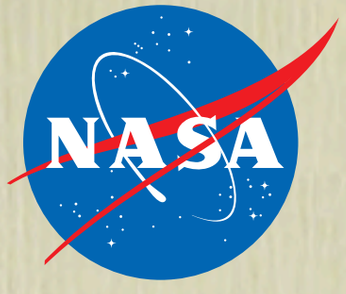
National Aeronautics and
Space Administration



ROI_pac Internals

Under the Hood

Eric Fielding
Jet Propulsion Laboratory, California Inst. of Tech.

UNAVCO InSAR Short Course
August 17-19, 2009

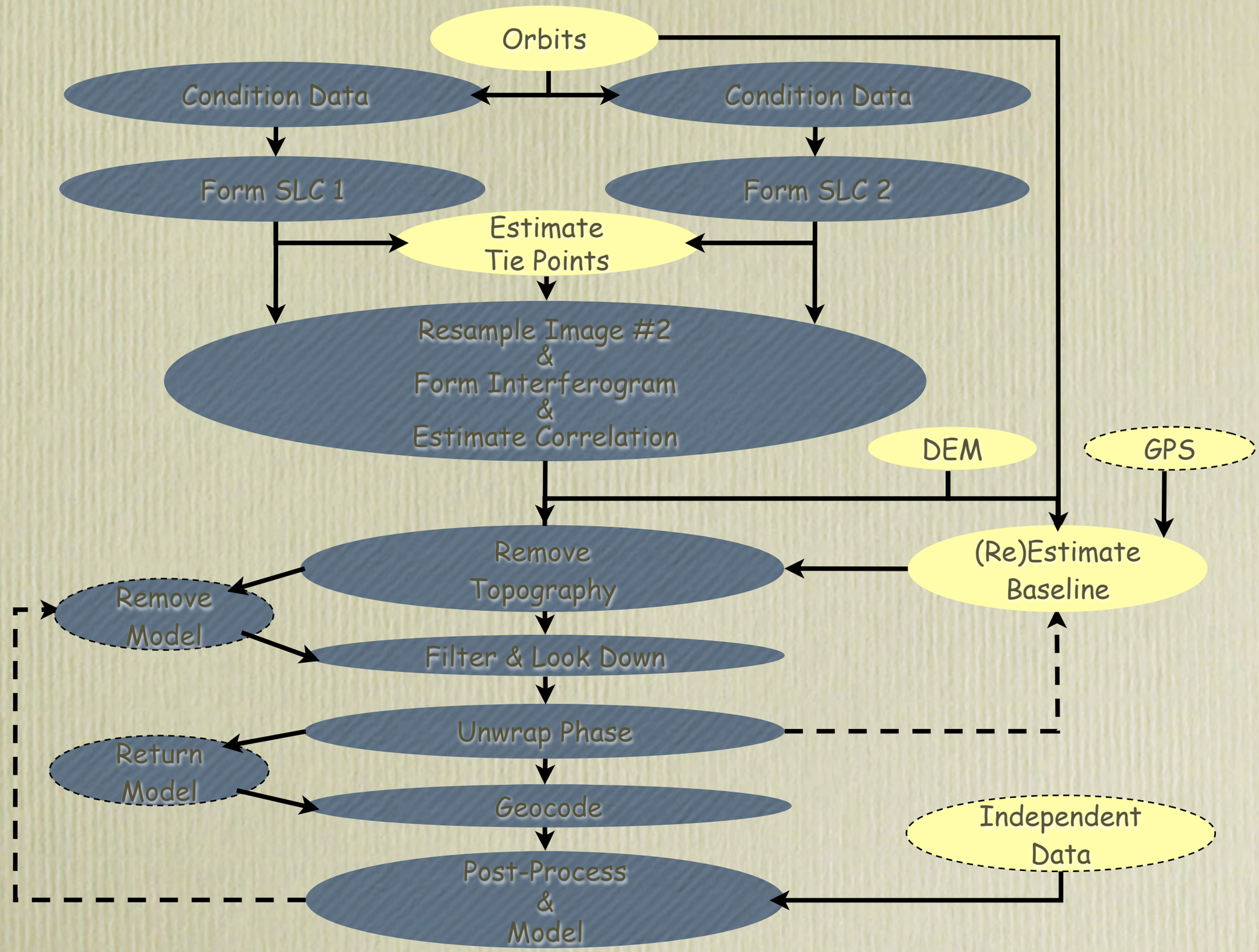
Processing Setup

- typical directory structure:
  - main directory for a given track (e.g., d170)
  - directory for topo data (DEMs), need at least 90 m resolution, 30 m is better
  - below main directory create a directory for each date (e.g., 930110) or orbit (e.g., O25557)
  - need satellite orbit files, usually in central directory

ROI_pac operation

- Obtain raw data
- Data ingestion “make_raw”
- SAR processing, interferogram formation, etc. “process_2pass”
- Post-processing done outside of ROI_pac, “mdx”, Matlab or GMT programs

ROI_pac Two-pass Processing Flow



modified from Mark Simons

File Types

- ROI_pac uses a number of file types as both input and output files. Most of the output files created have a standard set of suffices to indicate the type.

- Output file types:

- "file.type".rsc = ASCII text file containing metadata about "file.type"

- “file”.rsc.hst = history of reads and writes to “file”.rsc

- "file".rsc = ASCII text file with other metadata such as baseline for interferogram pair or reformatted orbit data

File Types

- Output file types (cont'd):
- *.raw = binary file of raw data, I,Q 1-byte integer values for each echo sample
- *.int = complex real 8-byte binary file containing real and imaginary parts of the interferogram (can also be read as real or float values band interleaved by pixel)
- *.amp = binary file with amplitudes of the two SAR images used to form an interferogram, real 4-byte values band interleaved by pixel (can also be read as complex 8-byte values)

File Types

- Output file types (cont'd):
- *.slc = complex real 8-byte binary file containing real and imaginary parts of the single-look complex (SLC) image (and multilooked versions of the SLC)
- *.cor = binary file with average amplitude of SAR images used to form an interferogram and correlation measure of coherence, real 4-byte values band interleaved by line (BIL or "rmg")

File Types

- Output file types (cont'd):
- *.off = ASCII text file of offsets measured between two images
- *.out = ASCII text informational output of a program
- *.hgt = binary file with simulated SAR amplitude image and elevation in radar coordinates, real 4-byte values band interleaved by line (BIL or "rmg")

File Types

- Output file types (cont'd):
- *.aff = ASCII text file of affine transformation to map simulated image to actual SAR image
- *.unw = binary file with SAR amplitude image and unwrapped phase, real 4-byte values band interleaved by line (BIL or "rmg")
- *.flg = binary file with flags used and resulting from unwrapping with standard unwrapper, 1-byte values with flags set in bits

File Types

- Output file types (cont'd):
- *.msk = binary file with SAR amplitude image and coherence with zeros in masked out areas, real 4-byte values band interleaved by line (BIL or "rmg")
- *.trans = binary file with inverse mapping transformation from SAR to DEM coordinates, two bands are range and azimuth pixel locations of SAR for each DEM pixel, real 4-byte values band interleaved by line (BIL or "rmg")

File Types

- Input file types:
- *.proc = ASCII text file specifying parameters used in processing (suffix is allowed to be different on int.proc)
- *.dem = binary file with digital elevation model in lat-long or UTM coordinates, signed integer 2-byte values (in meters unless scaling specified)
- *.in = ASCII text file created by ROI_pac scripts used as input for a compiled program

SAR satellites

	repeat cycle (days)	wave-length (cm)
European ERS-1/ERS-2 1992-2001(+)	35 (1,3,183)	6
Canadian Radarsat-1 1995-present	24	6
European Envisat 2003-present	35	6
Japanese ALOS launched Jan. 2006	46	24
German TerraSAR-X launched July 2007	11	3
Italian COSMO/SkyMed 2 launched 2007	16/2	3
Canadian Radarsat-2 launched Dec. 2007	24	6

SAR satellites processing

	Modes	ROI_pac proc.
ERS-1/ERS-2	1 strip map	yes raw
Radarsat-1	7 standard strip 5 fine strip ScanSAR	strip: yes CEOS, soon STF ScanSAR: no
Envisat	7 standard strip (IM) ScanSAR (WS)	strip: yes raw ScanSAR: soon
ALOS	strip FBS, FBD, POL ScanSAR	strip: yes raw ScanSAR: maybe
TerraSAR-X	strip ScanSAR Spotlight	strip: soon SLC ScanSAR: maybe? Spotlight: soonish SLC
COSMO/SkyMed	strip, ScanSAR, Spotlight	strip: probably SLC
Radarsat-2	strip (4 types), ScanSAR, Spotlight	strip: maybe SLC

other SAR spacecraft

	repeat cycle (days)	wave-length (cm)
US SeaSat 1978 (110 days) strip: ?	17/3	24
Japanese JERS-1 1992-1998 strip: yes raw (except ASF)	44	24
US Shuttle Imaging Radar–C (SIR-C) 1994 (2x 10-day flights) strip: yes raw (reformatted signal data) & interferometric SLC, requires extra work	1 day, 6 months	3, 6, 24

Data Ingestion

- Different ingestion programs for each satellite type
- ERS-1 and ERS-2: use “make_raw.pl”
- Envisat: use “make_raw_envi.pl”
- ALOS: use “make_raw_alos.pl”
- JERS-1: use “make_raw_jers.pl”
- Radarsat-1 has two data products from ASF
  - CEOS: use “make_raw_RSAT-CEOS.pl”*
  - STF: use “make_raw_RADARSAT_swath.pl”†

* from roipac.org † needs a small amount of work

Data Ingestion (cont.)

- name for scene (“date”) must match directory name
- output of make_raw:
  - $date.raw–unpacked raw data 8 bits I,Q for each sample, one line per echo record, ERS has 412 bytes extra at start of lines
  - $date.raw.rsc–metadata for raw data
  - hdr_data_points_$date.rsc–orbit data reformatted
  - other intermediate files, including dop.unw–output of programs to measure Doppler centroid of data (fit put into .raw.rsc)

Data Ingestion (contd.)

- "make_raw.pl" for ERS SAR data
- move ERS-1 and ERS-2 data into scene directory, each frame in "CEOS" format has three files:
  - data: IMAGERY19930110182527 39T1Of3
  - leader: SARLEADER19930110182527 39T1Of1
  - volume directory: VDF19930110182527 39T1
- make_raw.pl can concatenate many frames from a directory, uses IMAGERY* names to determine order
- use SARLEADER of first scene if multiple
- use HDR (header), PRC (DPAF) or ODR (Delft) orbits

Data Ingestion (contd.)

- "make_raw_envi.pl" for Envisat ASAR data
- move Envisat data into scene directory, each frame or pass has one file:
  - ASA_IM__0CNPDE20040623_031040_000000752028_00018_12096_0013.N1 (ESA name)
    - *or* ENV1_4_442_2871_2889_18031.baq (WInSAR name)
- "make_raw_envi.pl" can concatenate many frames from a directory, uses ASA* or ENV* names to determine order
- use DOR (ESA DORIS) or ODR (Delft) orbits in "make_raw_envi.pl", then use "OrbitType=HDR" in processing

Data Ingestion (contd.)

- Envisat ASAR data name (from ESA):
- ASA_IM__0CNPDE20040623_031040_000000752028_00018_12096_0013.N1

processing level must be 0 — date — time hhmmss start — length (secs) — track — orbit

- Envisat ASAR data name (from WInSAR):
- ENV1_4_442_2871_2889_18031.baq

beam — track — frame start — frame end — orbit

Data Ingestion (contd.)

- "make_raw_alos.pl" for ALOS PALSAR data
- move ALOS data into scene directory, each frame format has three files (make sure you have raw "1.0" data):
  - data: IMG-HH-ALPSRP111820640-H1.0__A
  - leader: LED-ALPSRP111820640-H1.0__A
  - volume directory (not used): VOL-ALPSRP111820640-H1.0__A
- make_raw_alos.pl can concatenate many frames from a directory, uses IMG* names to determine order
- use HDR (header) orbits (contained in LED file)

Data Ingestion (contd.)

- ALOS PALSAR data name:
- IMG-HH-ALPSRP111820640-H1.0__A

polarization

orbit

frame

processing level
must be 1.0

pass direction

InSAR processing

- "process_2pass.pl" runs all the steps after raw data ready
- "int.proc" file (can actually have any name) in main directory (e.g. D170) specifies two scene directories (date or orbit) and other processing parameters for a given interferogram
- "roi.proc" file in main directory can specify additional parameters to apply to focussing all scenes in that main directory

InSAR processing

- "process_2pass.pl int.proc [DoItFrom EndItAt]" runs the steps, optionally can specify start and stop points to do or re-do part of the processing

- "int.proc" file must contain:

  - SarDir1 = name of image directory 1 (master scene) with raw data

  - SarDir2 = name of image directory 2 (slave scene) with raw data

  - IntDir = name of interferogram directory (e.g., int_031101_060916), better to include full path

SAR Image Formation

$DoItFrom "raw"

- "process_2pass.pl" first calls "raw2ampintcor.pl" to do SAR image formation, tie matching, interferogram and correlation

- ROI_pac does azimuth spectrum filtering as part of focussing the raw data to single-look complex (SLC)

- "dop_avg.pl" calculates the average of the Doppler centroids of the two input scenes and the amount of azimuth spectrum overlap for filtering

SAR Image Formation (cont'd)

- "roi_prep.pl" prepares input files for "roi" program for each scene, and creates $date.slc.rsc files

  $EndItAt "roi_prep"
- "baseline.pl" calculates baseline between two orbits for the area covered by the scenes (and gross offsets)=> $date1_$date2_baseline.rsc

  $EndItAt "orbbase"
- "roi" (repeat orbit interferometry) does SAR image formation or focussing (range and azimuth compression)
- "roi" output is full resolution single-look complex image "$date.slc"
- option: "concurrent_roi = yes" runs roi on both scenes at same time (good with multiple CPUs and enough RAM)

SAR Image Formation (cont'd)

- Full resolution $date.slc averaged by "look.pl" with 16 looks in range and $pixel_ratio(default=5)*16 looks in azimuth

- output "$date_16rlks.slc"

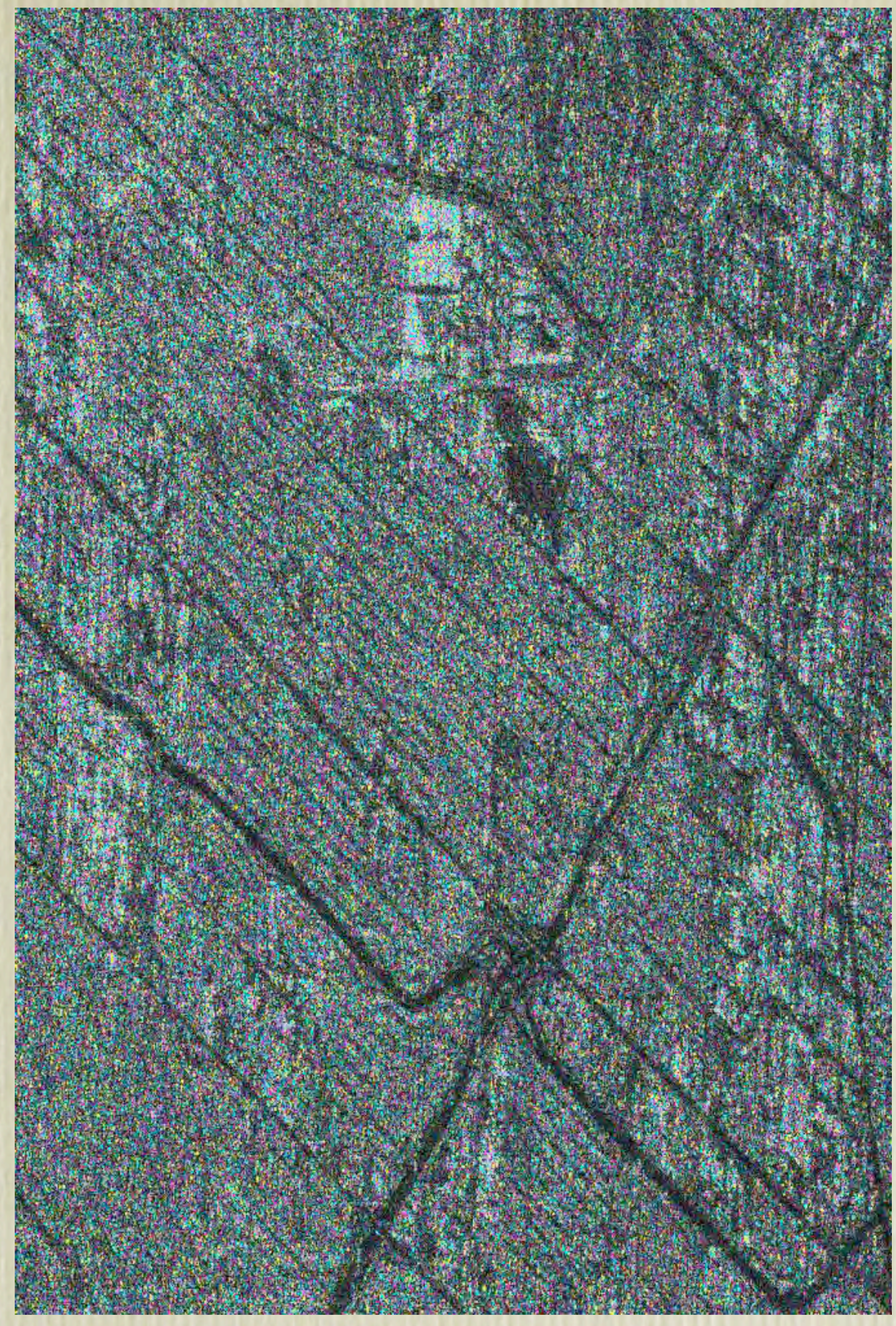
- should be image of full area of SAR scene in radar geometry (increasing range to right, along track down)

- may have extra "black" at bottom due to last processing patch extending past data

track 170 over LA

SAR Image Formation (cont'd)

- Full resolution $date.slc in radar geometry
- has "stretched" look due to azimuth spacing smaller than range spacing
- ERS & Envisat I2: azimuth ~4 m range: 8/sin(23°)=20 m
- phase is random
- example from descending Envisat track 170 over LA



$EndItAt "slcs"

Image coregistration

$DoItFrom "slcs"

- "raw2ampintcor.pl" moves IntDir to do rest of processing
- "make_offset.pl" calculates 2D field of offsets between two SLC images
- initial estimate of offsets between scenes was calculated from orbits and scene parameters saved in IntDir/$date1_$date2_baseline.rsc
- initial offsets can be manually overridden by "x_start" and "y_start" in int.proc file

$EndItAt "offsets"

Image coregistration (cont'd)

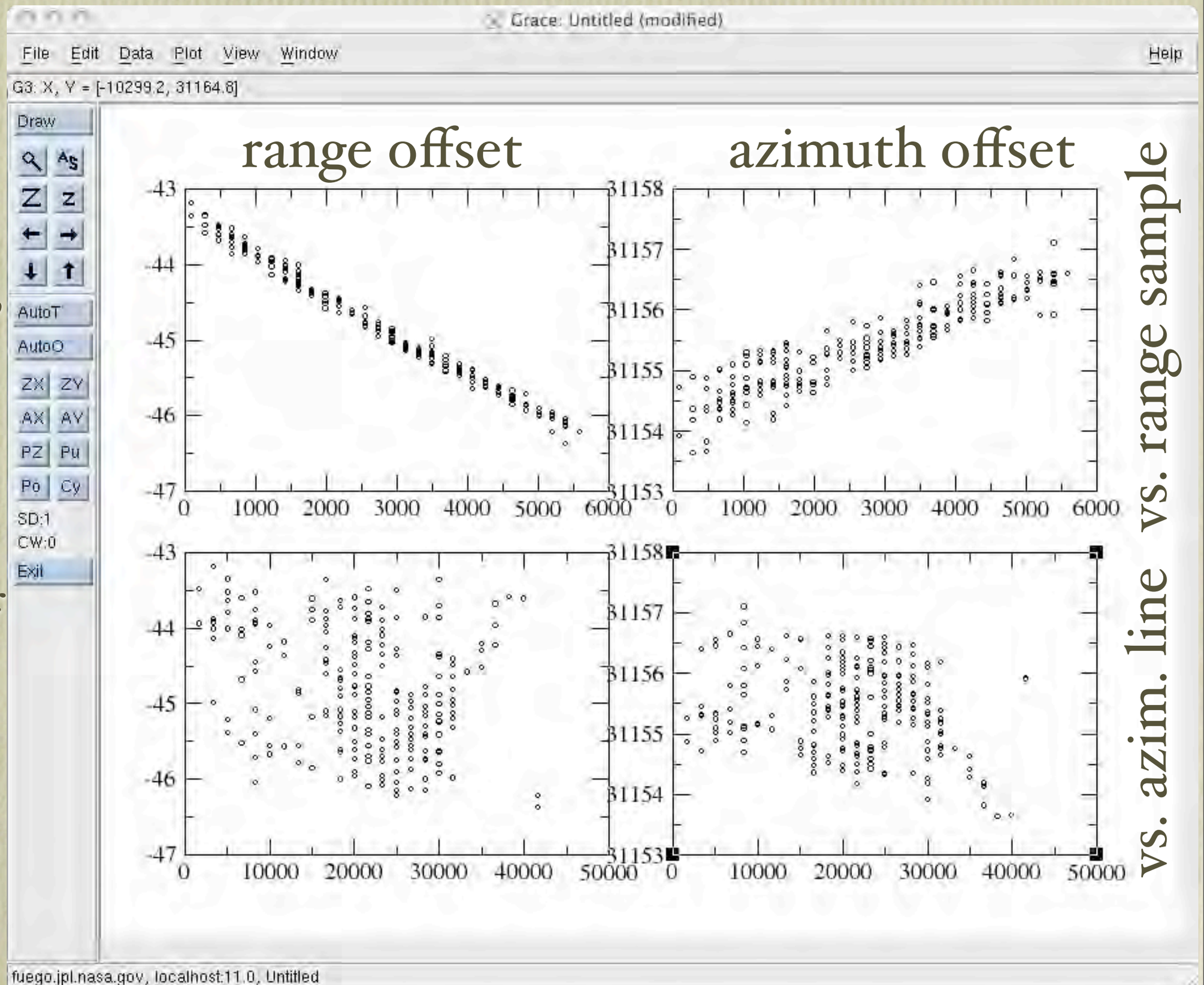
- "make_offset.pl" first runs a gross matching to refine the initial offsets from the orbits, calls "offset.pl"
- "offset.pl" runs "ampcor" the matching program
- input: $date1-$date2_ampcor_gross.in
- output: $date1-$date2_ampcor_gross.off, .out
- "make_offset.pl" runs "fitoff" to do first order affine transformation fit to gross offsets and cull points that exceed 0.5 pixels from fit: $date1-$date2_cull_gross.off, fitoff_ampcor_gross.out
- fails if initial offsets are incorrect or scenes change a lot

Image coregistration (cont'd)

- "make_offset.pl" takes average of gross offsets after cull
- "make_offset.pl" next runs fine matching to refine the offsets from gross matching, calls "offset.pl" again
- input: $date1-$date2_ampcor.in
- output: $date1-$date2_ampcor.off, .out
- "make_offset.pl" runs "fitoff" to do first order affine transformation fit to fine offsets and cull points that exceed 0.08 pixels from fit: $date1-$date2_cull.off, fitoff_ampcor.out
- occasionally fails if scenes are very different (e.g., ocean)

Image coregistration (cont'd)

- Can look at offsets with "PlotOffset.pl" (requires xmgrace)
- e.g., $date1-$date2_cull.off
- this pair has Bperp ~240 m



Form Interferogram

$DoItFrom "offsets'

- “raw2ampintcor.pl” calls “resamp.pl” that sets up input and runs “resamp_roi” program
- “resamp_roi” fits a second order polynomial function to culled offsets and uses this to resample the “slave” $date2 SLC to coregister it to the “master” $date1 SLC
- interferogram is formed by multiplying each complex pixel of $date1 by complex conjugate of $date2 pixel
- $pixel_ratio looks are taken in azimuth before writing interferogram to complex output file $date1-$date2.int
- amplitudes of $date1 and $date2 are put into $date1-$date2.amp

$EndItAt “resamp”

Form Interferogram (cont'd)

- "raw" interferogram $date1-$date2.int has all phase components included, orbit geometry, topography, deformation, atmosphere, etc.

- magnitude is combination of coherence and backscatter

- example: Envisat D170 pair 030927-051001.int (top left corner)

- P_BASELINE_TOP_HDR 4.663504062735̄72 m

- P_BASELINE_BOTTOM_HDR -27.4076891813446 m

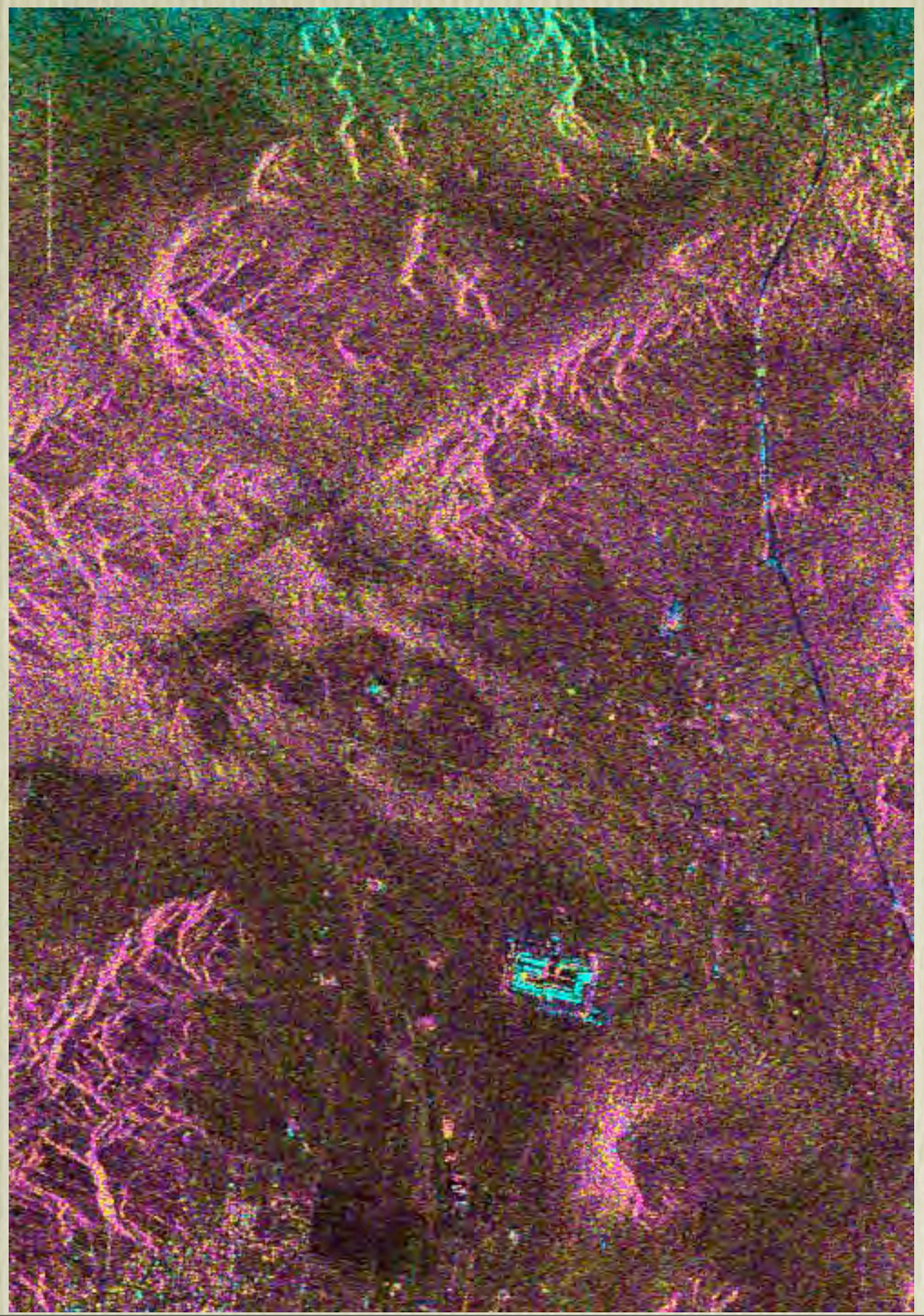
Form Interferogram (cont'd)

- amplitudes (sqrt(power)) of $date1 and $date2 in $date1-$date2.amp interleaved by pixel

- can be displayed as "complex" with "mdx" to view differences in amplitudes between scenes

- example: Envisat D170 pair 030927-051001.int (top left a little to the right of corner)

- Garlock fault and Ft. Irwin?



Flatten Interferogram

$DoItFrom "resamp"

- ROI_pac does an initial flattening of the interferogram by removing the expected phase for the InSAR orbit geometry with a “curved Earth” without any topography

- “raw2ampintcor.pl” creates “reference.hgt” which is fake DEM with elevation constant at $ref_height (default zero)

- “raw2ampintcor.pl” calls “diffnsim.pl” to calculate and subtract phase based on “reference.hgt” and orbit $OrbitType (e.g., PRC) from raw interferogram to produce “flat_PRC_$date1-$date2.int” and “ramp_PRC_4rlks.unw”

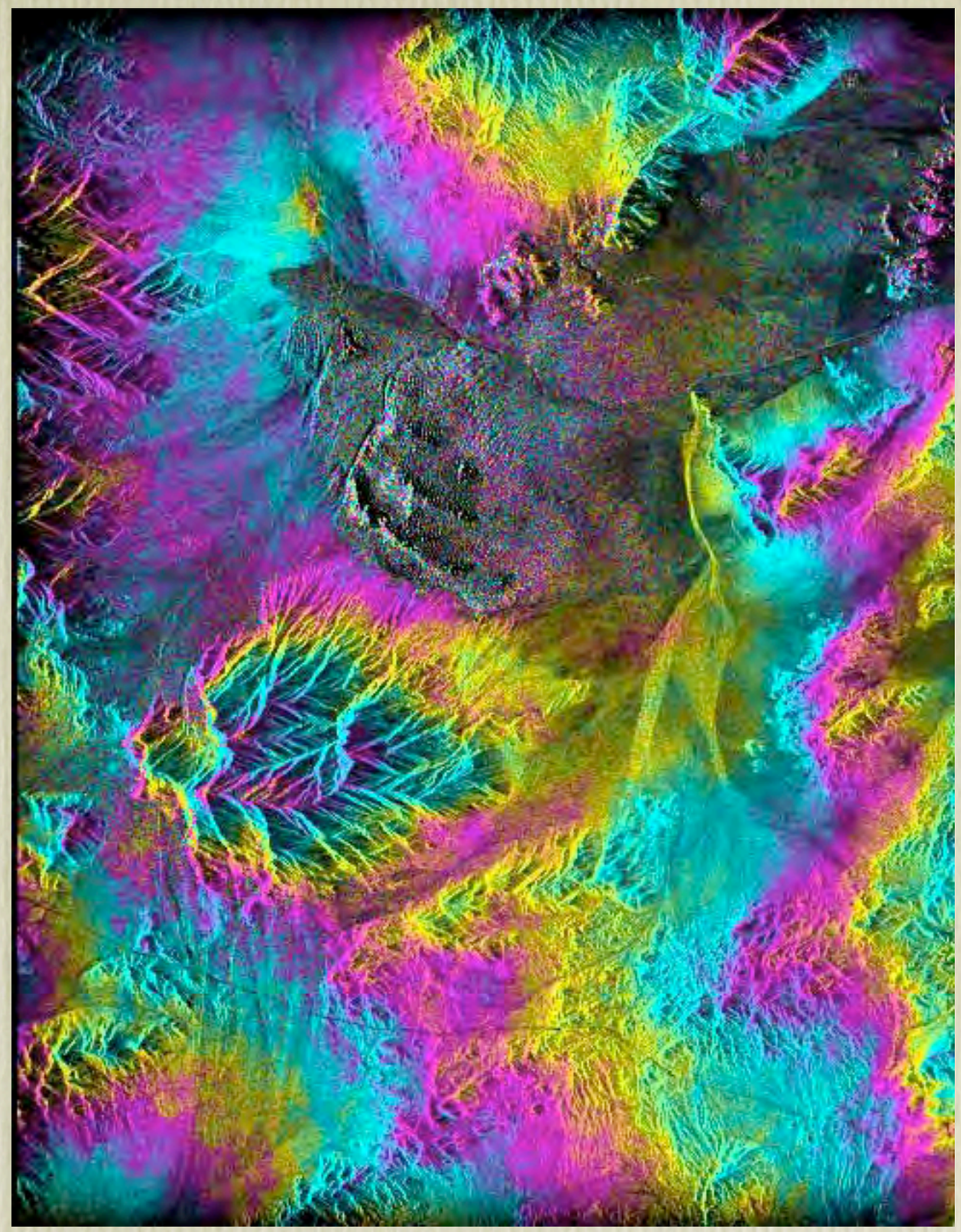
$EndItAt “flatorb”

Flatten Interferogram (cont'd)

- "raw2ampintcor.pl" calls "look.pl" to take looks (default 4) on "flat_PRC_$date1-$date2.int", "ramp_PRC.unw" and "reference.hgt"

- flat interferogram still has phase proportional to elevation but "orbit fringes" have been removed

- example: flat_PRC_930110-950523_4rlks.int (from ROI_pac test data)



Flatten Interferogram (cont'd)

- "ramp_PRC.unw" and "ramp_PRC_4rlks.unw" have the "orbit" phase that was subtracted, this is the phase due to the baseline and baseline changes assuming no topography

- example: ramp_PRC_4rlks.unw (from ROI_pac test data)

- P_BASELINE_TOP_PRC
  -45.3217489736805 m

- P_BASELINE_BOTTOM_PRC
  -41.6760201631844 m

Calculate Correlation

$DoItFrom “flatorb"

- After the initial flattening of the interferogram, ROI_pac calculates the spatial correlation of the phase as an estimate of the interferometric coherence {0.0-1.0}

- “raw2ampintcor.pl” calls “make_cor.pl” which runs “cchz_wave” to calculate the correlation on the flattened interferogram flat_PRC_ $date1-$date2.int and the $date1-$date2.amp using a 5 x 5 pixel triangular weighted window=> $date1-$date2.cor with amplitude and correlation band interleaved by line (rmg)

- The amplitude of the correlation image is the average of the amplitudes of the two scenes

$EndItAt “full_res”

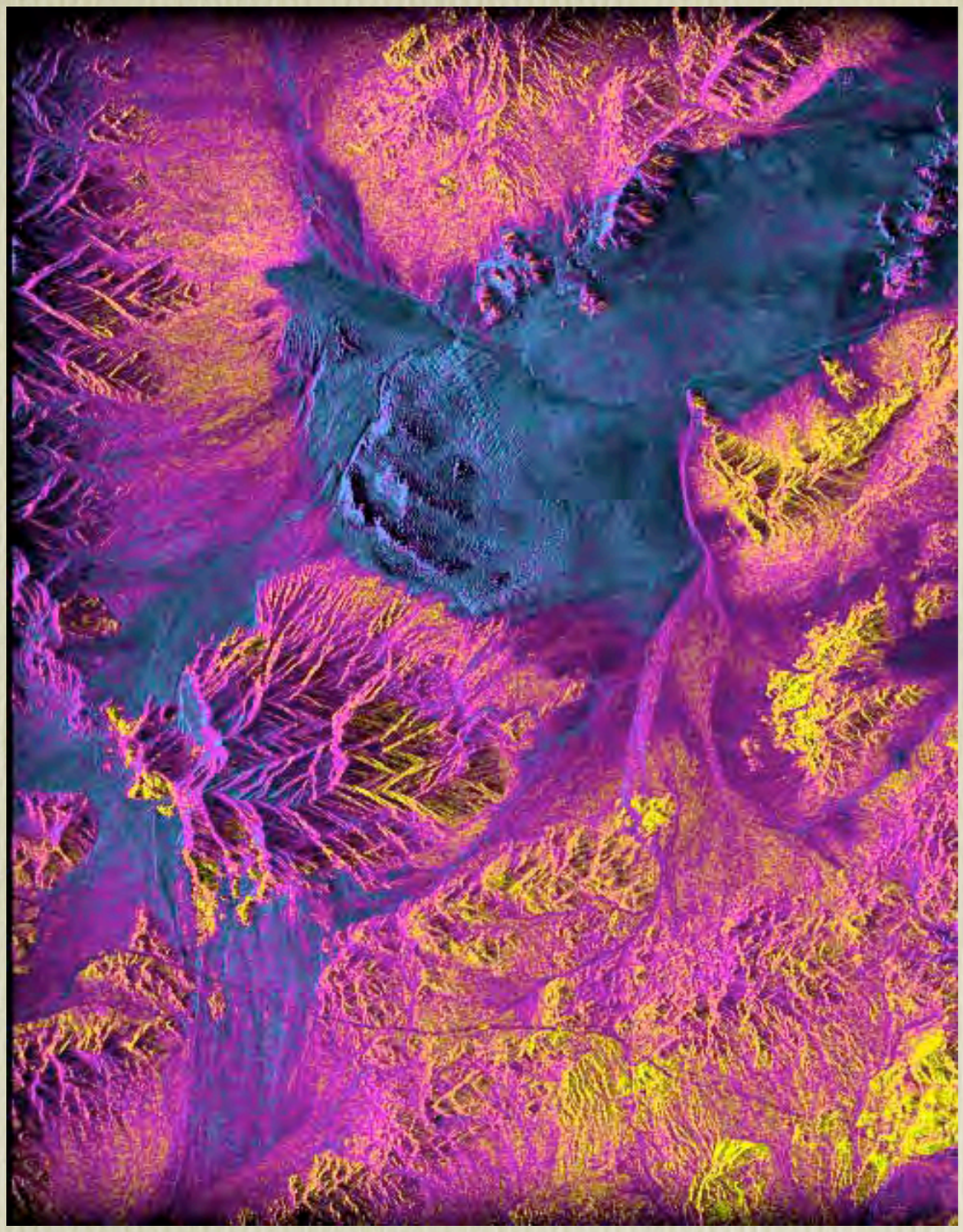
Correlation (cont'd)

- "raw2ampintcor.pl" calls "look.pl" to take looks (default 4) on "$date1-$date2.cor"

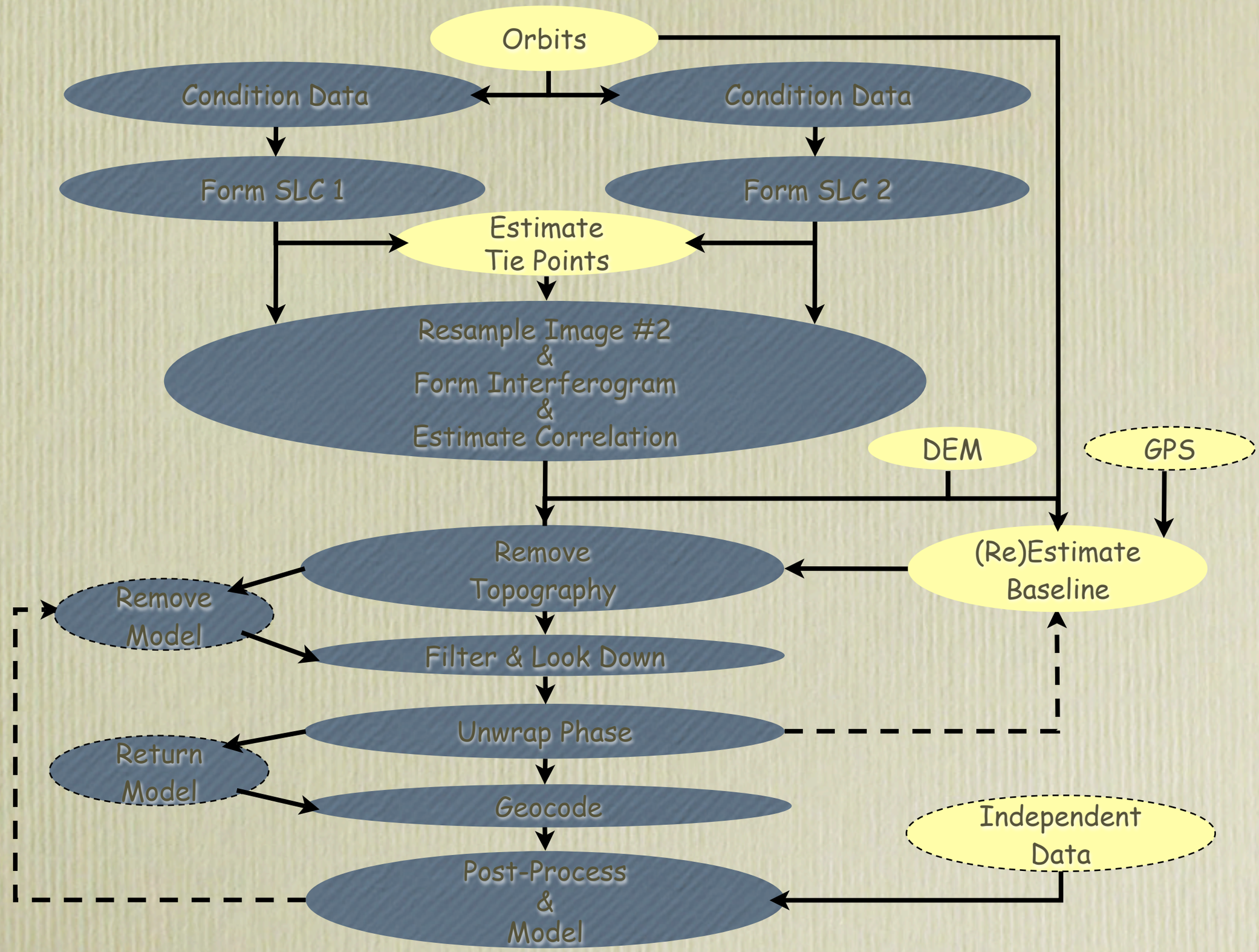
- InSAR coherence measures how much radar backscatter of each pixel changed between scenes, but is also affected by steep slopes if baselines long

- example: 930110-950523_4rlks.cor (from ROI_pac test data)

- color wrap 1.2: blue, purple low; yellow, green high



ROI_pac Two-pass Processing Flow



modified from Mark Simons

Remove Topography

$DoItFrom
"full_res"

- ROI_pac does a simulation of the radar image amplitude from a DEM and the orbit for the master scene, and projects the elevations into the radar coordinates of the inteferogram, then calculates the topographic phase and subtracts it from the original interferogram

- "process_2pass.pl" calls "dem2diff.pl" that then calls "make_sim.pl" (if "do_sim" = "yes") to run simulation in "SimDir" (default "SIM") with "$DEM" at $Rlooks_sim looks

$EndItAt "seismic"

Remove Topography (cont'd)

- "make_sim.pl" first moves to SIM directory and converts the orbit data (specified by OrbitType) to $date1.orrm file
- Then it calls "gradient.pl" to calculate the gradient (slope) of the DEM.dem file, creating DEM.slp that has the dz/dx and dz/dy interleaved by pixel (complex format)
- The DEM can be either in latitude-longitude (LATLON, e.g., 1-arcsecond spacing) or UTM (e.g., 30 m spacing) projection
- YSTEP negative if first DEM sample at top
- for UTM projection, specify zone (e.g., UTM10) and datum (NAD27 or WGS84)

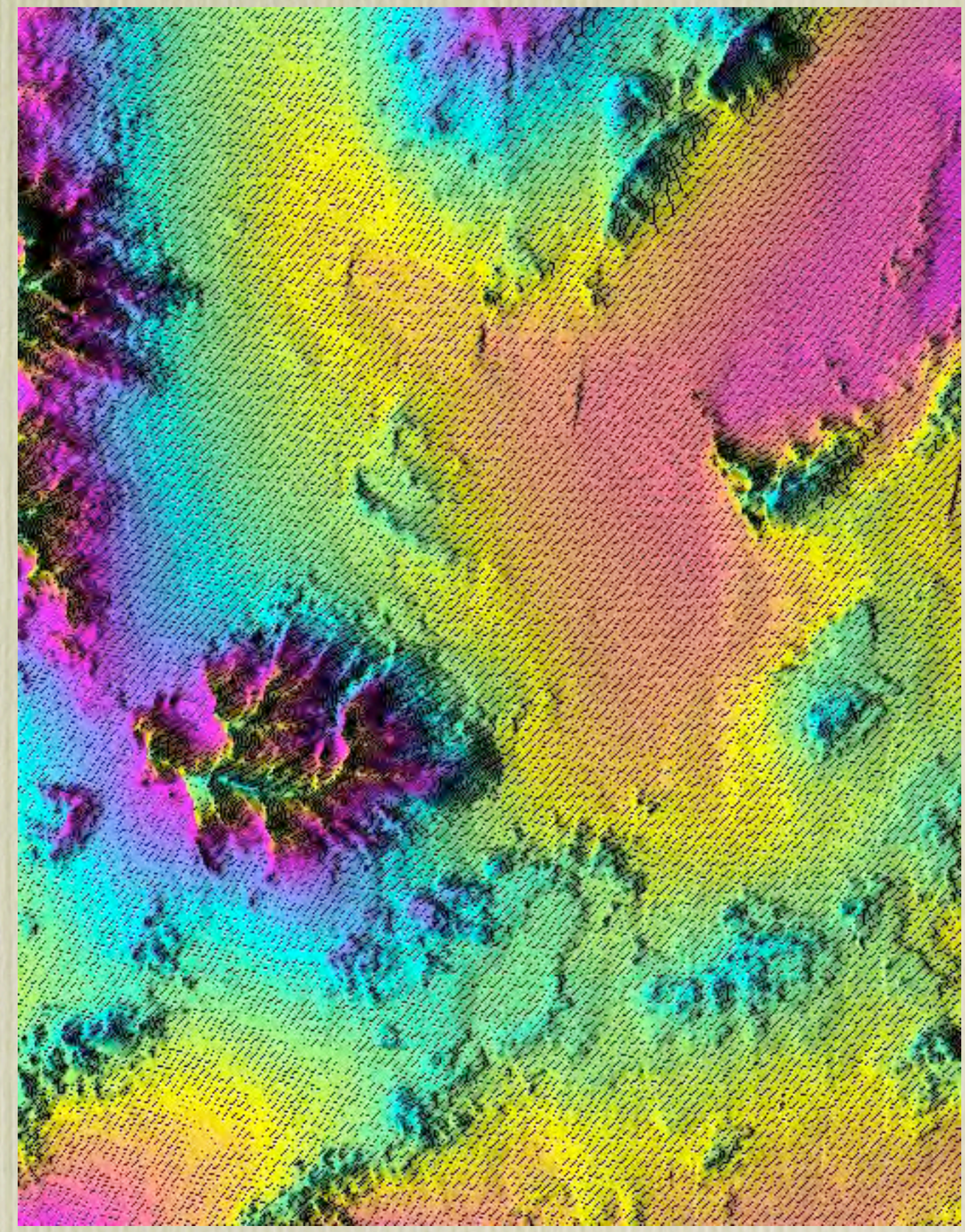
Remove Topography (cont'd)

- "make_sim.pl" sets up and runs "IntSim" to do the simulation in radar geometry, producing SIM_raw.hgt (rmg file with amplitude and height)

- example: SIM_raw.hgt from ROI_pac test data

- color wrap 1000 m

- note that DEM samples are irregularly spaced in radar coordinates, especially on slopes



Remove Topography (cont'd)

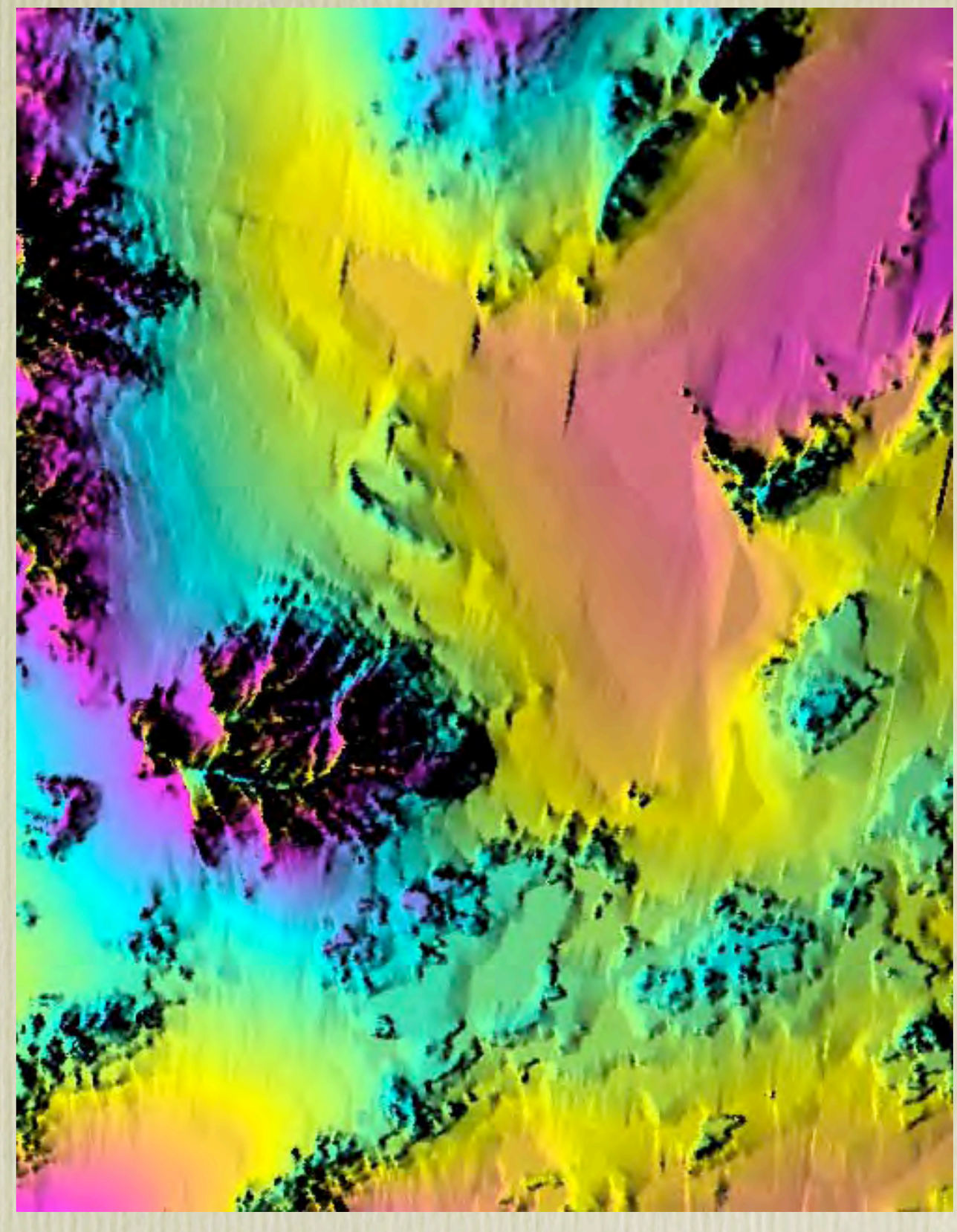
- "make_sim.pl" sets up and runs "Aik_resample" to interpolate simulation, producing SIM_4rlks.hgt (rmg file with amplitude and height)

- also takes looks to make SIM_16rlks.hgt

- example: SIM_4rlks.hgt from ROI_pac test data

- color wrap 1000 m

- note that this is a poor quality pre-SRTM 3-arcsecond DEM



Remove Topography (cont'd)

$DoItFrom "begin_sim"

- "dem2diff.pl" moves to IntDir and creates symbolic links to SIM_4rlks.hgt and SIM_16rlks.hgt in SIM directory
- Then it calls "synth_offset.pl" to check coregistration between simulated image and SAR image (using simulated amplitude and amplitude in 4rlks.cor file)
- "synth_offset.pl" calls "offset.pl" that runs "ampcor" to do gross matching "ampmag_gross.off" and fine matching (ampmag.off), similar to "make_offset.pl"
- then it runs "fitoff" to determine affine transformation between simulation and SAR image=> "cull.out"
- sometimes fails if not enough topographic features

$EndItAt "done_sim_off"

Remove Topography (cont'd)

$DoItFrom
"done_sim_off"

- "dem2diff.pl" calls "synth2radar.pl" to resample simulation in exact coregistration with SAR image using affine transform

- "synth2radar.pl" calls "find_affine.pl" to extract affine transform parameters from "cull.out" and puts it in date1-date2_Rlooks_sim_SIM.aff file (e.g., "930110-950523_4rlks_SIM.aff")

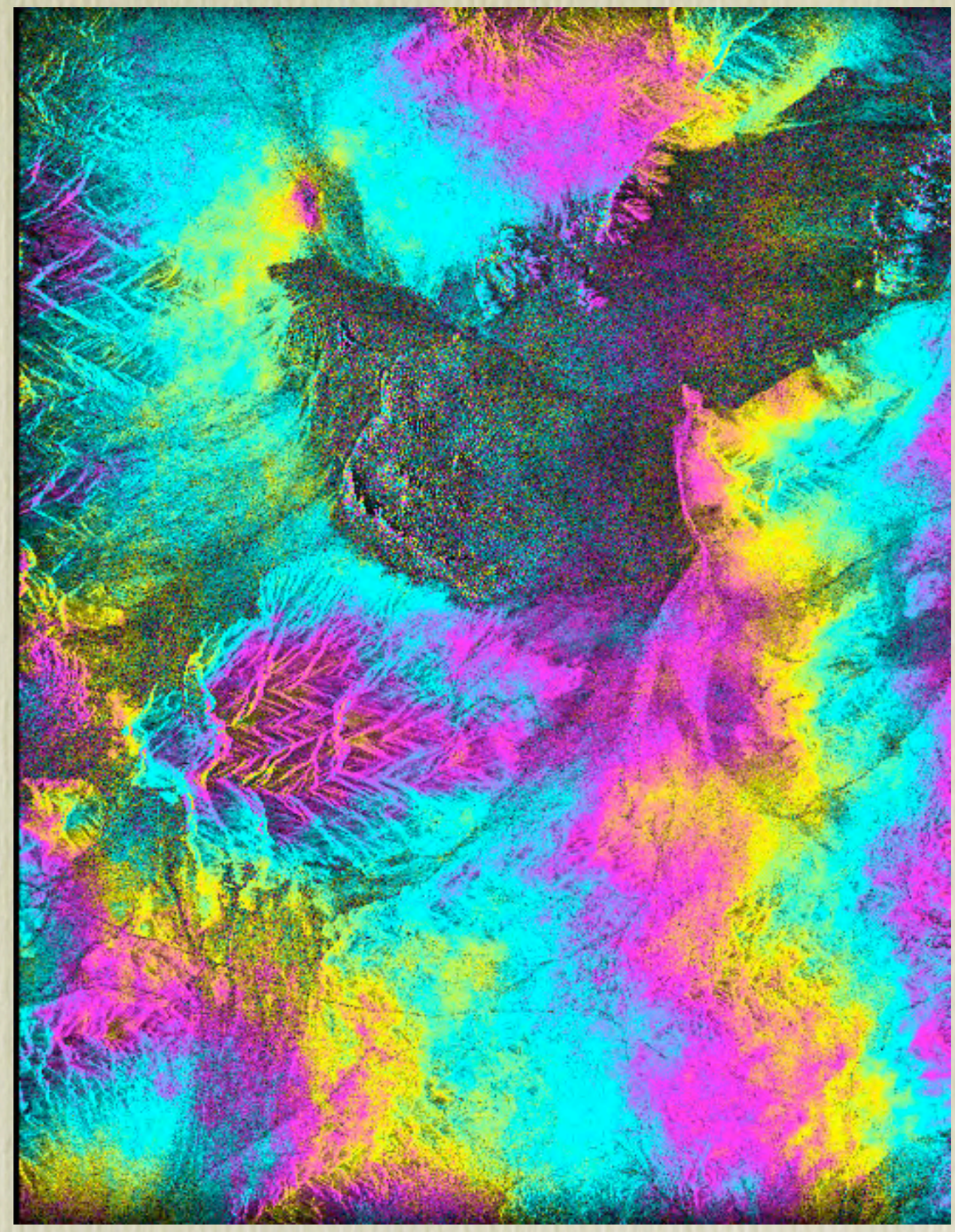
- then it calls "rect.pl" to do the resampling, output in "radar_4rlks.hgt" (assuming Rlooks_sim=4) should be coregistered to SAR image

Remove Topography (cont'd)

- "dem2diff.pl" calls "diffnsim.pl" to calculate phase due to topography on curved Earth with orbit $OrbitType and subtracts it from original interferogram, producing output at $Rlooks_sim resolution

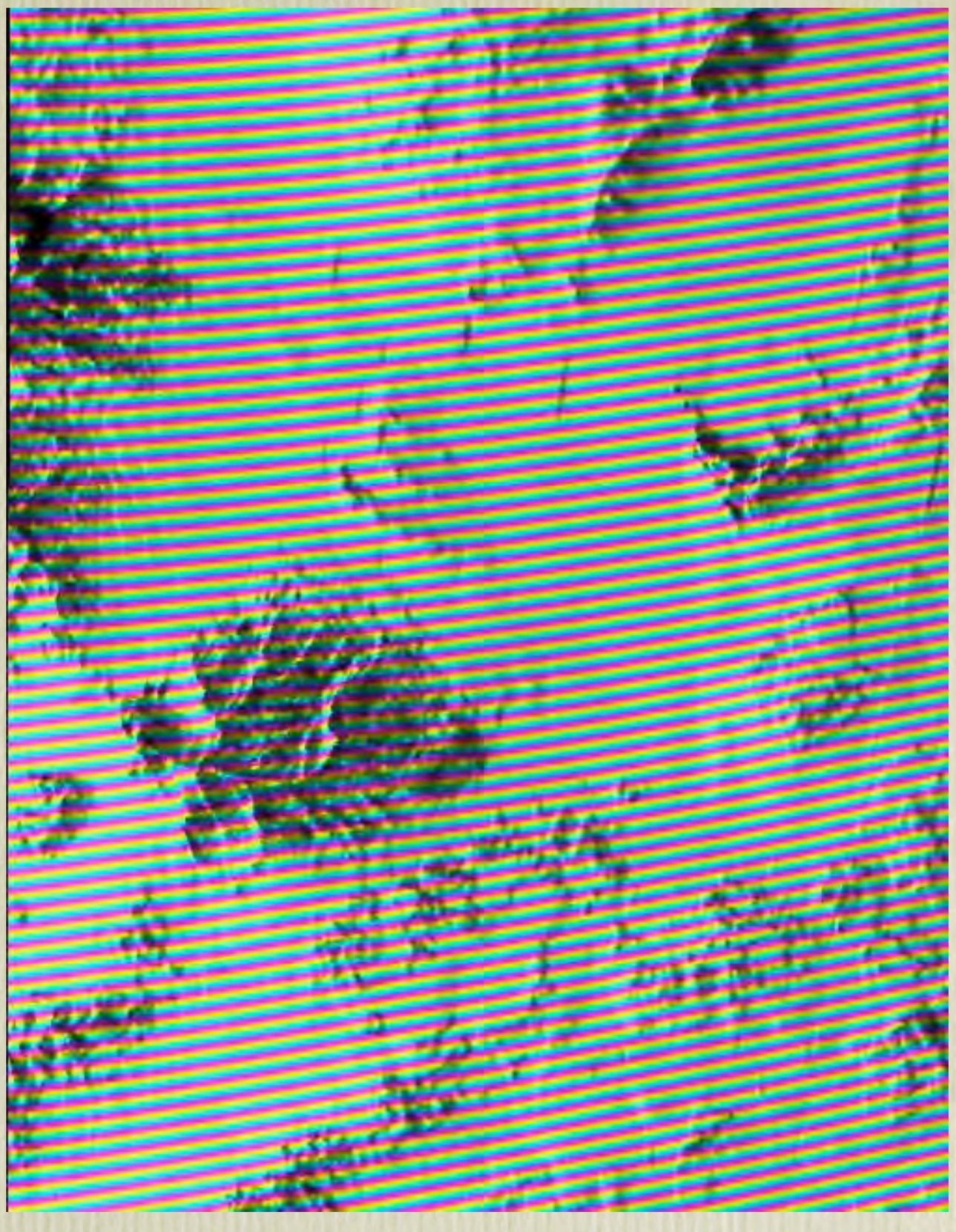
- example: 930110-950523-sim_PRC_4rlks.int (from ROI_pac test data)

- amplitude is still interferogram amplitude that includes coherence



Remove Topography (cont'd)

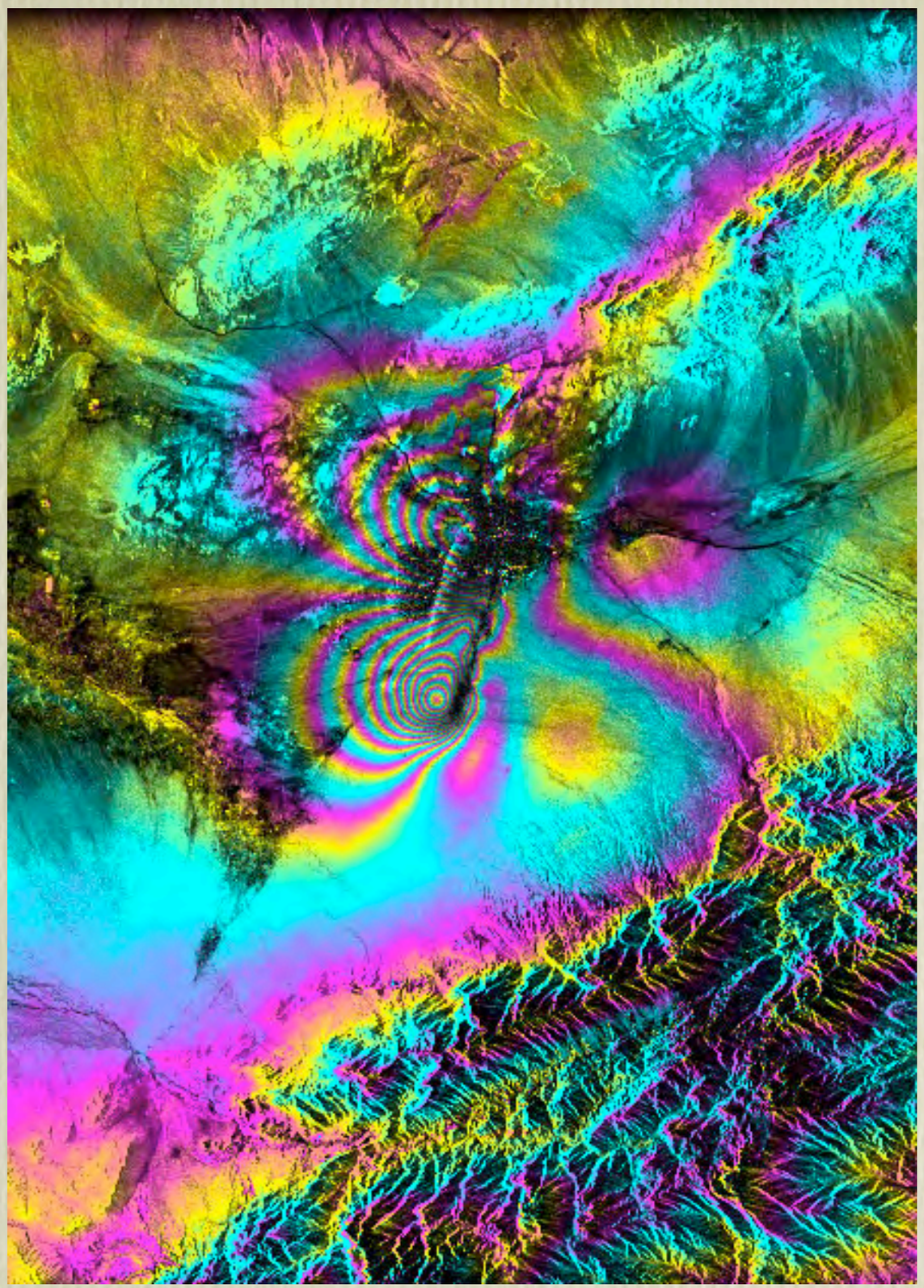
- "diffnsim.pl" also writes out the phase due to topography on curved Earth with orbit $OrbitType that was subtracted (rmg file)
- example: radar_PRC_4rlks.unw (from ROI_pac test data)
- amplitude is simulated amplitude
- phase usually dominated by "orbit" phase



$EndItAt "done_sim_removal"

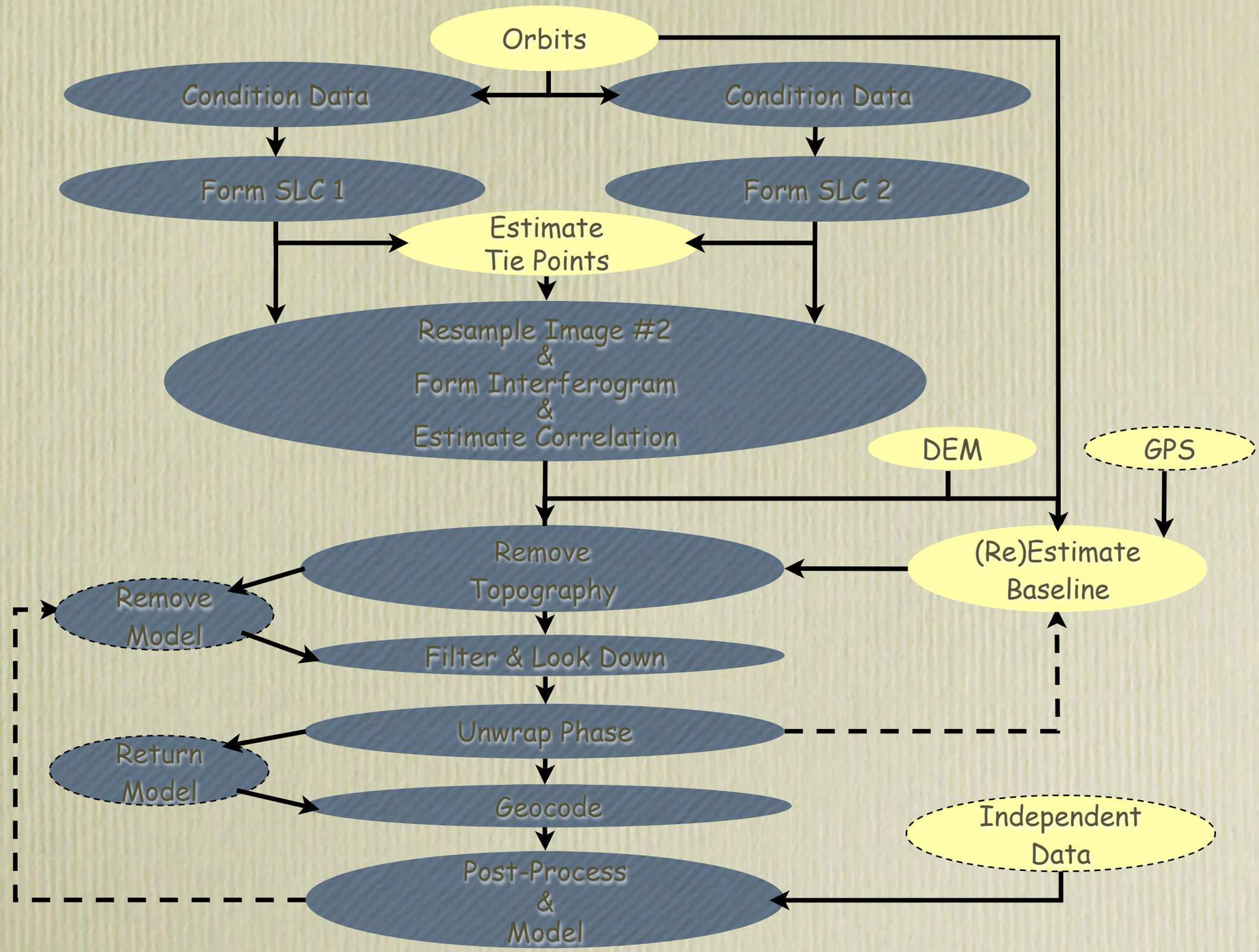
Remove Topography (cont'd)

- "process_2pass.pl" finally takes optional looks on outputs of "dem2diff.pl" to change from Rlooks_sim to Rlooks_unw if the latter is more looks
- this is called the "seismic" step because earthquakes should be clear at this point
- example: 031203-040211-sim_HDR_8rlks.int (Envisat pair from 2003 Bam, Iran earthquake)



$EndItAt "seismic"

ROI_pac Two-pass Processing Flow



modified from Mark Simons

Remove Model

$DoItFrom
“seismic"

- “process_2pass.pl” checks the MODEL parameter, which can include CO_MODEL (coseismic models) and INTER_MODEL (interseismic models)
- Models are in original DEM coordinates, but can be at coarser resolution
- Models are projected into SAR geometry and subtracted from the “seismic” interferogram at the Rlooks_unw resolution

$EndItAt “begin_filt”

Filtering interferogram

$DoItFrom “begin_filt"

- “process_2pass.pl” calls “int2filtmaskunwrap.pl” to do filtering, masking and unwrapping
- Filtering is done differently depending on unwrapping method (unw_method)
- For “classic” unwrapping (unw_method = “old”, the default), filtering is done by “filter.pl” before masking
- When unwrapping with “icu” (unw_method = “icu”), the filtering is done inside “icu”

$EndItAt “done_filt”

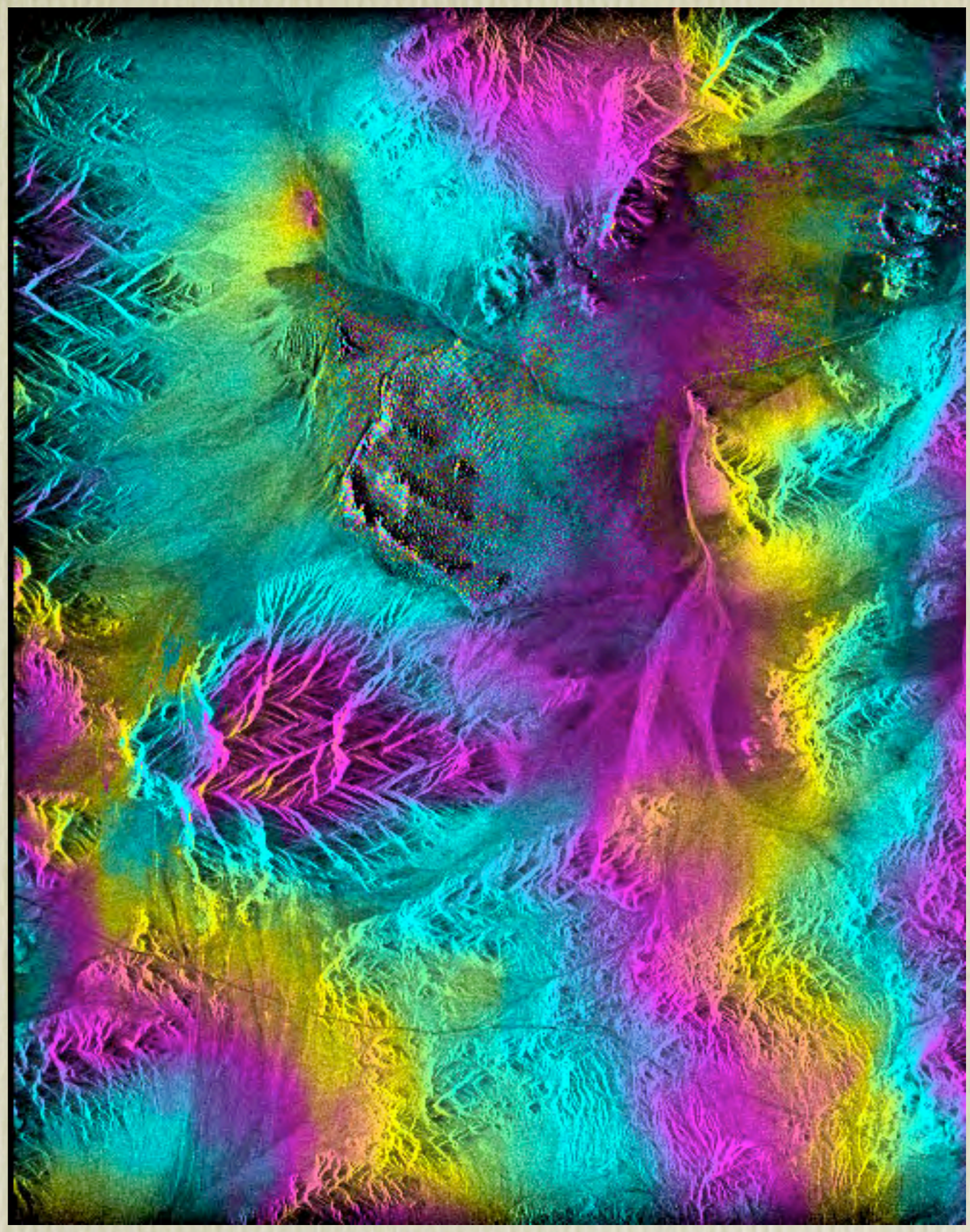
Filtering interferogram (cont'd)

- "int2filtmaskunwrap.pl" calls "filter.pl" to do filtering based on "Filt_method" which is either "psfilt" (default) or "adapt_filt"
- "filter.pl" first runs power spectrum filtering (Goldstein and Werner, 1998, GRL) with the power spectrum exponent specified by "FilterStrength" (default 0.75)
  - if "Filt_method" = "adapt_filt" then "filter.pl" calls "adapt_filt.pl" to run an additional filtering (sometimes useful with very noisy data, usually ugly)
  - there is also a way to run a simple boxcar filter with "lowpass.pl" but this is not in the regular workflow

Goldstein, R.M., and Werner, C.L., 1998, Radar interferogram filtering for geophysical applications: Geophysical Research Letters, v. 25, p. 4035-4038.

Filtering interferogram (cont'd)

- after running filter(s) "filter.pl" replaces the amplitude of the result "filt_date1-date2-sim_{OrbitType}_{Rlooks_unw}rlks.int" (which is heavily blurred by the filtering) with the average SAR amplitude from the ".cor" file
- example: filt_930110-950523-sim_PRC_4rlks.int (from ROI_pac test data, default FilterStrength=0.75)



$EndItAt "done_filt"

Masking interferogram

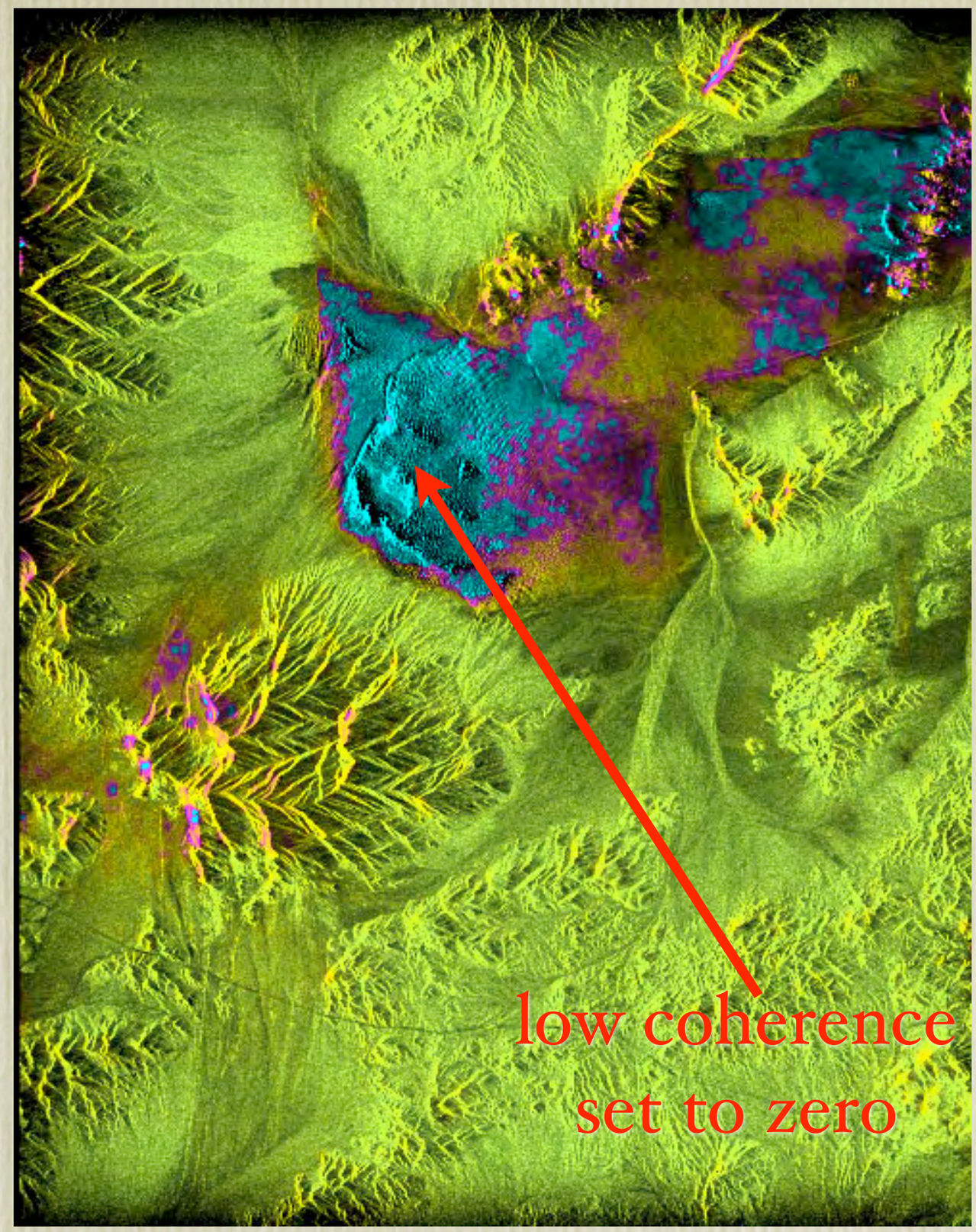
$DoItFrom “done_filt"

- “int2filtmaskunwrap.pl” calls “make_mask.pl” to make a mask for area of interferogram to unwrap (only done with classic unwrapping “unw_method” = “old”)
- “make_mask.pl” calls “phase_slope” (which removes local phase slope) and “phase_mask” programs which make a mask based on the local phase variance (to remove areas of low coherence)
- then “make_mask.pl” calls “int_thr” to make a mask based on the amplitude “Threshold_mag” (default 5.0e-5), which cuts out dark edges of interferogram, and combines this with coherence mask

$EndItAt “make_mask”

Masking interferogram (cont'd)

- "phase_mask" program calculates phase variance from filtered & slope-removed interferogram so effective coherence is greatly increased
- "phase_mask" applies phase variance threshold "sigma_thresh" (default 1.0), converts it to an effective coherence and writes out phase_var_PRC_4rlks.msk
- example: phase_var_PRC_4rlks.msk (from ROI_pac test data with defaults)



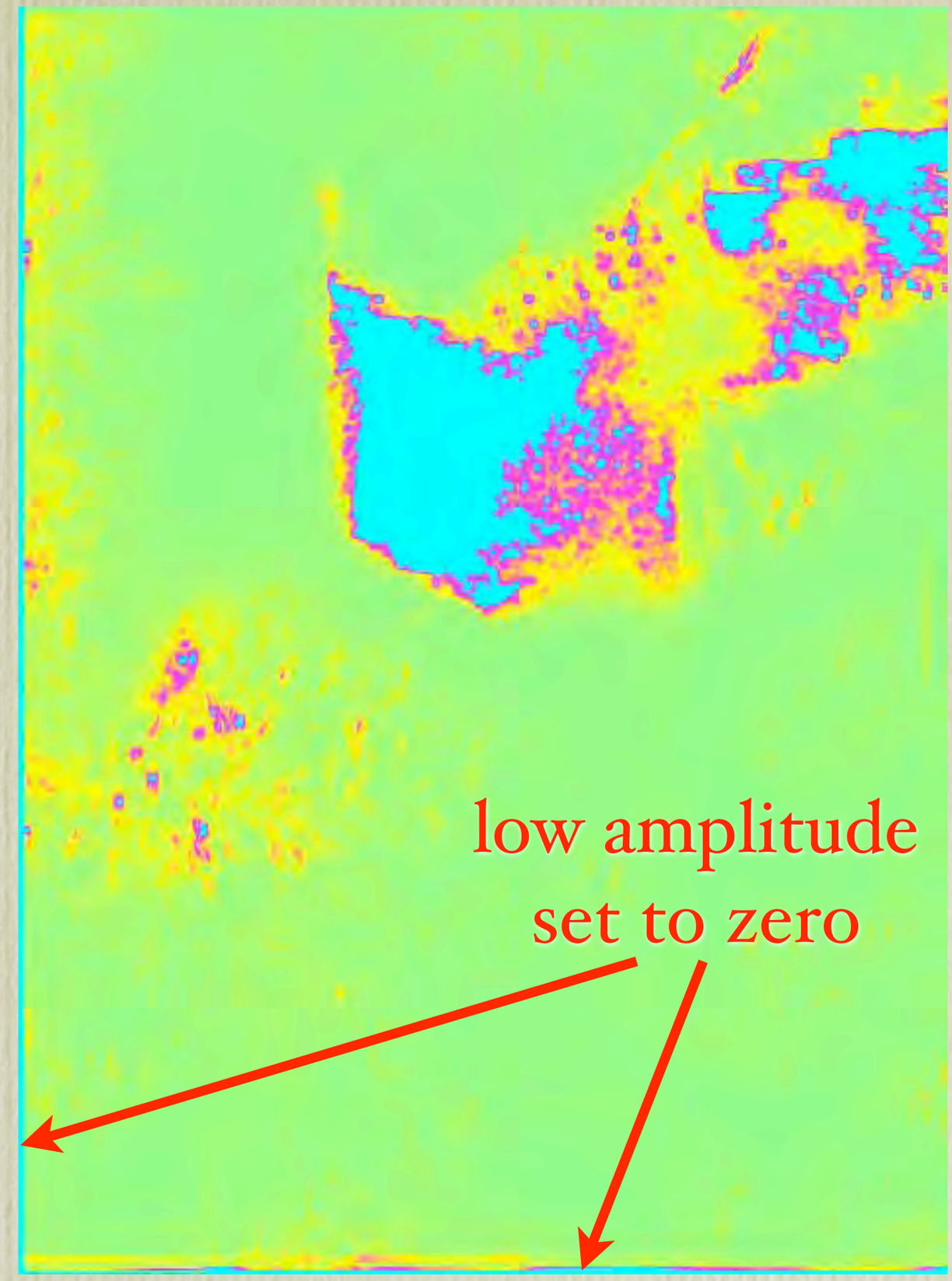
Masking interferogram (cont'd)

- "make_mask.pl" combines low amplitude mask with coherence mask to make "low_cor_PRC_4rlks.msk"

- example: low_cor_PRC_4rlks.msk (from ROI_pac test data with defaults)

- color wrap 1.2: masked coherence cyan, low magenta, high yellow and green

$EndItAt "make_mask"



Unwrapping

$DoItFrom " make_mask"

- For the classic phase unwrapping ("unw_method" = "old"), "int2filtmaskunwrap.pl" first calls "new_cut.pl" to do the first steps (see Goldstein, Zebker and Werner, 1988; GZW88 for short)
- then "int2filtmaskunwrap.pl" calls "unwrap.pl" to complete the unwrapping with output in "filt_$date1-$date2-sim_$OrbitType_$Rlooks_unw_c$UnwrappedThreshold.unw"

Goldstein, R.M., Zebker, H.A., and Werner, C.L., 1988, Satellite radar interferometry—two-dimensional phase unwrapping: Radio Science, v. 23, p. 713–720.

$EndItAt "unwrapped"

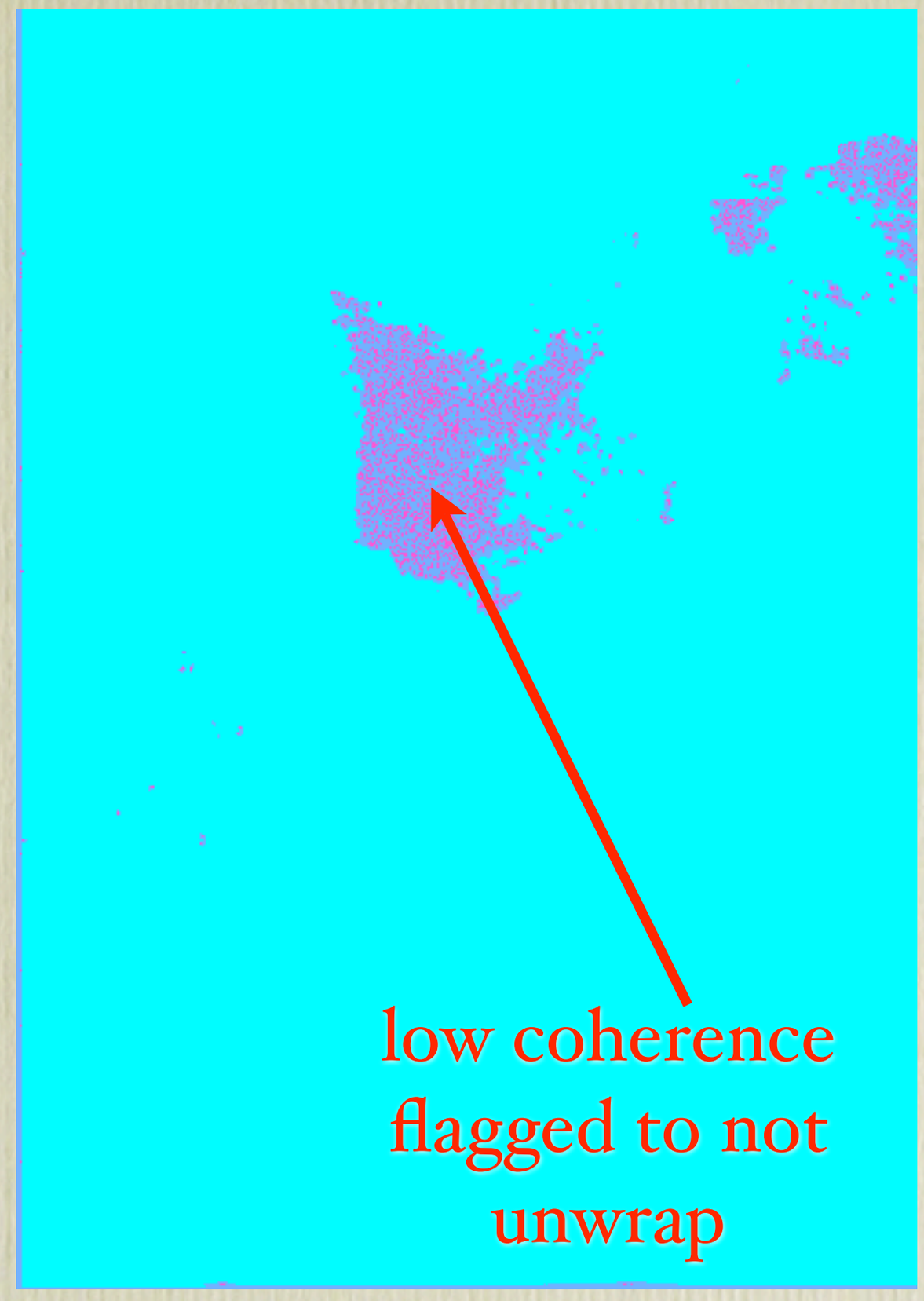
Unwrapping (cont'd)

- "new_cut.pl" calls "residue" program to calculates phase residues from filtered & slope-removed interferogram (see GZW88)
- residues are stored in "filt_$date1-$date2-sim_$OrbitType_${Rlooks_unw}rlks_cut.flg"
- Then it calls "trees" to draw phase cuts to connect the residues (see GZW88)

Unwrapping (cont'd)

- "unwrap.pl" calls "corr_flag" program to combine the unwrapping mask with the phase cut trees, optionally applying a higher coherence threshold "UnwrappedThreshold" (default 0.1), output is new flag file filt_$date1-$date2-sim_$OrbitType_${Rlooks_unw}rlks_c{10*UnwrappedThreshold}.flg

- example: filt_930110-950523-sim_PRC_4rlks_c10.flg (from ROI_pac test data with defaults)



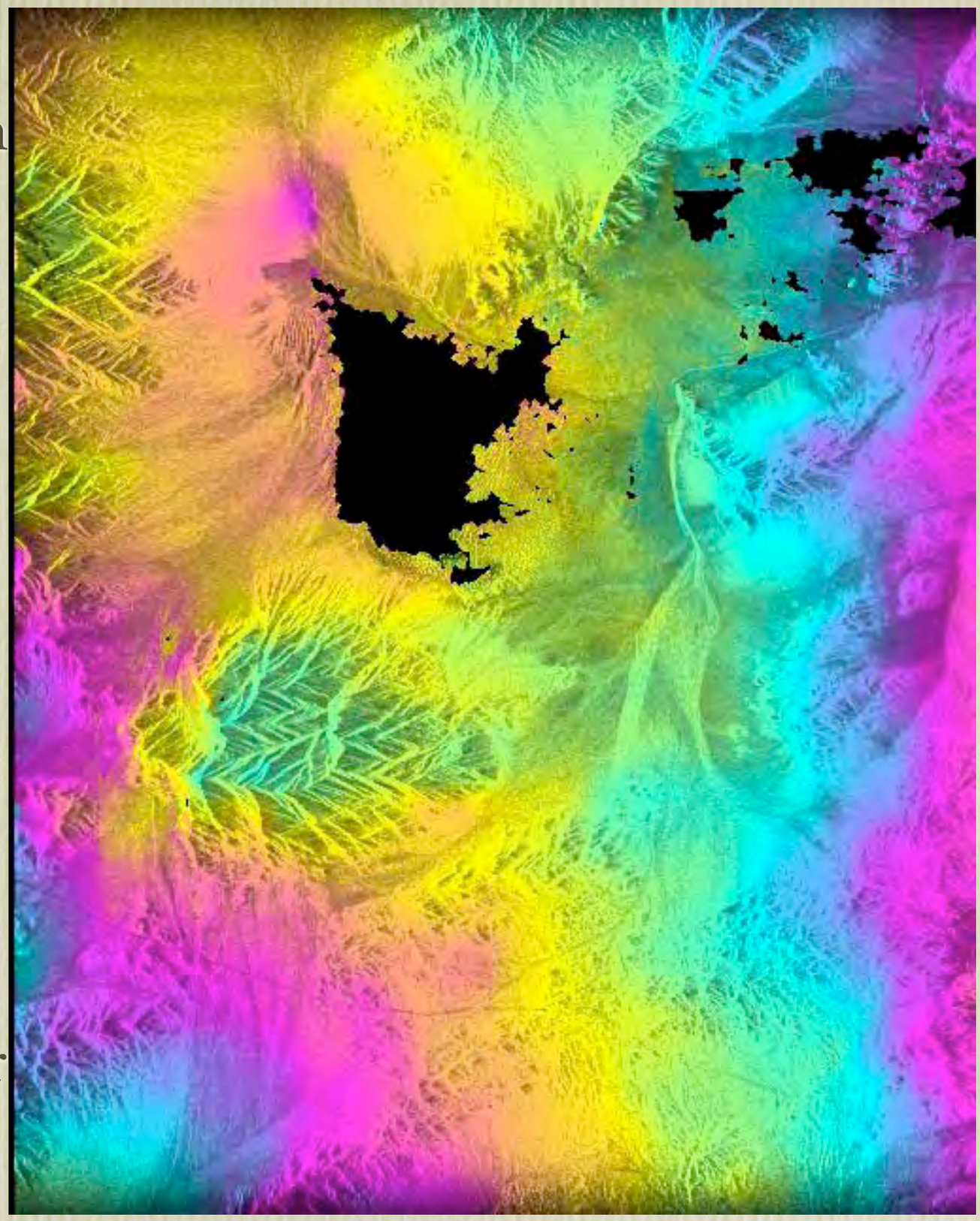
Unwrapping (cont'd)

- "unwrap.pl" calls "grass" to unwrap the filtered interferogram phase between the trees of phase cuts, using the mask and starting at the "seed" which is defined as zero phase (see GZW88)

- seed default location is center of image, can be changed with "unw_seedx" and "unw_seedy"

- example: filt_930110-950523-sim_PRC_4rlks_c10.unw (from ROI_pac test data with defaults)

- color wrap: 11.4 radians = 5 cm for C-band



$EndItAt "unwrapped"

Unwrapping (cont'd)

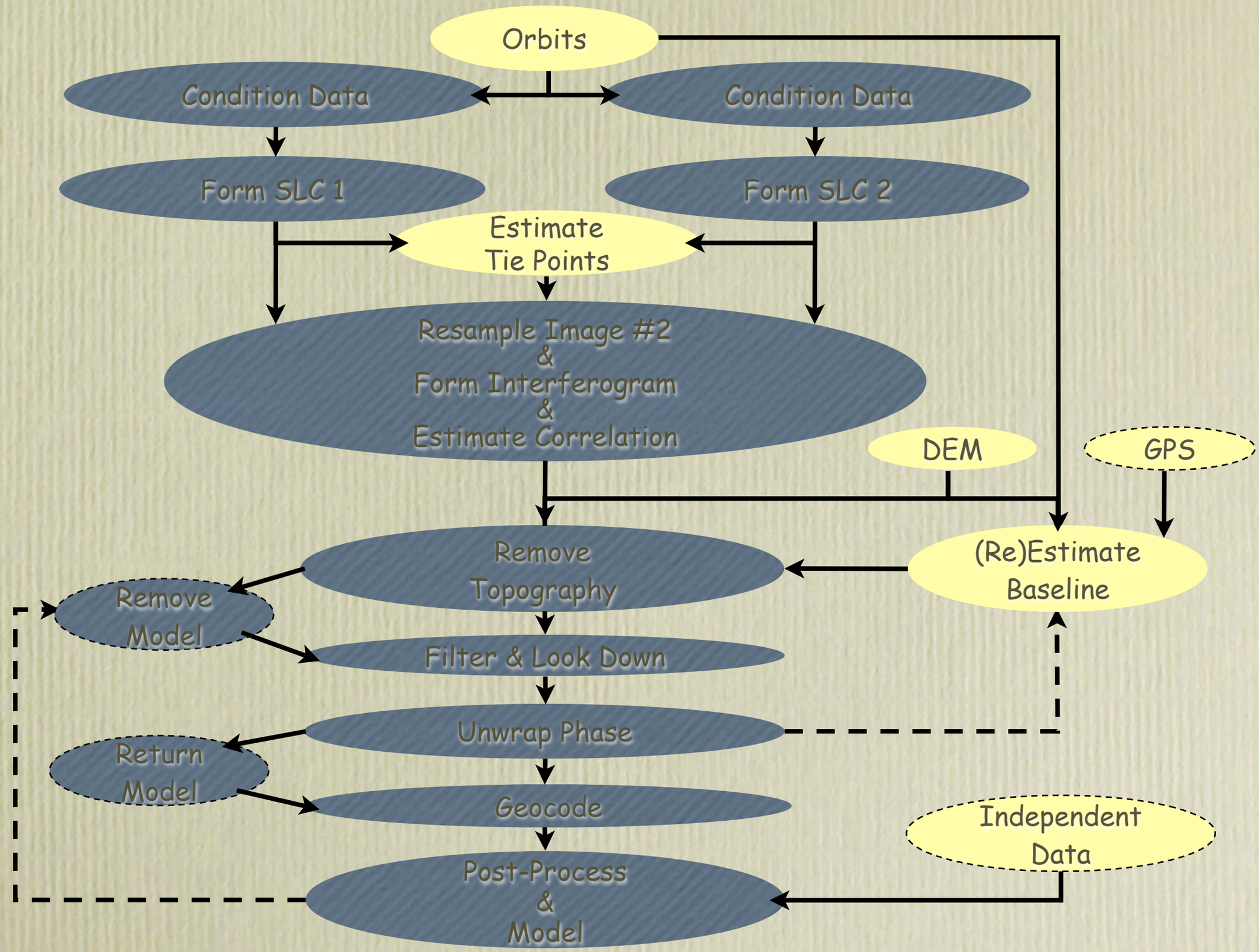
- Alternative unwrapping method "unw_method" = "icu" does approximately the same steps as the classic method of GZW88, but does it all in one program
- "int2filtunw.pl" calls "icu.pl" that sets up and runs "icu" program, putting output in "filt_$date1-$date2-sim_$OrbitType_${Rlooks_unw}rlks_c{10*UnwrappedThreshold}.unw"
- The "icu" program does the unwrapping in patches, so it can work on very large images that won't unwrap with classic programs, and it has additional techniques such as guiding centers for branch cuts
- Typically need to use much higher UnwrappedThreshold for "icu" such as 0.40

Unwrapping (cont'd)

- Another option is to use SNAPHU (Chen and Zebker, 2002), not included with ROI_pac but available from Stanford website
- SNAPHU unwraps everything, even where low (or no!)coherence
- "snaphu.pl" script can setup input file and run SNAPHU (will be in ROI_pac 3.0.1 release)

Chen, C.W., and Zebker, H.A., 2002, Phase unwrapping for large SAR interferograms: statistical segmentation and generalized network models: Geoscience and Remote Sensing, IEEE Transactions on,, v. 40, p. 1709-1719.

ROI_pac Two-pass Processing Flow



modified from Mark Simons

Baseline re-estimation

$DoItFrom “unwrapped"

- Default “flattening” is “orbit” where the baseline calculated from the orbits is used for the final products
- If “flattening” is “topo” then the unwrapped phase and the topography in the radar coordinates are used to refine the baseline length
- “process_2pass.pl” sets up files and calls “phase2base.pl” to estimate new “SIM” baseline
- “phase2base.pl” calls “baseest” estimate new “SIM” baseline and stores result in $date1-$date2_baseline.rsc

$EndItAt “redo_base”

Applying new baseline

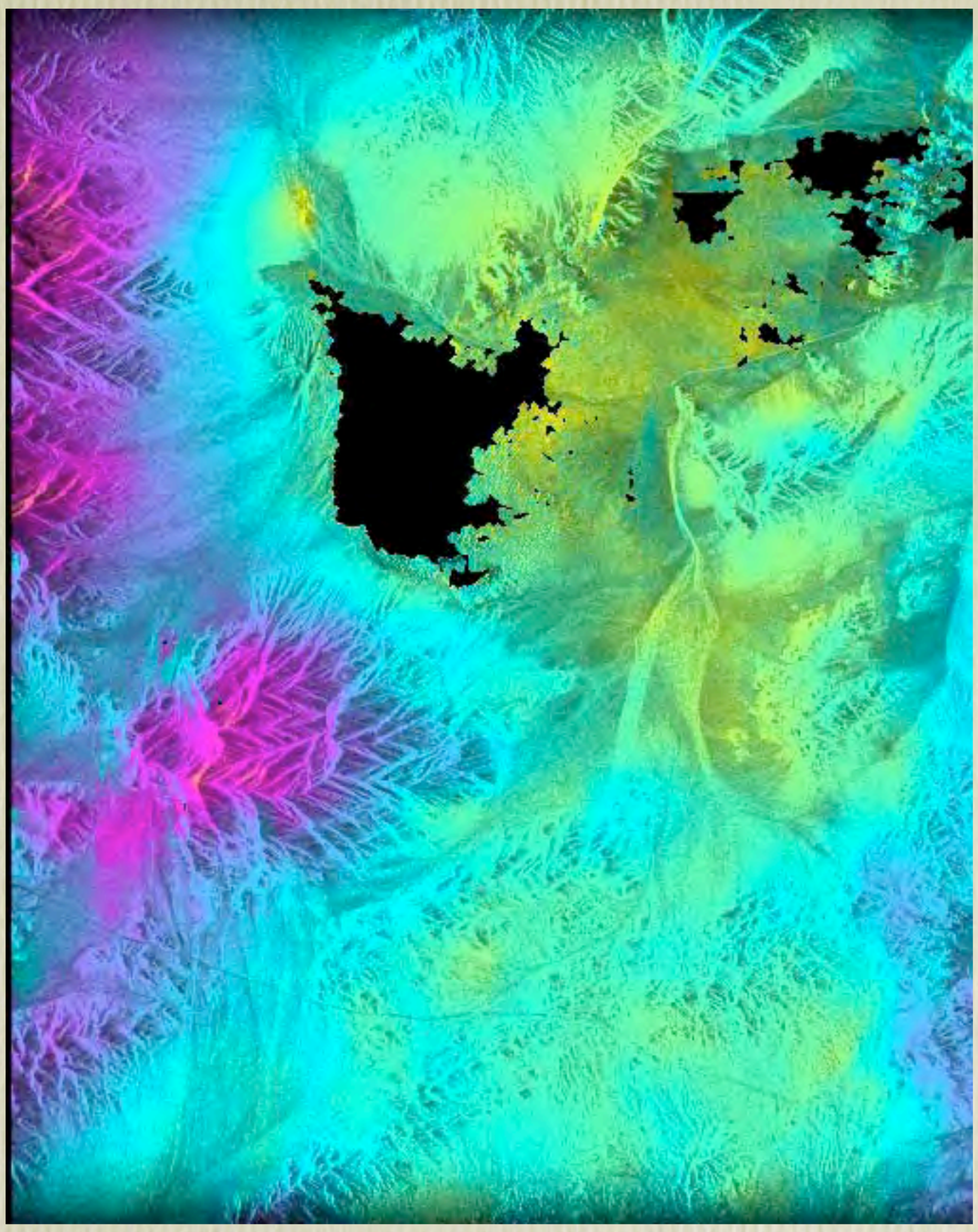
$DoItFrom “redo_base"

- “process_2pass.pl” calls “diffnsim.pl” again with new “SIM” baseline to subtract the phase due to the orbit and topography
- “phase2base.pl” calls “baseest” estimate new “SIM” baseline and stores result in $date1-$date2_baseline.rsc

$EndItAt “sim_removal_bsim”

Applying new baseline (cont'd)

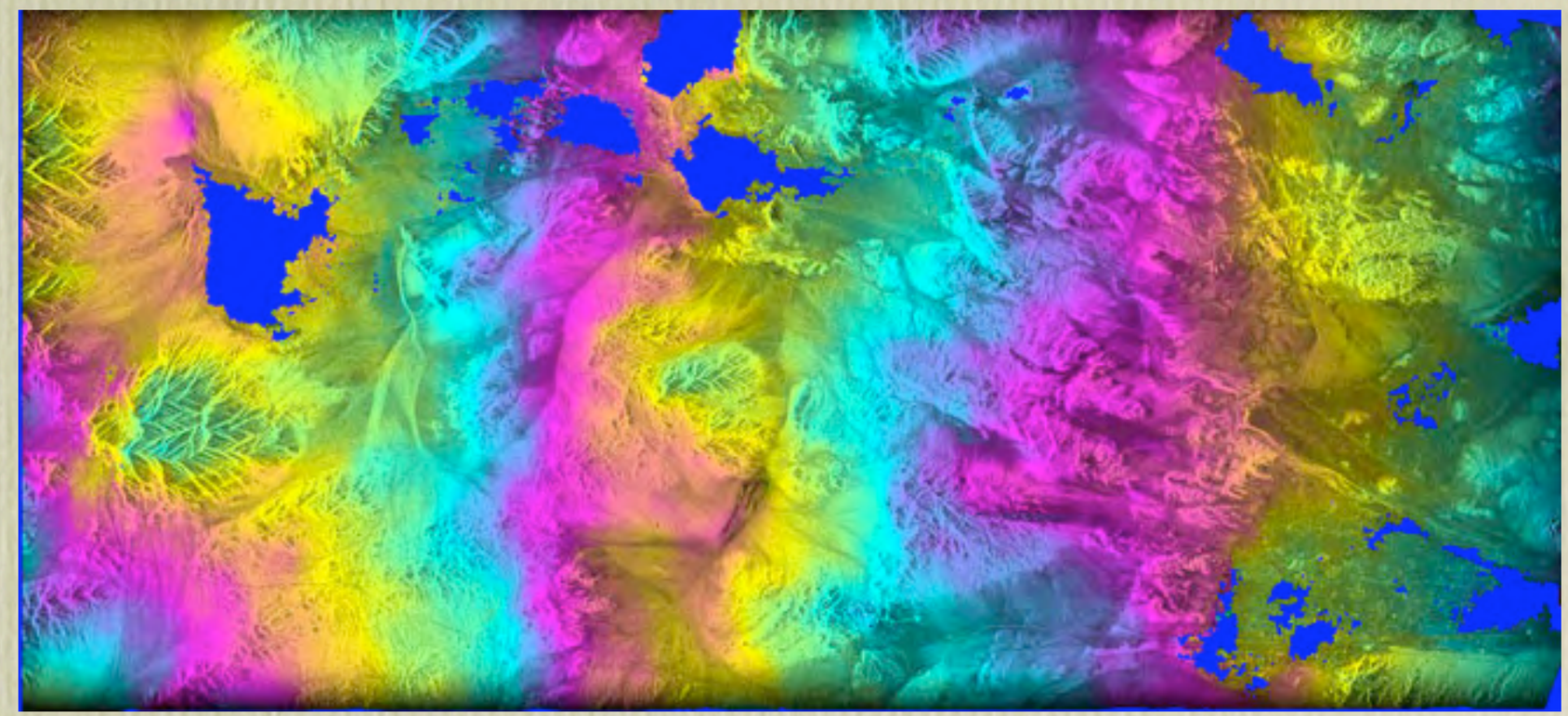
- "phase2base.pl" calls "baseest" estimate new "SIM" baseline and stores result in $date1-$date2_baseline.rsc
- example: filt_930110-950523-sim_SIM_4rlks_c10.unw (from ROI_pac test data with defaults)
- color wrap: 11.4 radians = 5 cm for C-band



$EndItAt "unwrapped_bsim"

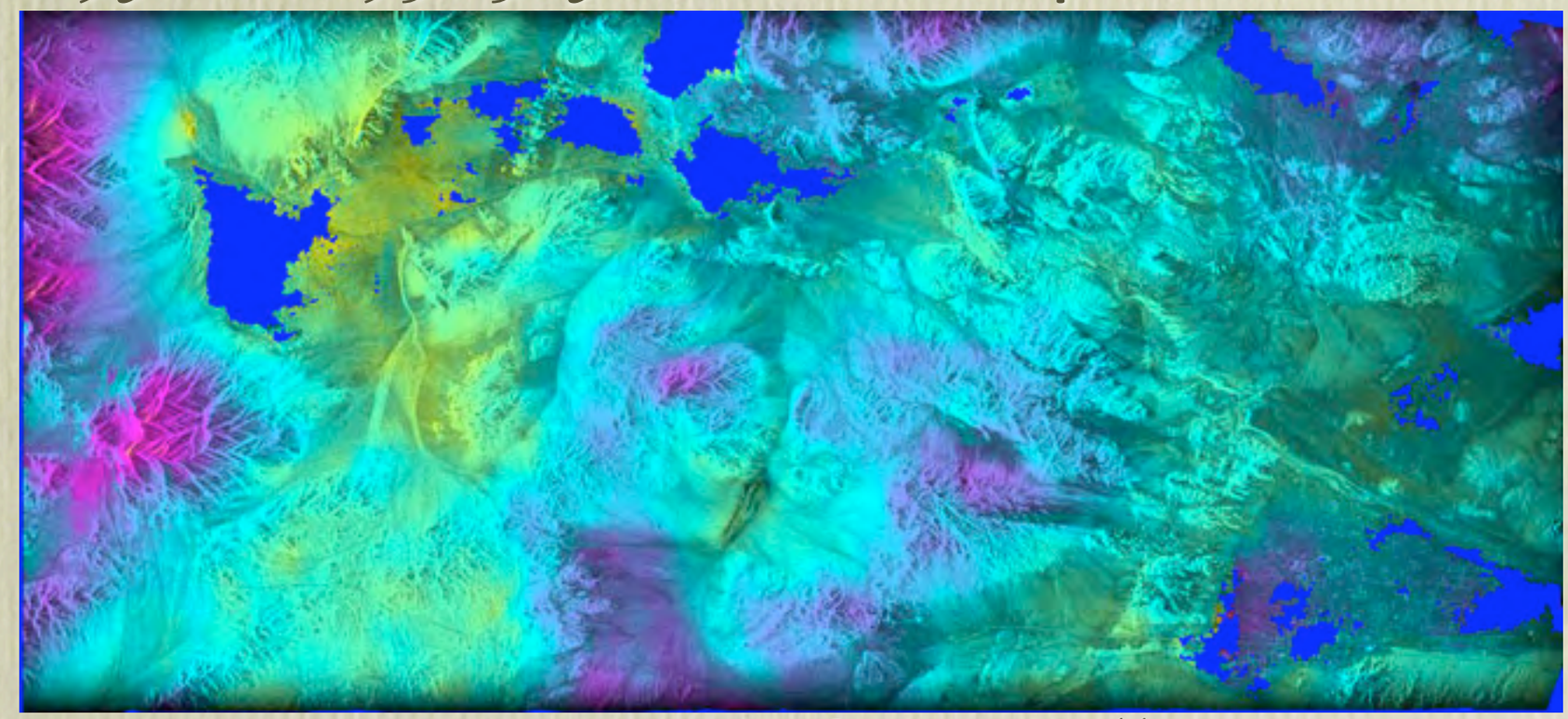
Baseline comparison

H_BASELINE_TOP_PRC
-23.257444795583 3
H_BASELINE_RATE_PRC
5.1081993245884 7e-05
H_BASELINE_ACC_PRC
-5.0454937562210 3e-14
V_BASELINE_TOP_PRC
-80.729422707311 8
V_BASELINE_RATE_PRC
1.8219590398637 2e-05
V_BASELINE_ACC_PRC
7.3277925805769 6e-13



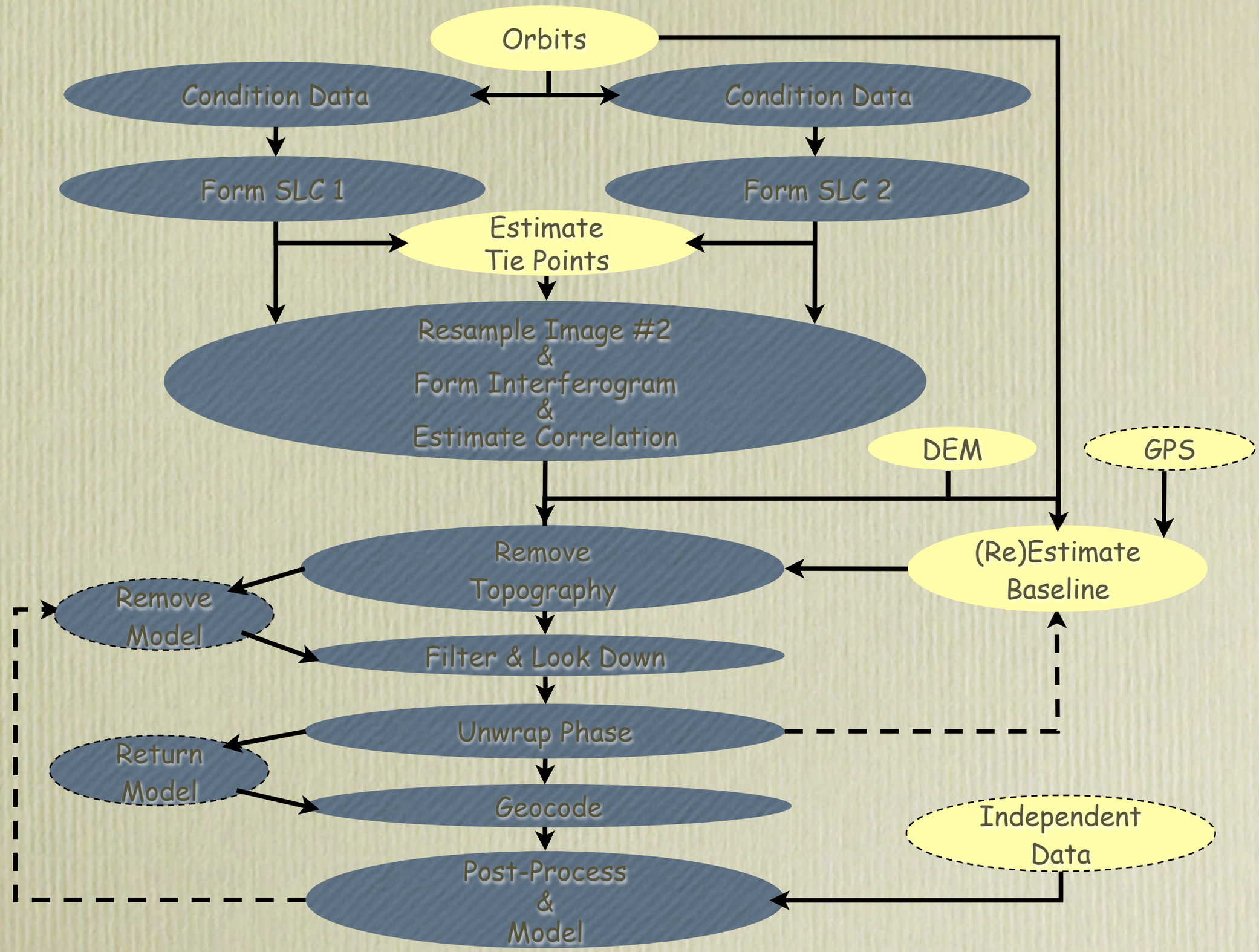
filt_930110-950523-sim_PRC_4rlks_c10.unw

H_BASELINE_TOP_SIM
-21.932151167651199
H_BASELINE_RATE_SIM
5.3348099880124 1687E-005
H_BASELINE_ACC_SIM
3.2059278013072 0777E-011
V_BASELINE_TOP_SIM
-80.302032316584373
V_BASELINE_RATE_SIM
1.7668313481262 4133E-005
V_BASELINE_ACC_SIM
-1.7862763712508 2563E-011



filt_930110-950523-sim_SIM_4rlks_c10.unw

ROI_pac Two-pass Processing Flow



modified from Mark Simons

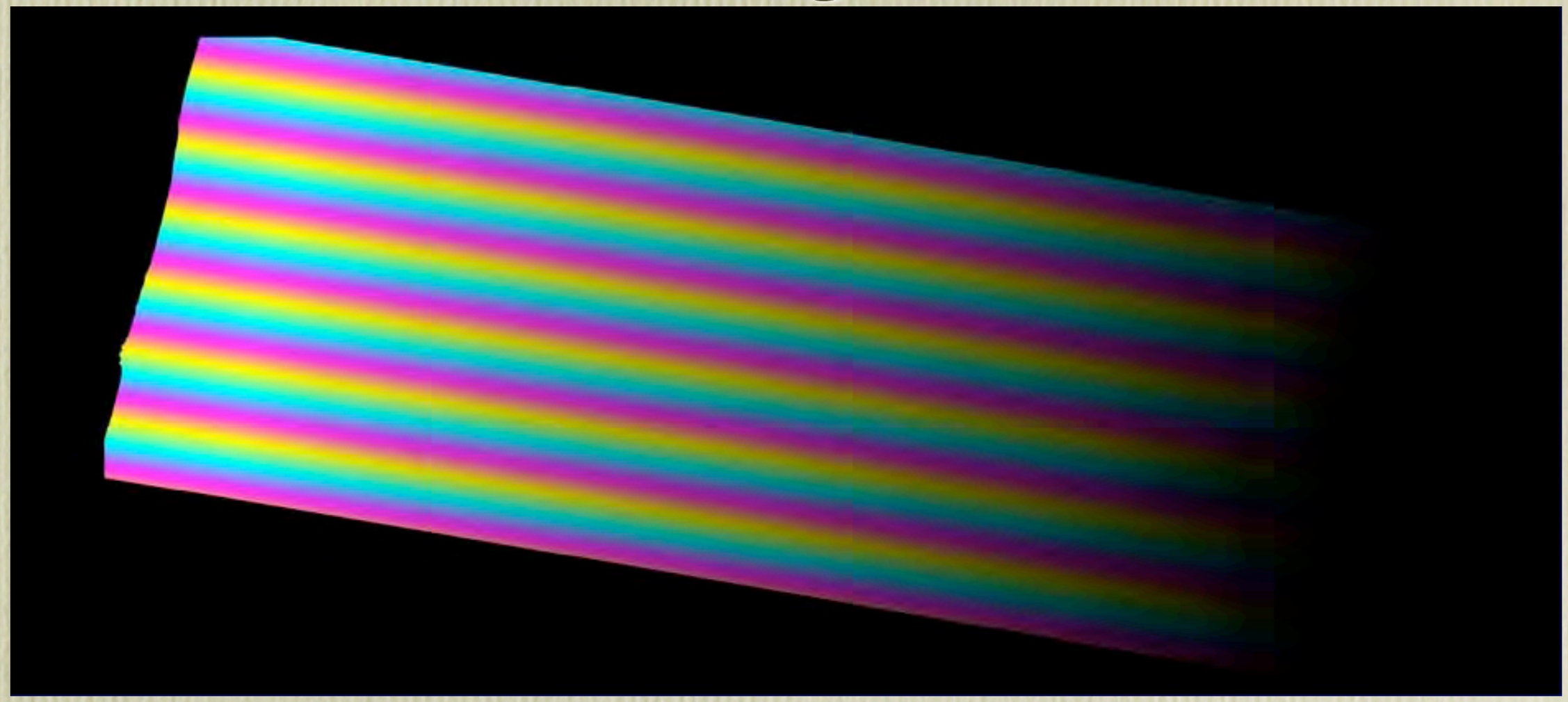
Geocoding

$DoItFrom “unwrapped"
or “unwrapped_bsim”

- “process_2pass.pl” calls “radar2geo.pl” to geocode final unwrapped phase from radar coordinates to geographic coordinates (same as used in DEM)
- geocoded result usually a subset of DEM area, but has same grid spacing and is shifted by an integer number of DEM pixels
- if coseismic or interseismic model was subtracted previously (if “MODEL” not “NULL”), then “process_2pass.pl” calls “geomodel.pl” to add back the model or models to the unwrapped phase and the result is geocoded

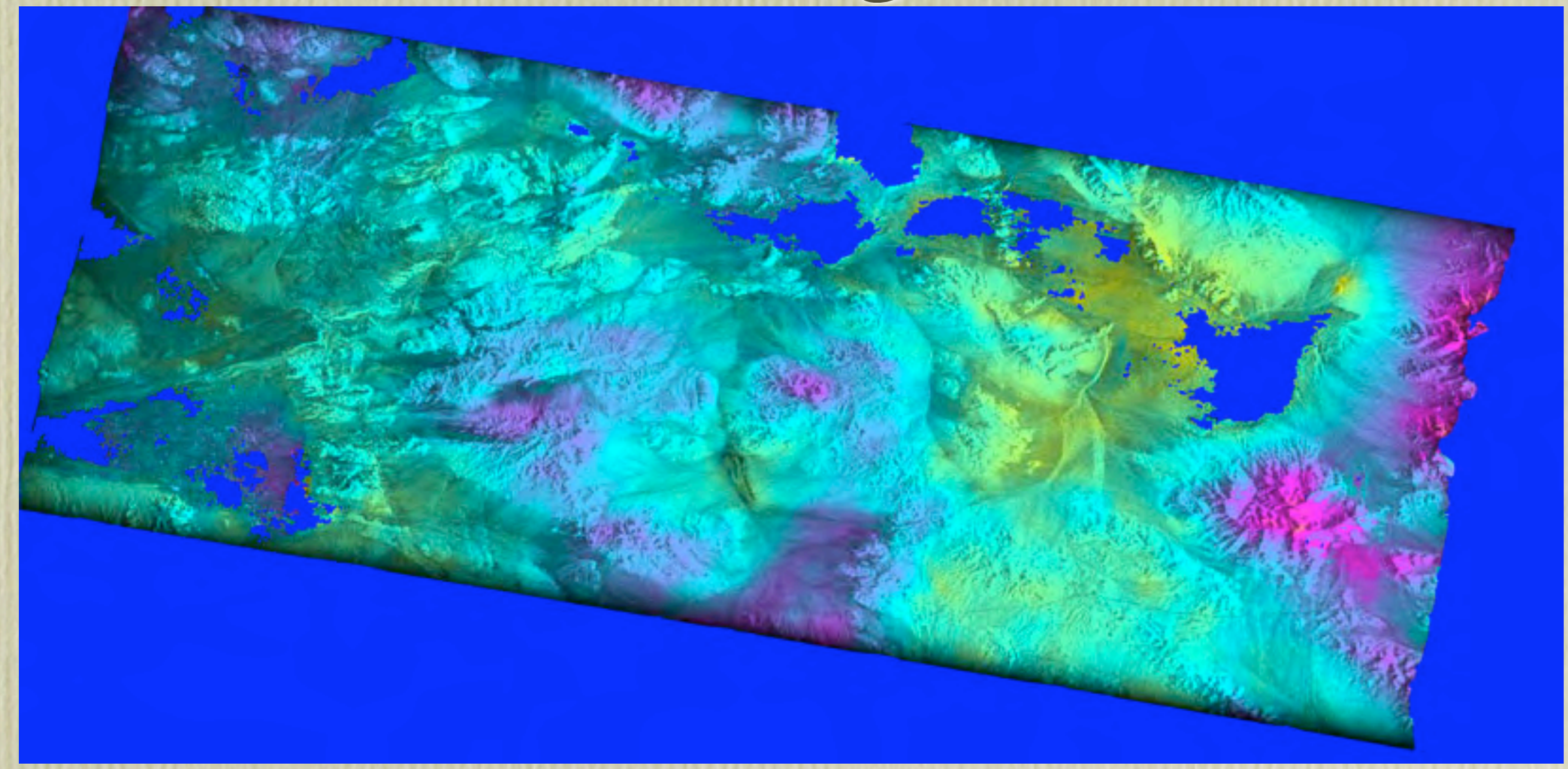
$EndItAt “done”

Geocoding (cont'd)



- "radar2geo.pl" calls "make_geomap.pl" to create the geocoding transformation lookup file "geomap_{Rlooks_unw}rlks.trans"
- "make_geomap.pl" moves to "GeoDir" and runs "IntSim" with the inverse of the affine transform from the simulation to the radar
- example: geomap_4rlks.trans (from ROI_pac test data with defaults)
- color wrap: 100 pixels

Geocoding (cont'd)



- final geocoded interferogram amplitude is zero where not unwrapped or outside interferogram
- example: geo_930110-950523.unw (from ROI_pac test data with defaults)
- color wrap: 11.4 radians = 5 cm

$EndItAt "done"