Single Interferogram Processing

Piyush Agram
Jet Propulsion Laboratory

Jun 29, 2015
@UNAVCO

Thanks to my colleagues from JPL, Caltech, Stanford University and from all over the world for providing images and material for this talk.

Overview

- Detailed look at traditional workflows
  - Process2pass from ROI_PAC
  - insarApp from ISCE

- For each processing step
  - Implementation details
  - Important approximations
  - Shortcomings

- From single interferogram processing to stack processing

- Disclaimer: insarApp.py is a part of ISCE but does not represent all functionality available within ISCE. Hands-on exercises will demonstrate this.

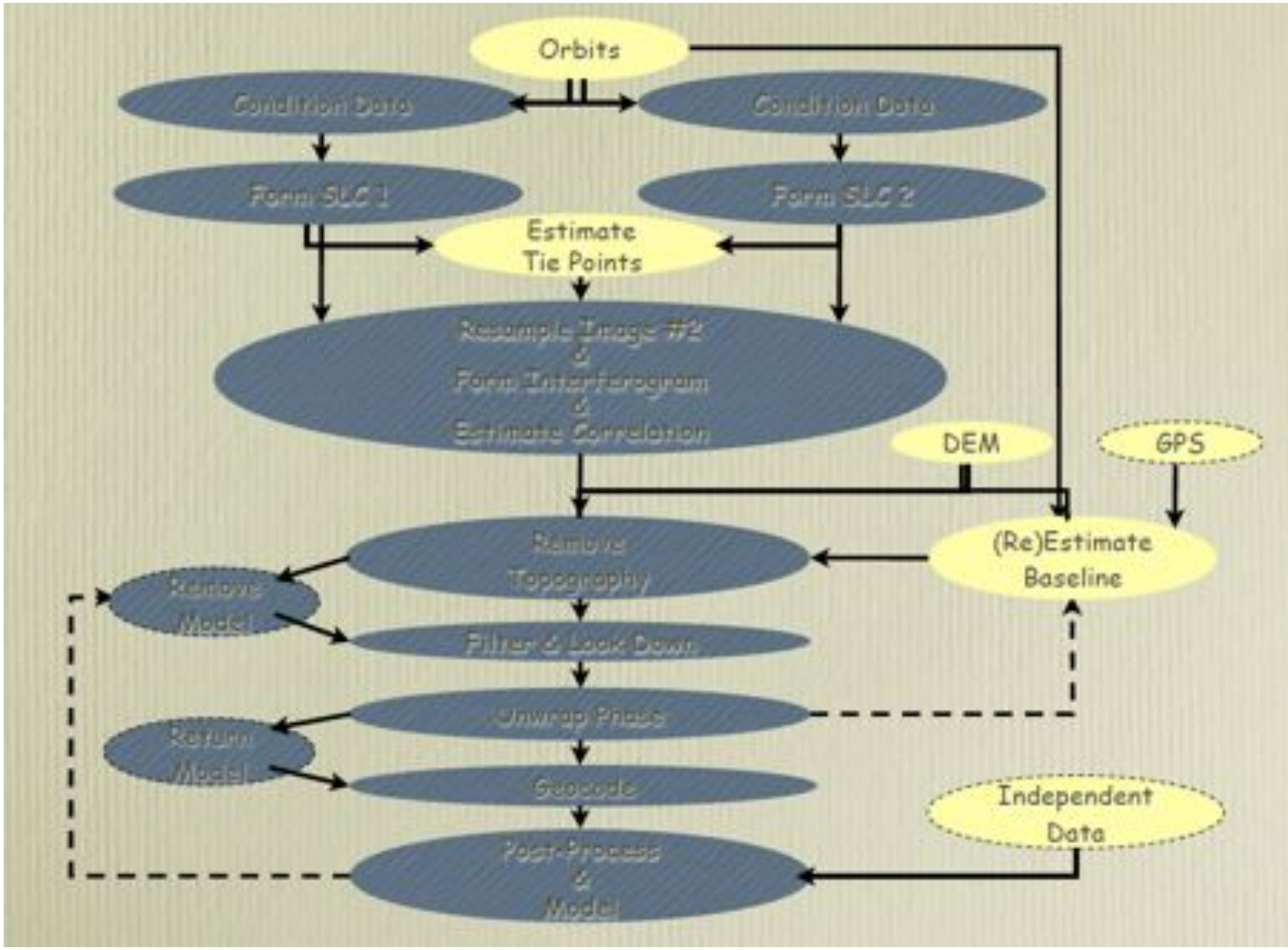
One size doesn't fit all

- Process2pass / insarApp are attempts at creating one workflow that suits the needs of most applications

- For better results / more rigorous science – need to understand what happens in these workflows better and make appropriate changes for your work

- SAR / InSAR – more rigorous you are, the better your results are

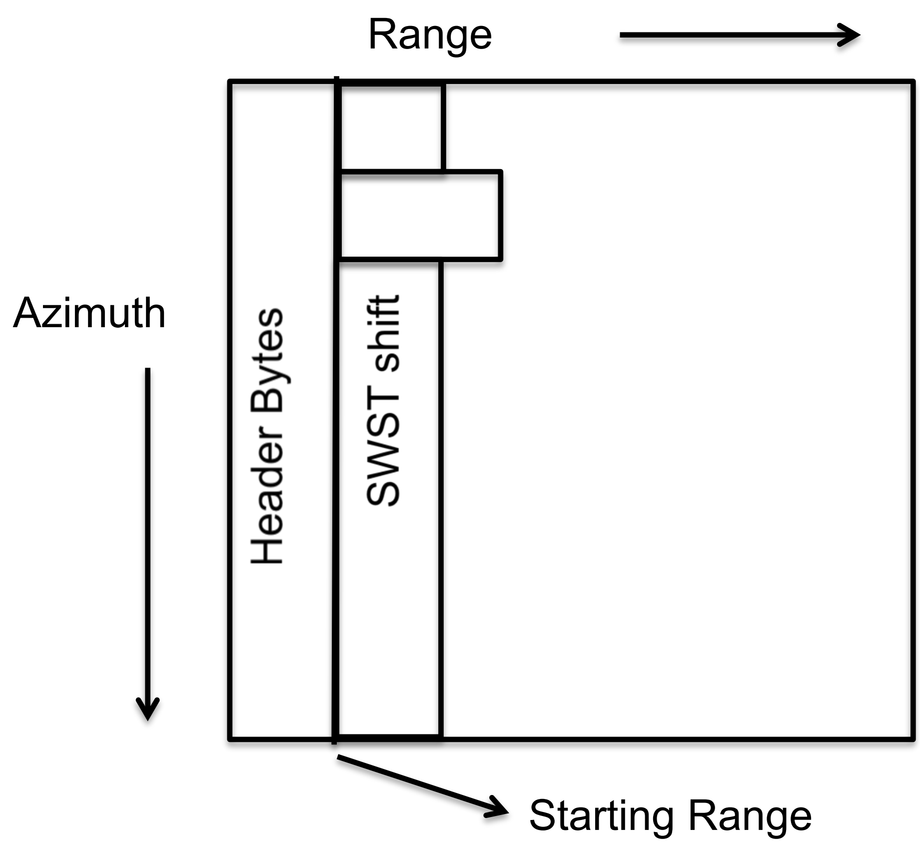
Process2pass.pl / insarApp.py



From: Fielding (2013/2014)

Step 0: make_raw / preprocess

- Unpacks data in formats provided by different missions and converts them into a standard internal format.
- Zero pads data to account for SWST shifts, if needed.
- Missing lines are filled with replicas / random numbers if needed.



<u>Metadata</u>

1. Starting range
2. Chirp slope
3. Orbit
4. Sensing Start Time
5. Wavelength
6. Antenna Length
7. Range Sampling Rate

<u>Metadata / Estimates</u>

1. I/Q means
2. Doppler centroid vs range

Imaging Mode

- Primarily designed for Stripmap data
  - constant PRF data
  - Starting from RAW data

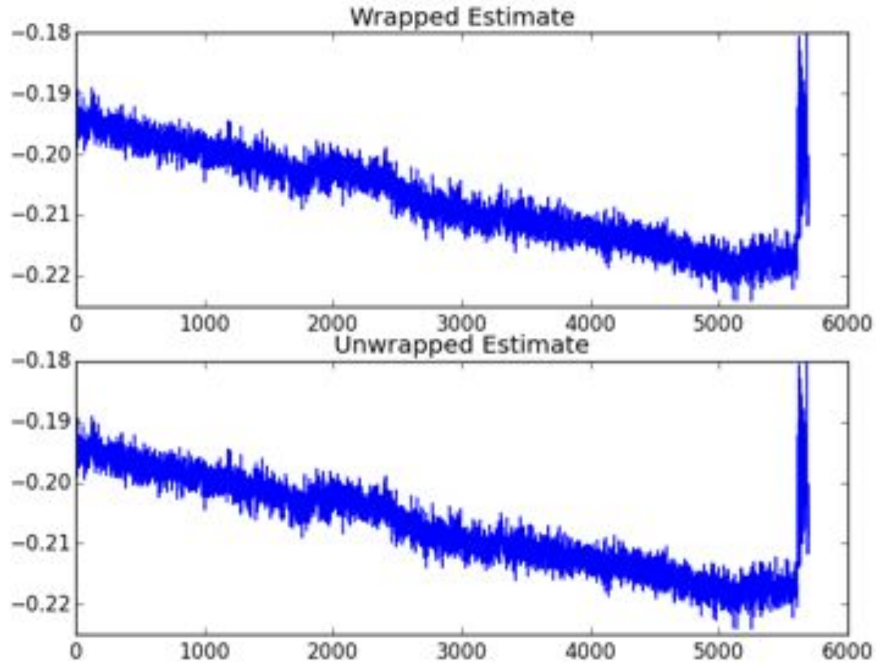
- Starting from pre-focused SLCs
  - An approximation (insarApp)

- Application to other imaging modes
  - Spotlight / SCANSAR
  - Sub-optimal processing

Doppler Centroid Estimation

- Line-by-line cross correlation method
  - Range compression before doppler estimation
  - -0.5 PRF < fd < 0.5 PRF
  - Quadratic / cubic function of range.

$$\arg\left(L_i \cdot L_{i+1}^*\right) = \frac{-2\pi f_d}{\mathrm{PRF}}$$



Example: Doppler estimate over Long Valley, CA - 2000716

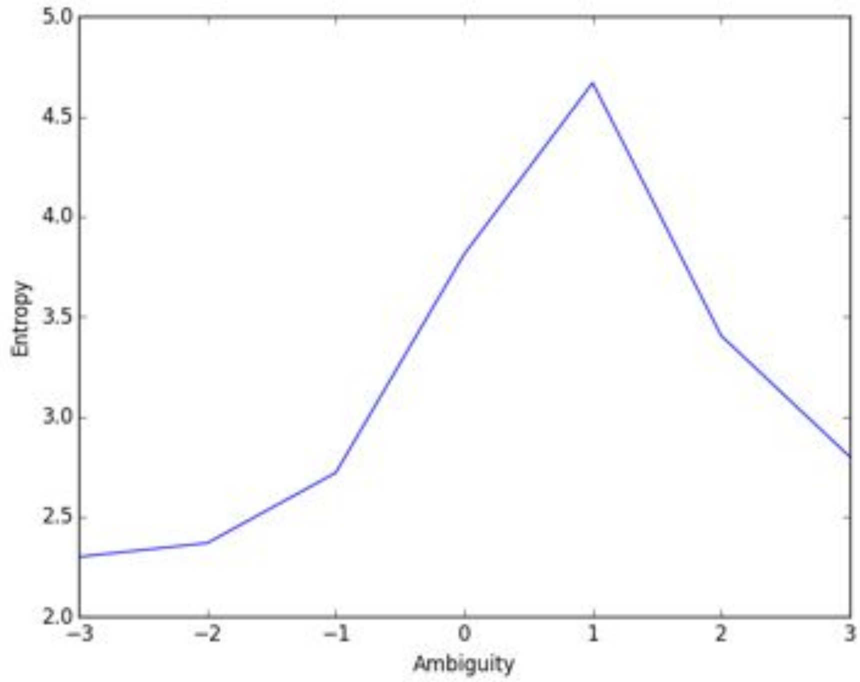
Doppler Ambiguity estimation

- Maximum entropy method

$$S = \frac{E\left(A^4\right)}{E\left(A^2\right)^2} - 1$$

- Works well for ERS-2 data post 2001.
- Estamb module in ISCE.

Example: Doppler ambiguity estimate over Long Valley, CA - 20070715



Comments on doppler model

- Error in doppler centroid results in slightly degraded point target response
- No azimuth variation considered
  - Implications for multi-frame processing
  - Most PS applications improve focusing by using azimuth-dependent doppler functions
- Doppler estimates from the data can be noisy over regions with low backscatter (water)
- Alternate methods can be used for estimating doppler ambiguity
  - Offset between sub-band SLCs

Doppler correction in ISCE

- Doppler centroid is not all due to squint
- Largest contributor to doppler is the vertical component of velocity
- Stdproc – focuses data to an ideal circular orbit
  - Effect of vertical velocity needs to be removed
  - Hence, doppler centroid is adjusted to take into account only for squint when applying motion compensation

Step 1: slcs/ formslc

- The most critical step in the workflow.
- Assumptions and parameters used during this stage, determines quality of end products
  - Resolution and bandwidths
  - Geometry system
  - Antenna patterns
- Geometry system determines the rest of the workflow
  - For pre-focused SLCs the rest of the workflow should match the geometry system used to generated the SLCs

Focusing in process2pass

- Processed to common doppler centroid quadratic/ cubic polynomial
  - Native doppler
  - Azimuth resolution decreased to account for common bandwidth only
  - Implications for multi-frame processing /PS analysis

- Constant velocity and actual orbit
  - Implications for multi-frame processing

- Antenna patterns not applied
  - Implications for amplitude-based PS analysis

Focusing in insarApp

- Processed to common doppler centroid (constant)
  - Native doppler
  - Azimuth resolution decreased to account for common bandwidth only
  - Implications for multi-frame processing / PS analysis

- Constant velocity but mocomp orbit
  - Implications for multi-frame processing

- Antenna patterns not applied
  - Implications for amplitude-based PS analysis

Pre-focused SLCs

- Delivered by mission as SLC products
  - Majority are delivered in Zero Doppler geometry
  - Except: ALOS-1 and UAVSAR

- Post-calibration corrections applied to data
  - Example: Range bias, Timing bias, Dry troposphere corrections
  - May not be available to users working with RAW data

- For best results, the rest of the interferometric processing should be done in the same geometry system as the SLCs.

Geometry systems

Actual Orbit
&
Native Doppler
(ROI_PAC, ALOS-1)

Actual Orbit
&
Zero doppler
(Most common format from currently active sensors)

Deskewed geometry

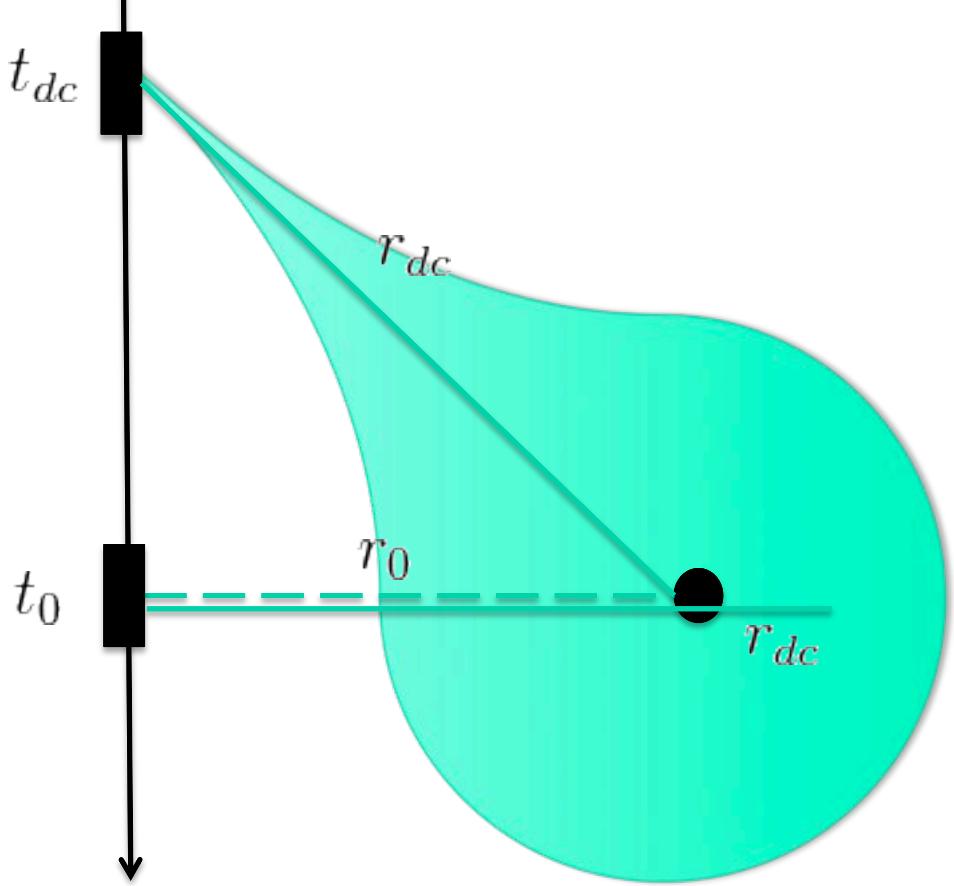
Mocomp ideal circular orbit
&
Native Doppler
(UAVSAR, Historic JPL format)

Mocomp ideal circular orbit
&
Zero Doppler

Supported by ISCE and tested

Native doppler vs Zero doppler

$$2\frac{\hat{v}\cdot\left(\vec{T}-\vec{S}\right)}{\lambda} = f_d\left(||\vec{T}-\vec{S}||\right)$$



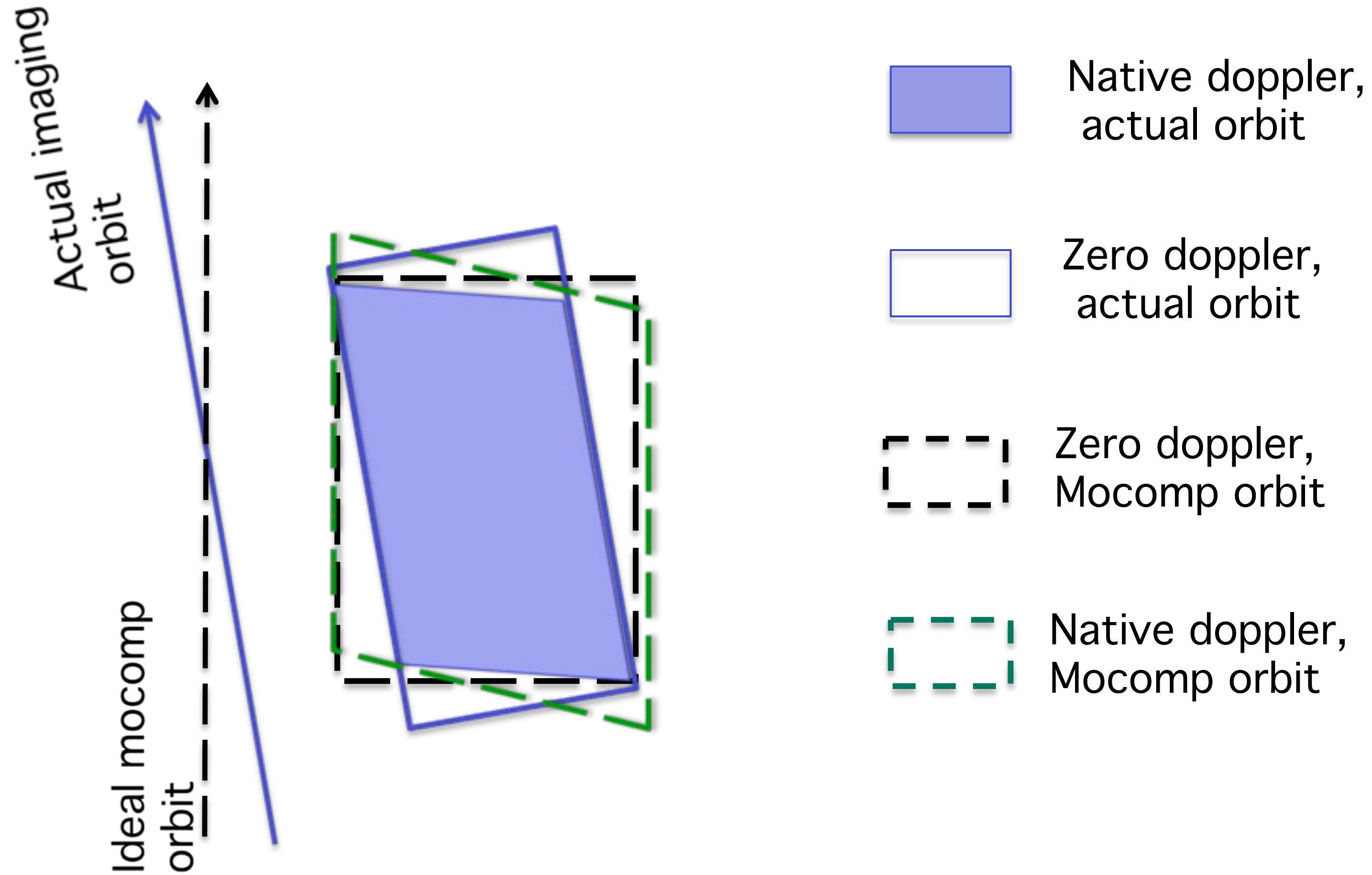
Different mappings for (line, pixel) to (azimuth time, range)

Native Doppler : (t_{dc}, r_{dc})
Target at center of imaging beam

Zero Doppler: (t_0, r_0)
Target at nearest distance to satellite track

Deskew: (t_0, r_{dc})
Hybrid coordinate system – cumbersome to do things accurately

Geometry regimes



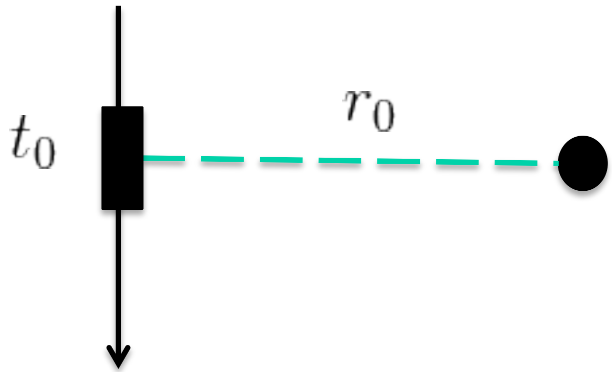
Doppler centroid and Azimuth carrier

$$\phi(t_{az}, r) \approx \exp(i2\pi f_d(r)\, t_{az})$$

- Native doppler:
  - Doppler centroid and geometry are coupled
  - Fixing one fixes the other
  - Great for InSAR / Bad for other wide area SAR applications

- Zero doppler:
  - For geometry computations, fd = 0
  - For resampling, the azimuth carrier on the signal is the same as native doppler.
  - Great for most SAR applications / Little extra work for InSAR

“Deskewed” geometry is not zero doppler geometry



H = 600km
V = 7 km /sec
F_d = 200 Hz
Wvl = 0.25m

Slant Range (R_0)	Along track shift	$R_{dc} - R_0$	w.r.t near range
800km	2857m	5.101m	0
900km	3214m	5.738m	64 cm

Geolocation shift is very small (sub meter) but is a function of range. Last column is error due to treating deskewed products as zero doppler products.

Deskewed geometry ...

- If the repeat pass occurs with zero geometric baseline
  - Same shift is seen on both master and slave
  - Has no impact

- However, this is almost never the case
  - Each interferogram affected by a ramp depending on the geometric baseline and doppler centroid.

- Might be ok for local deformation studies
  - Earthquakes, volcanoes, local subsidence etc
  - Not a recommended approach for wide area analysis

- OK for backscatter analysis or coherence analysis

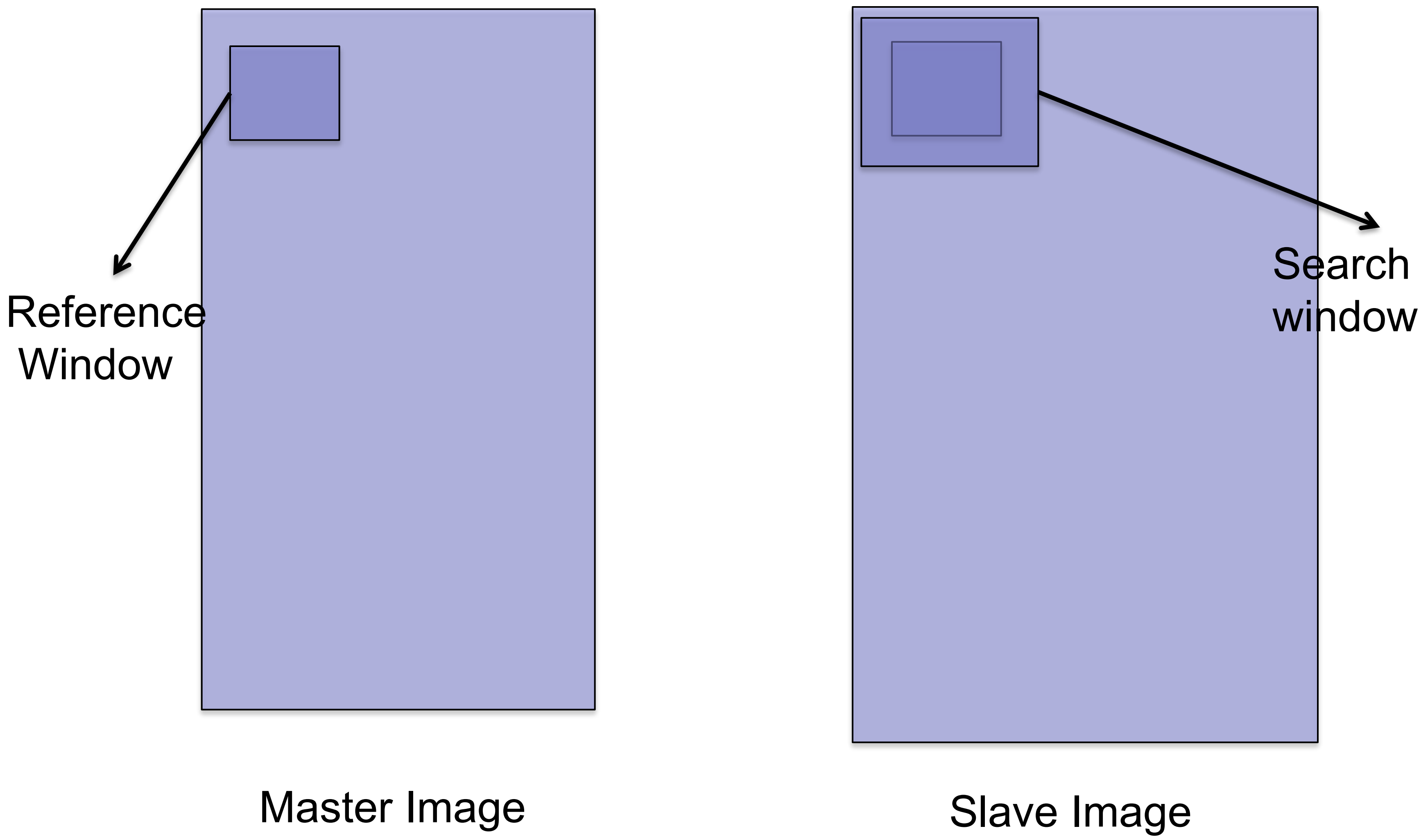
SLCs

- SLCs represent precise measurements of amplitude and phase
- When done right, everything from this stage onwards is geometry and bookkeeping
- SAR Focusing requires signal processing / EE background
- Everything else is pure geometry and trigonometry. No engineering background required.

Step 2: offsets / offsetprf

- Amplitude correlation between SLCs
- Agnostic to imaging geometry, except for estimating gross offsets
  - Currently, only native doppler
- Typical window size – 32 / 64
  - 40 windows in azimuth
  - 40 windows in range
  - Search area can be controlled by user
- Quality metrics
  - SNR and covariances also computed

Offsets



Offsets

- Different pixel spacings handled
  - PRFs in azimuth
  - Sampling rates in range

- Work with both complex and real data
  - Can be used for any generic image matching application

- ISCE includes multiple variants
  - Ampcor (time domain convolution)
  - EstimateOffsets (FFT-based convolution)
  - DenseAmpcor (process-based parallel ampcor)
  - EstimateOffsets (thread-based parallel estimateoffsets)
  - N-stage (Multiple stage ampcor for data with poor orbits, Radarsat-1 and some ERS)

Step 3: fitoff / offoutliers

- Iterative outlier removal from estimated offsets

- Simple affine model used to fit data
  - Points with residual > Nσ removed
  - N -> [10, 5, 3]

- SNR taken into account during affine fit

- Minimum number of valid offsets needed to proceed

- Can estimate polynomial fits and only use coefficients
  - But not the case

Step 4: resamp

- Offsets used to estimate range and azimuth polynomials
  - Range offset – 4^{th} order polynomial
  - Azimuth offset – 2^{nd} order polynomial
- Two step interpolation
  - (slave az, slave rg) -> (slave az, master rg)
  - (slave az, master rg) -> (master az, master rg)

$$\Delta x = f\left(x, y + \Delta y\right)$$

$$\Delta y = f\left(x, y\right)$$

Resamp ....

- Range spectral filtering using the offset polynomial (no topography)

- Hamming window applied to SLCs before cross multiplication.
  - Filtered SLC intensities stored in amplitude file

- Multi-looking applied before output

- Implications
  - resolution of interferogram
  - applications that use coherence

Baseline

- Process2pass / insarApp
  - Require data to be processed to common doppler
- Process2pass
  - Doppler not taken into account when estimating baselines
  - Quadratic baseline model
- insarApp
  - Line-by-line baselines
  - Should also keep track of individual mocomp baselines, i.e, actual orbit w.r.t mocomp orbit

Step 5a: flatorb

- In process2pass
  - Interferogram phase = flat_earth + topo
  - Height above best-fitting local sphere = 0
  - Flat_earth contribution removed before topo contribution is even estimated

- Not needed in insarApp
  - Interferogram phase = topo
  - Flat_earth correction included in formslc itself
  - Motion compensation performed when focusing the data on a range line–by–range line basis

Step 6: simulation / topo

- Project DEM into radar coordinates
  - Pixel-by-pixel height values
  - All input DEMs represent ellipsoid heights (and not geoid heights)
- Process2pass
  - For every DEM pixel, identify nearest range and azimuth pixel. Assign height to radar pixel.
  - Creates a hole-filled height image. Holes interpolated with Akima interpolation
  - Only the heights are used (Lat/Lon not used/saved).
  - Best performance when DEM pixels and radar pixels are approximately the same size.
  - Terser representation of the orbit used for simulations.
  - Doppler information used along with the master orbit.

Topo ...

- insarApp
  - Parallelized implementation
  - DEM is interpolated to generate pixel-by-pixel – lat, lon, heights
  - Biquintic interpolation used for continuity in range slope
  - Heights above best-fitting local sphere used for further processing.
  - State vectors are interpolated line-by-line for all geometry computations.
  - Doppler information used along with the common ideal mocomp orbit.

Step 7: simamp/ shadecpx2rg

- From DEM in radar coordinates simulate and amplitude image

- Process2pass $$\frac{1}{\tan\theta_{inc}}$$

- insarApp $$z(t, r+\Delta r) - z(t, r)$$

- Some representation of local range slope, projected into radar LOS

Step 8: synth_offset / rgoffset

- Estimate offsets between simulated amplitude and actual amplitude

- Shifts due to
  - Wrong timing information
  - Slant range changes (troposphere, delays etc)
  - Improper focusing

- Same offset code options available – ampcor, estimateOffsets etc.

- Ideally, zero offset desired.

Step 9: sim_removal/ resamp_only

- Align the interferogram and the DEM
- process2pass:
  - Interferogram is the reference
  - Height map is resampled to match the interferogram
  - Defeats the purpose of precise orbit information
- insarApp:
  - DEM is the reference
  - Interferogram is resampled to match the DEM
  - Precise orbit information given more importance

Step 10: done_sim_removal/ correct

- Uses the DEM to remove topo contribution

- Process2pass
  - Inputs:
    - Simulated DEM
    - Baselines estimated during orbbase
    - Interferogram
    - Doppler information
  - Outputs:
    - Flattened interferogram
    - Topography phase

Correct ....

- insarApp
  - Inputs:
    - Simulated DEM
    - Baselines estimated during mocompbaseline
    - Individual scene baselines w.r.t ideal mocomp orbit
    - Interferogram
    - Doppler information
  - Outputs:
    - Flattened interferogram
    - Topography phase

Step 11: Coherence

- Multi-looked/ single-look interferogram and amplitude files are used as inputs
  - Simple triangular window (3 x 3)

- Process2pass
  - Computed at single look. So, single look interferogram and amplitude are needed.
  - Computed with flat_earth removed but topography still included
  - Coherence product is multi-looked later
  - No need for resampling after alignment with DEMs

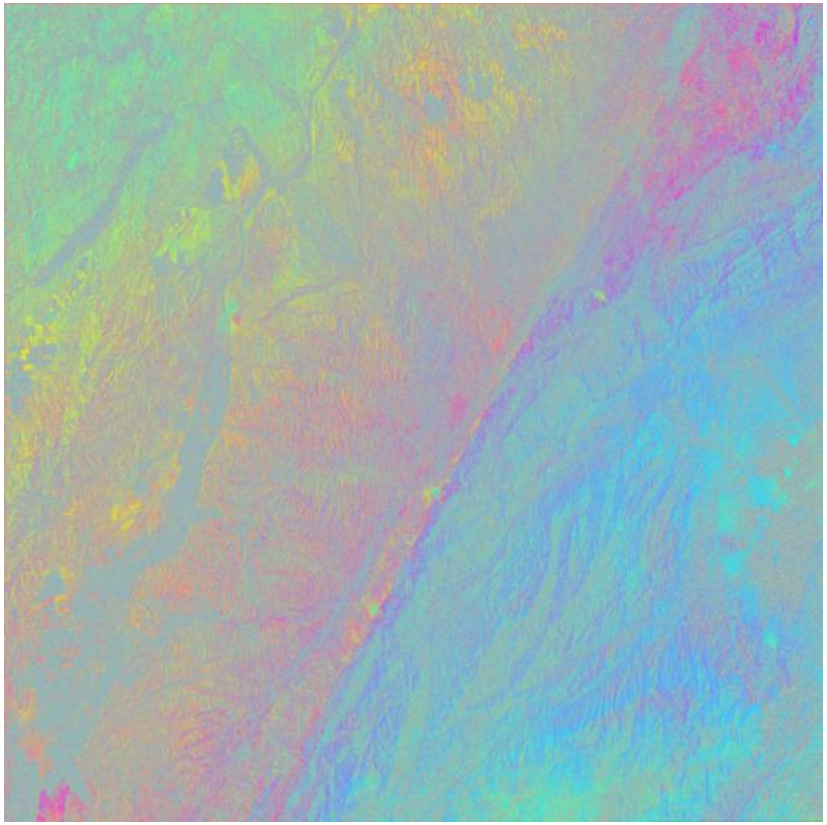
Coherence ....

- insarApp
  - Computed directly at multi-look level
  - Computed with both flat_earth and topography removed
  - To avoid resampling after alignment with DEM; computed with resampled interferogram and amplitude files
  - More efficient disk space and time-wise, but not as rigorous for coherence based applications
  - insarApp modules can be re-ordered to generate same set of products as process2pass

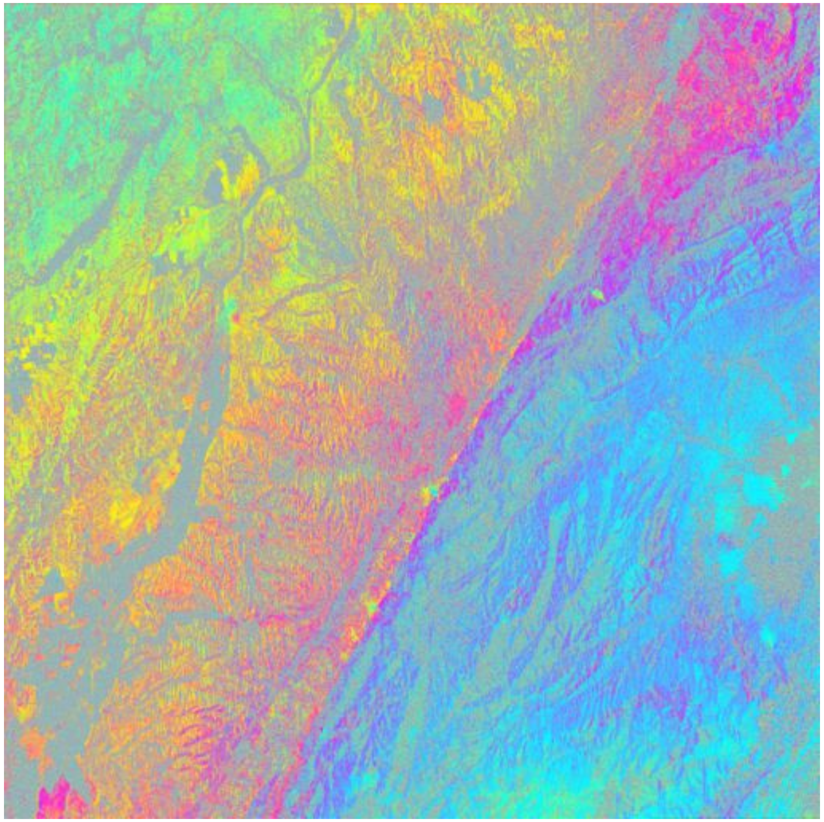
Step 12: Filtering

- Cleans up noise in InSAR phase
  - Helps phase unwrapping
- Process2pass
  - Low pass filter / Goldstein filter
- insarApp
  - Goldstein filter
- Reduces resolution significantly
  - Filtered product has resolution 2-4 times less than the original product

Filtering example



Unfiltered interferogram

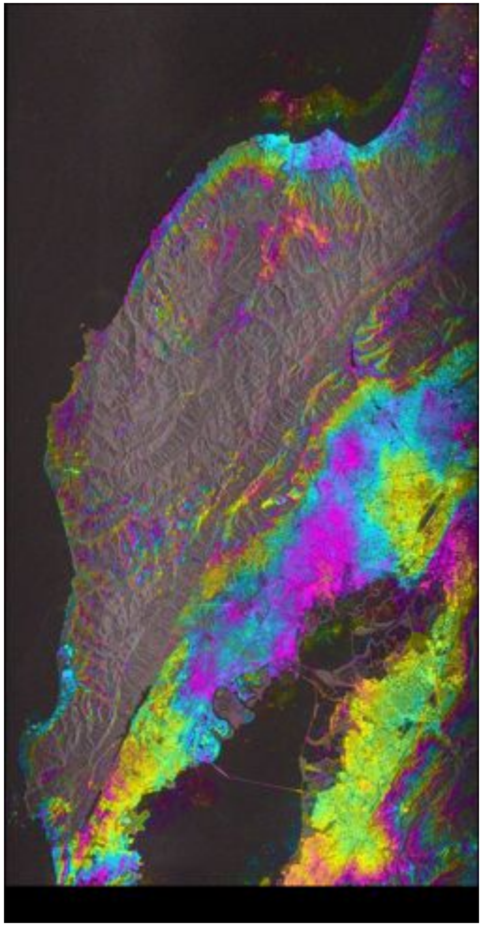


Filtered interferogram
Goldstein (0.4)

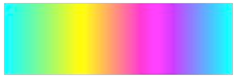
Step 13: Phase unwrapping

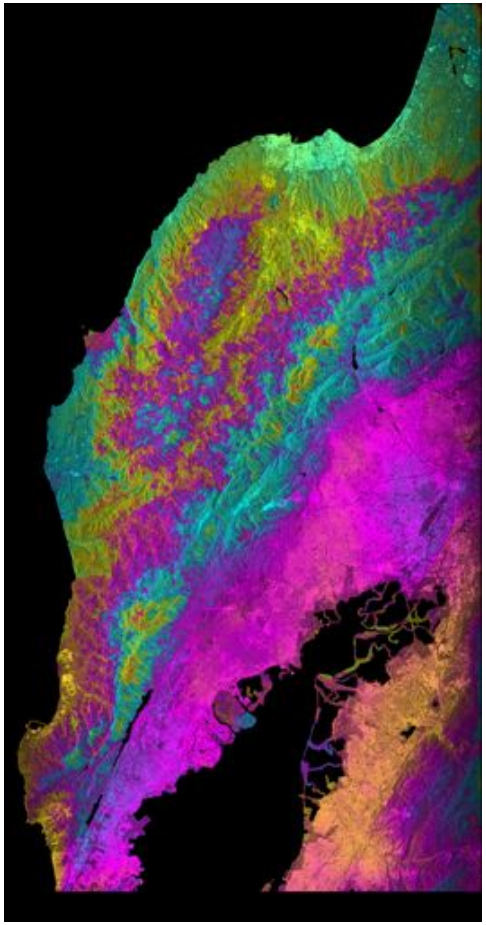
- Process2pass / insarApp
  - Grass, icu, snaphu, snaphu_mcf
- Performance depends on the dataset
  - No universal solution yet
- Grass / icu unwrap a single component
- Connected components are really important for interpretation
  - Largely ignored by novice users

Connected components

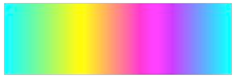


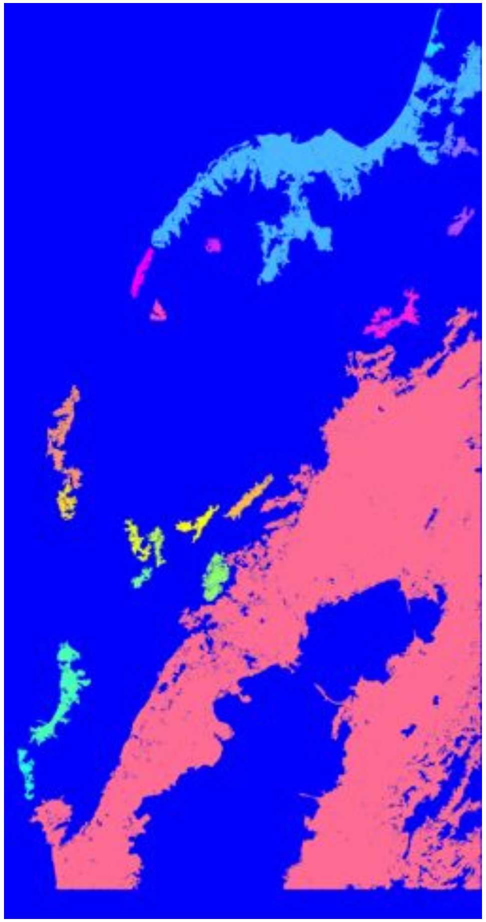
Wrapped Interferogram

-π  π



Unwrapped Interferogram

-12.5  12.5



Connected components
(Self-consistently
unwrapped regions)

Connected components

- For maximum spatial coverage
  - Two stage unwrapping is unavoidable
- Can be done by hand or visual inspection
  - For few connected components
- Simple sparse 2D unwrapping using only few points in connected components
- Never use SNAPHU output without taking connected components into account

Step 14: Geocoding

- Radar geometry to map coordinates

- Multiple interpolation options
  - Sinc, bilinear, bicubic, nearest neighbor

- Cautious when interpolating complex data
  - SLCs have azimuth/ range carriers
  - Interferograms are typically baseband data

- Adjustment of DEM / interferogram for alignment (synth_offset/rgoffset) guarantees that geocoding will be suboptimal

Geocoding....

- Process2pass
  - Lookup table for lat/lon grid to radar pixels computed for certain number of looks
  - Lookup table needs to be regenerated for different look combinations
  - Not generalized for all types of images

- insarApp
  - Generalized, parallelized implementation
  - Does not generate a lookup table
  - Output spacing controlled by inputDEM
  - upsampleDEM.py provided if users desired higher resolution geocoded products

Geocoding …

- Alternate method for geocoding
  - Easier for higher resolution products
  - Pixel-by-pixel lat/lon generated at topo step
  - Use gdal VRT to geocode any data on the same grid as the multi-looked interferogram

- Isce2gis.py vrt –I product.bin –lat lat.rdr –lon lon.rdr

- Gdal_translate / gdalwarp using product.bin.vrt