ψ_1, ψ_2 -> ϕ
    - less likely to have discrete jumps at patch boundaries
  - some discontinuities will remain due to imperfect motion compensation
- Proposed error metric
  - compute the difference in sample means of heights taken on either side of a boundary

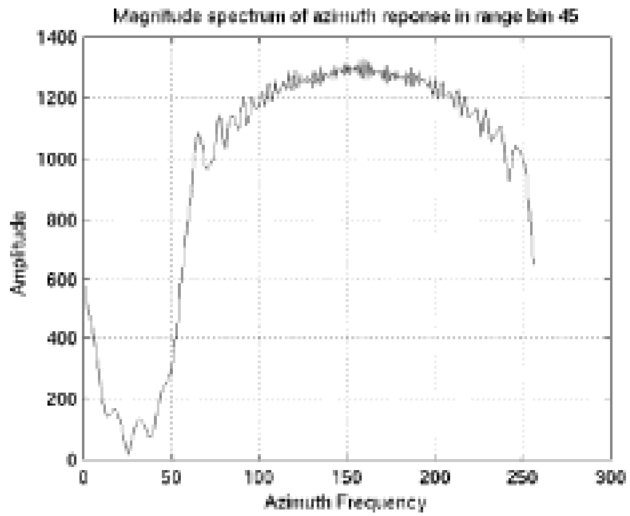
end presentation break

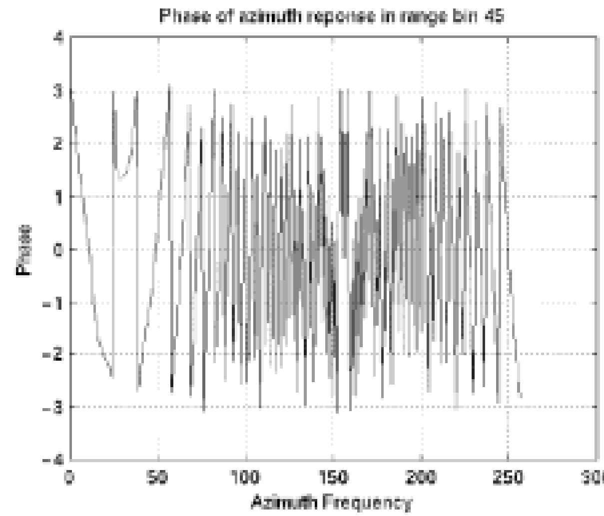
Nominal JPLIP Architecture: processing

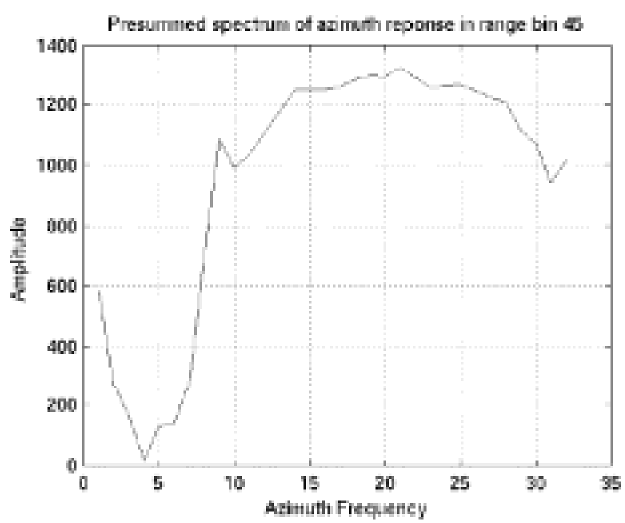
- Core range-Doppler processing done in one FORTRAN program
  - range compression
    - inverse DFT{DFT{pulse return}•DFT{range reference}}
  - azimuth compression
    - inverse DFT{DFT{1 patch of equi-range bin lines}•DFT{azimuth reference}}
    - while in frequency domain, fractional part of deskew is done
      - separate issue not involving patch boundaries

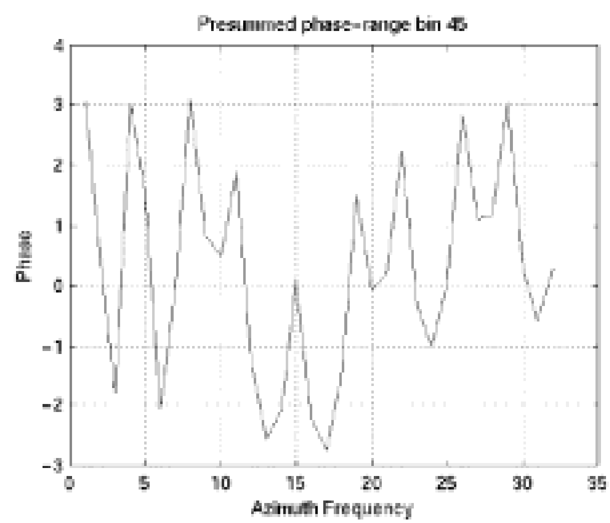
JPLIP: Jet Propulsion Laboratory Integrated Processor

Presumming's Effect on Phase





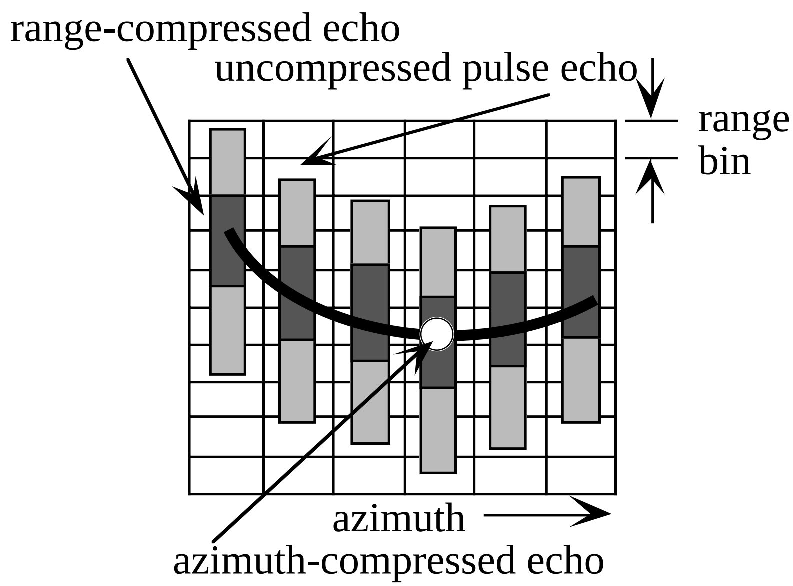




2D Uncompressed SAR Signal

- Transmit pulses *s(t)*:
  - windowed linear FM (chirp) measured in fast time
  - modulates a carrier
- Received signals *r(t)*:
  - attenuated, delayed version of *s(t)*
  - analog demodulated
  - sequence of *r(t)* signals modulated by Doppler response in slow time
- Convolve *r(t)* with *s(t)* to compress in range
- Convolve result with Doppler function to compress in azimuth

2D response of point target



[adapted from Morris and Harkness pg. 223]