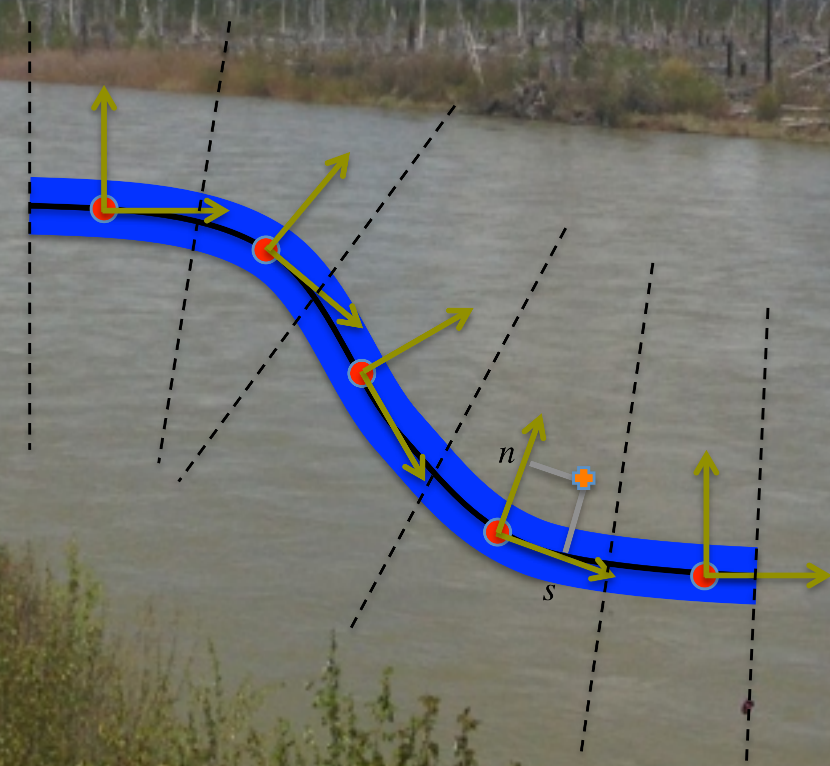
River Products from RiverObs

Michael Durand, Rui Wei and Renato Frasson

The Ohio State University
School of Earth Sciences
Byrd Polar & Climate Research Center



North Fork Flathead River
Montana

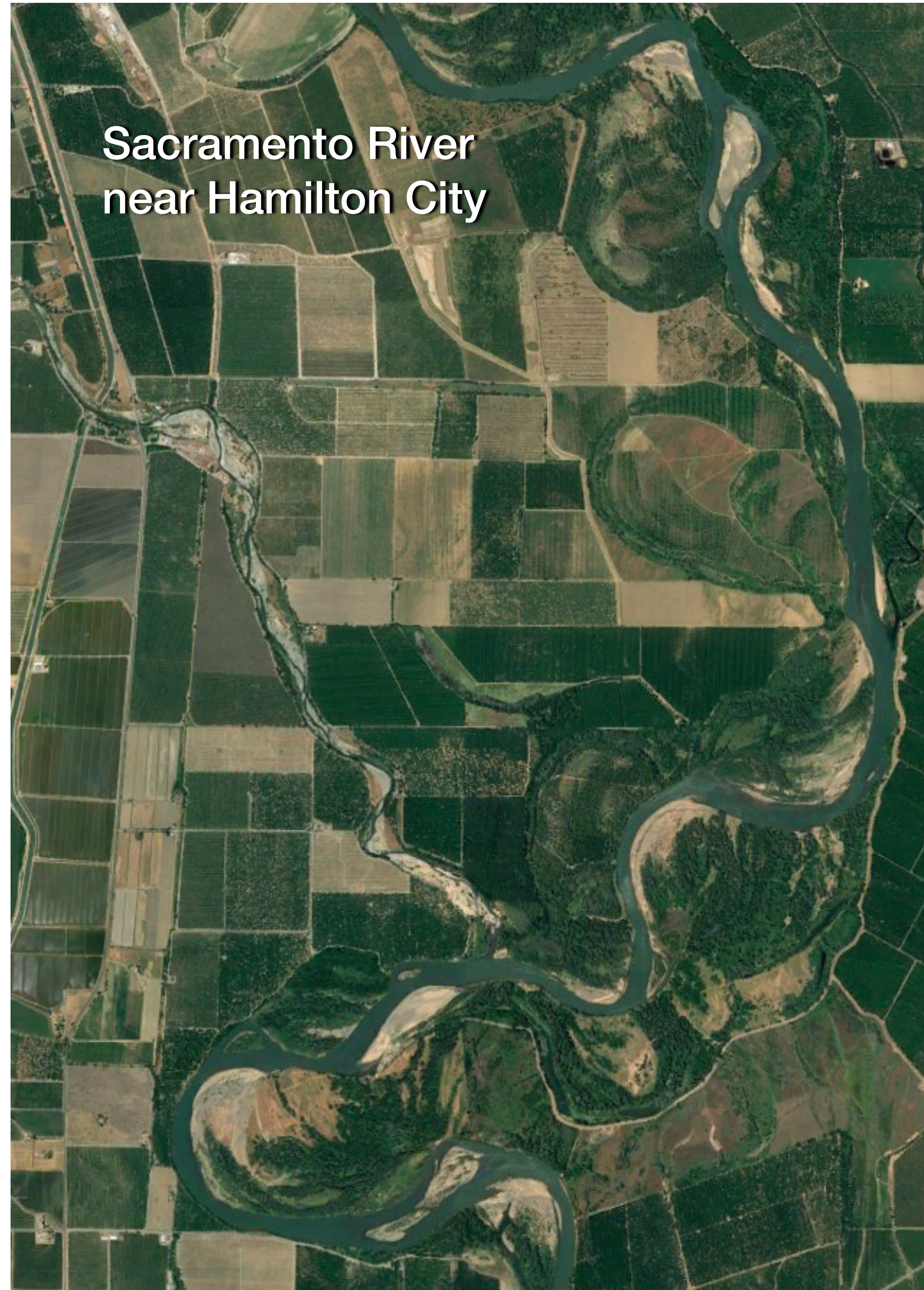
Overview

- Rivers are complex. Algorithms must be well-adapted to the variety found in nature

- We are simulating both simple and challenging cases, developing and coding algorithms, and perpetually testing

- Today: analysis of a Beta example data product on the Sacramento River.

- Future: Distribution of the Beta sample data product with representative format and expected errors



The needs for example river data products

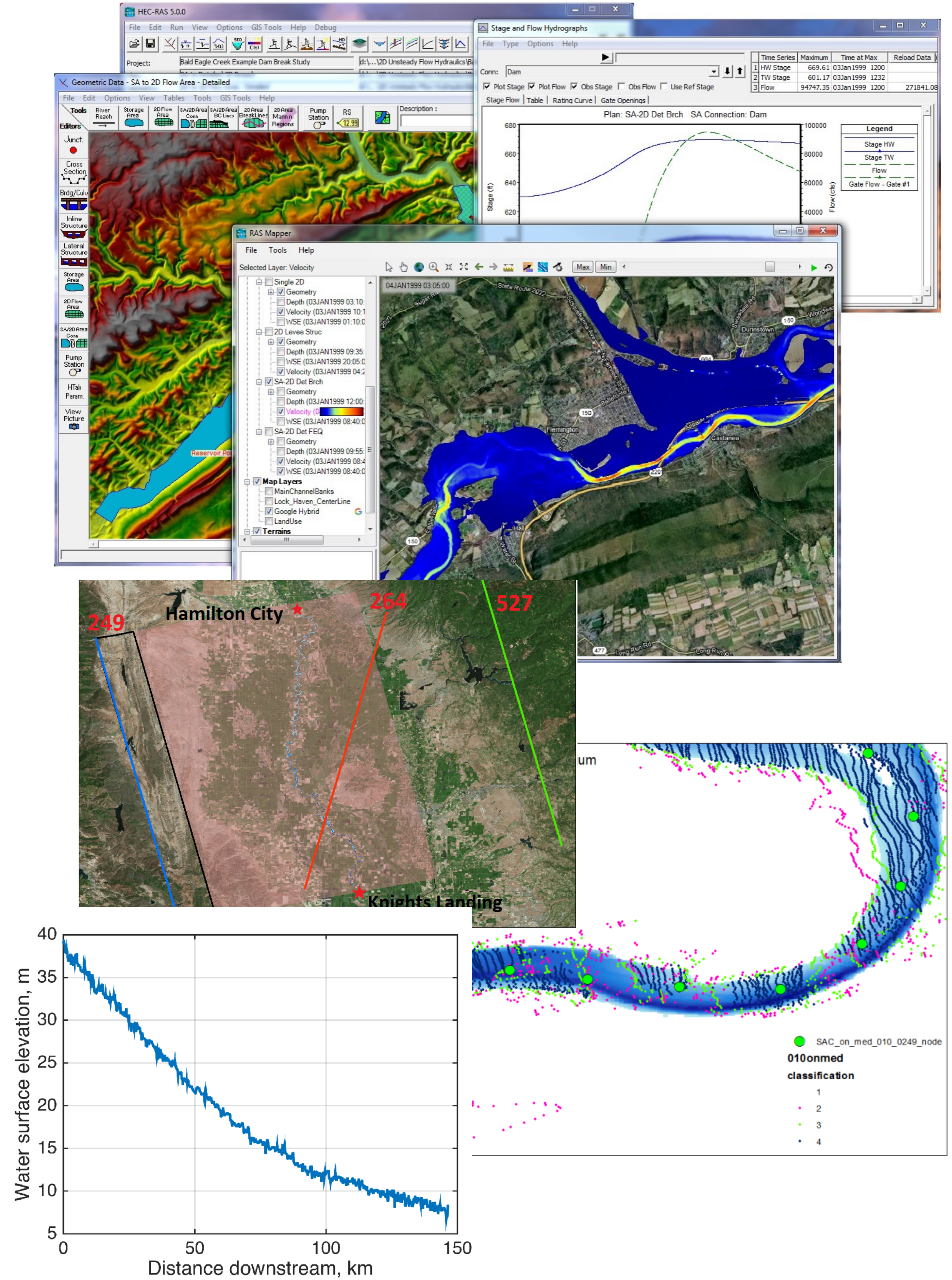
- Verifying Science Data System processing chains at JPL and CNES
- Ensuring data elements (including flags) meaningfully capture fluvial complexity
- Testing discharge algorithms
- Entraining new user communities

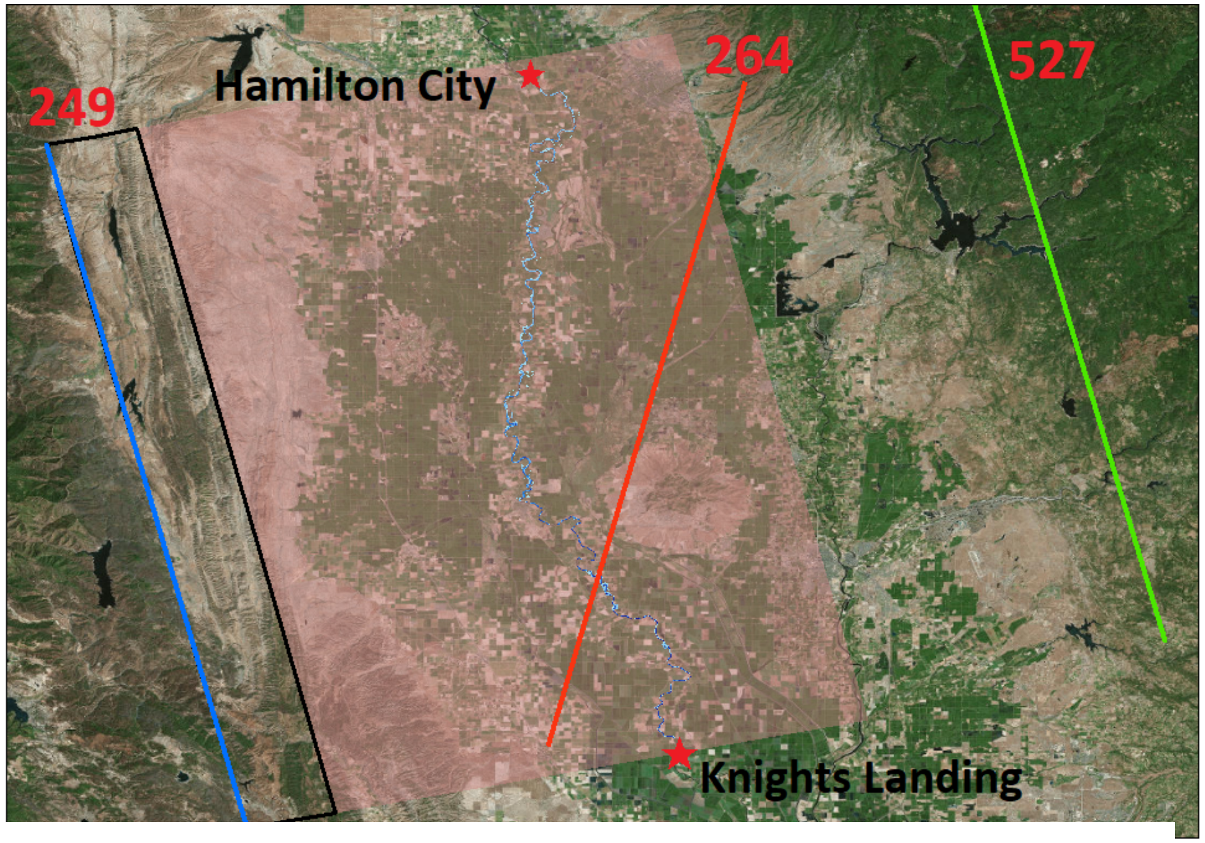
Process to develop example data products

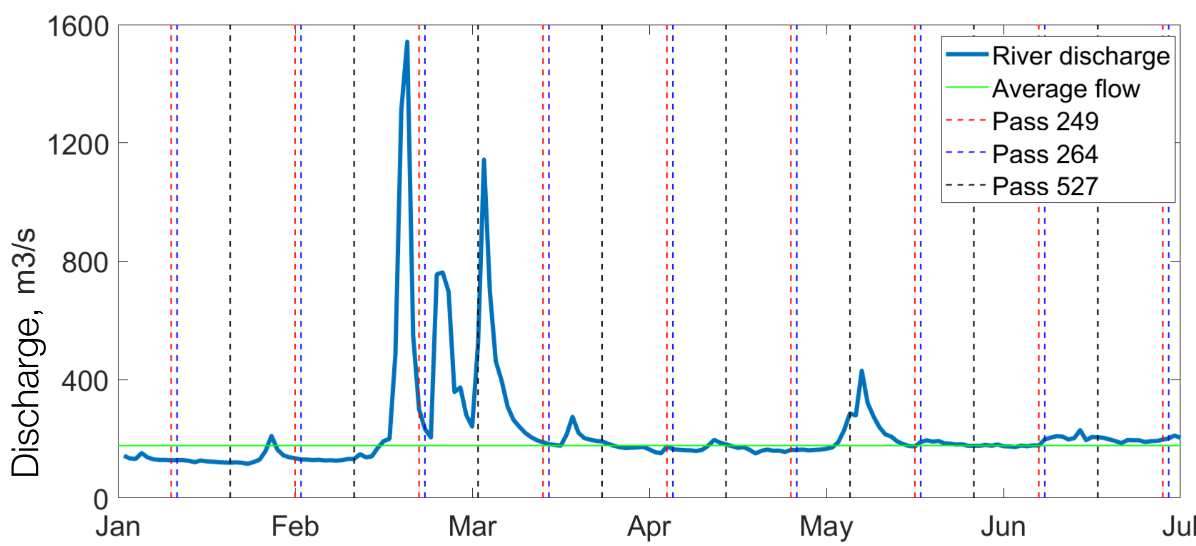
1. Two-dimensional, time-varying water level data produced by **hydraulic models**

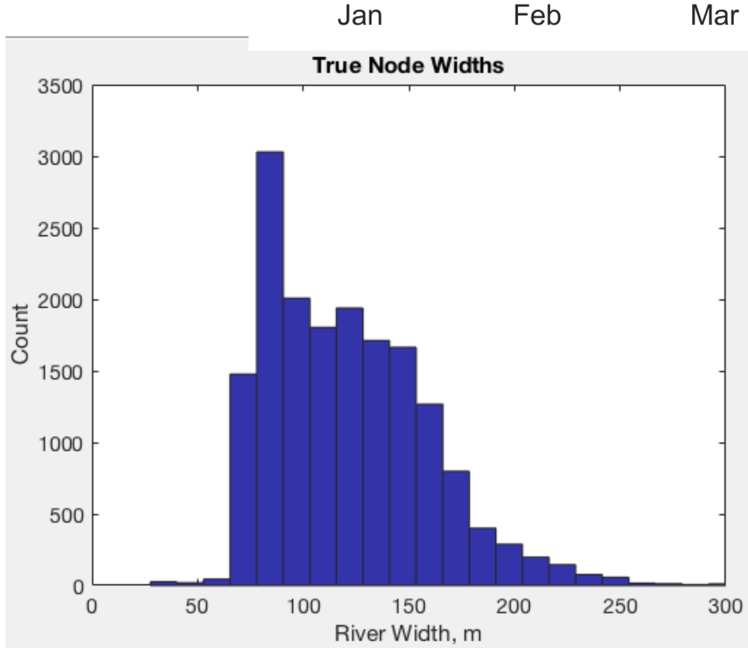
2. SWOT **Hydrology Simulator** computes separate pixel cloud for each pass

3. **RiverObs** maps the pixel cloud onto a centerline





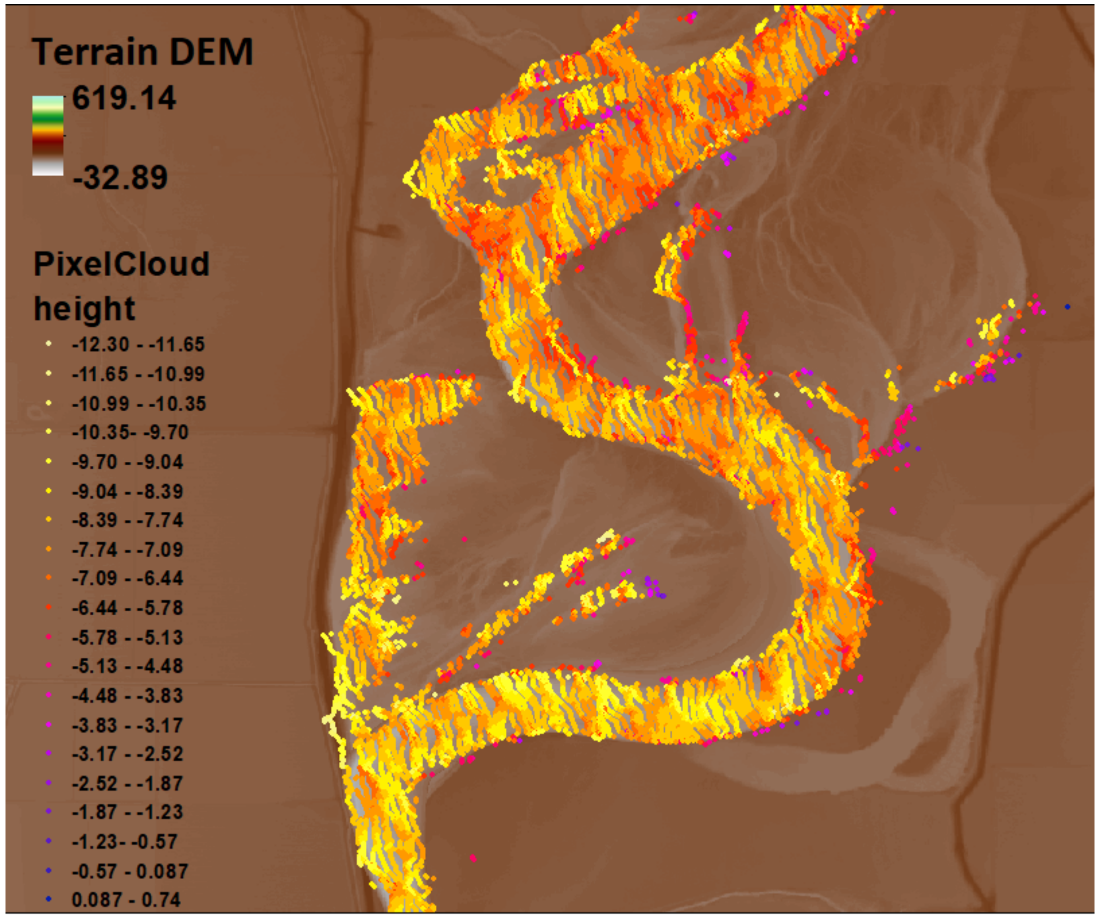


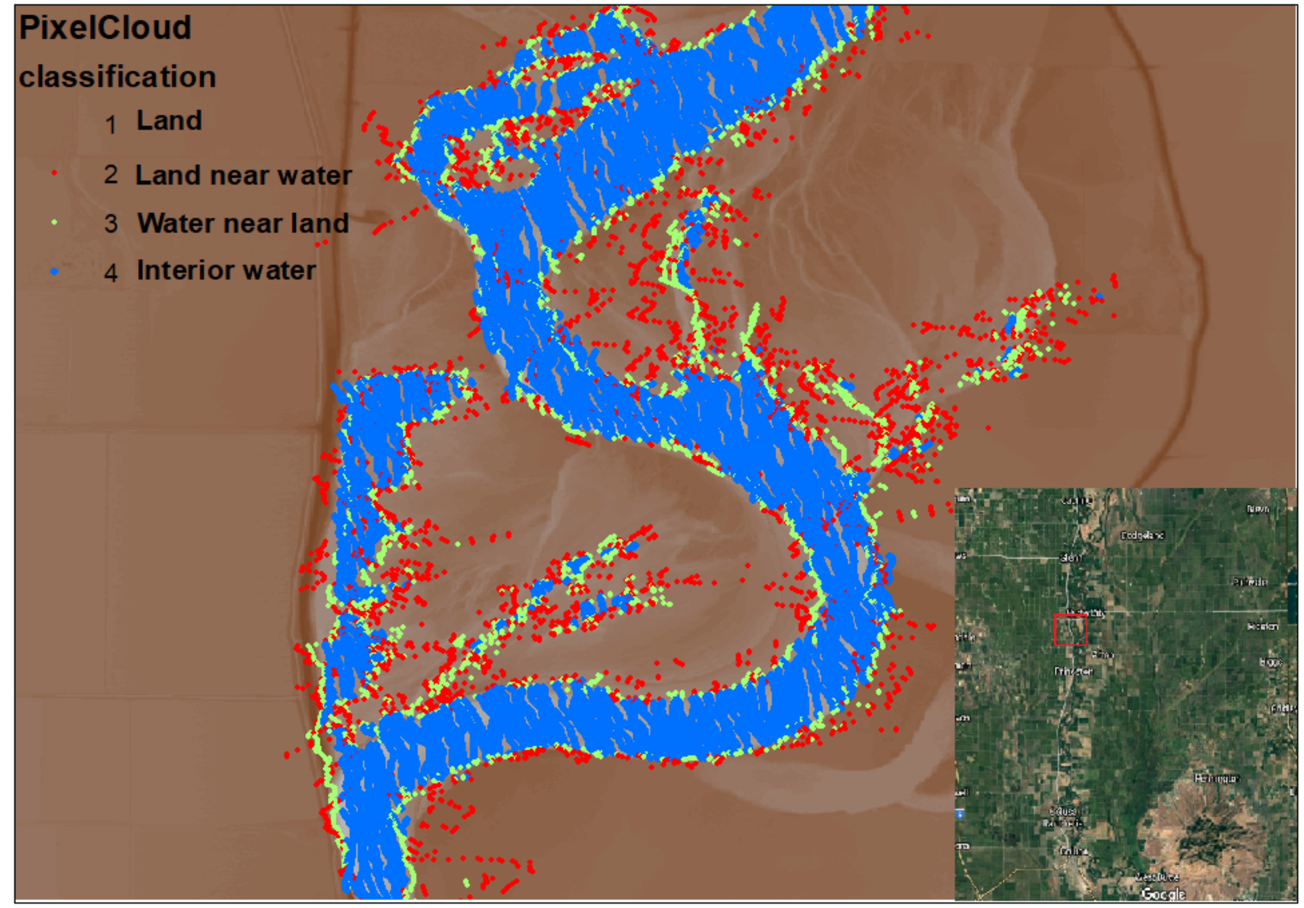


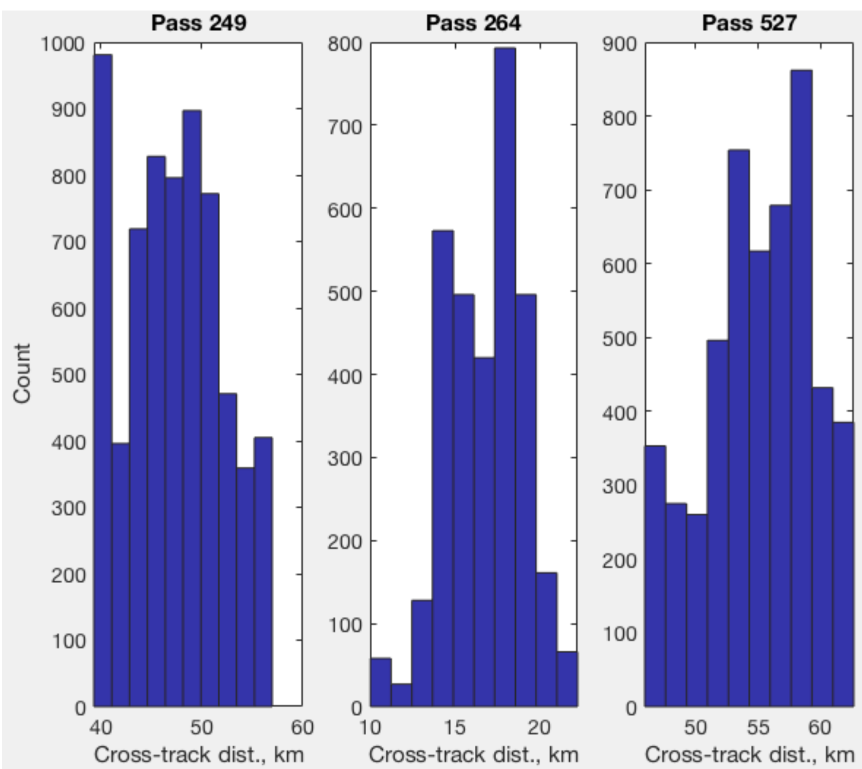
Upstream (shown) is more dynamic. Downstream is more channelized.

Study Domain

Domain: 147 km of the Sacramento River. Six months of simulation. Three passes, so 8-9 cycles. Widths: 122±42 m.







Note: Ellipsoidal heights are shown for the PixC

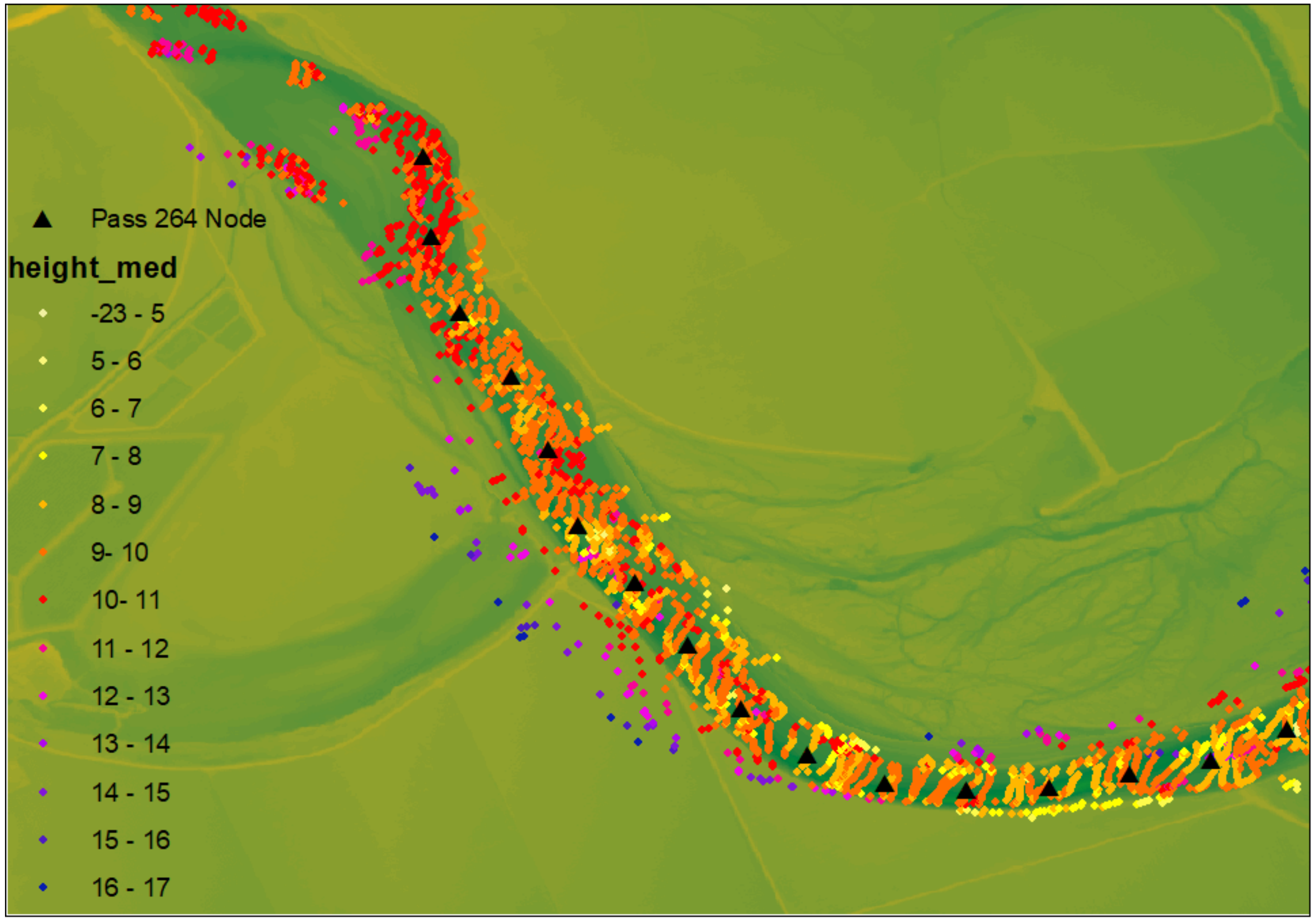
Cross-track distance mostly >40 or<25 km. Above are data from ~45 km cross-track distance.

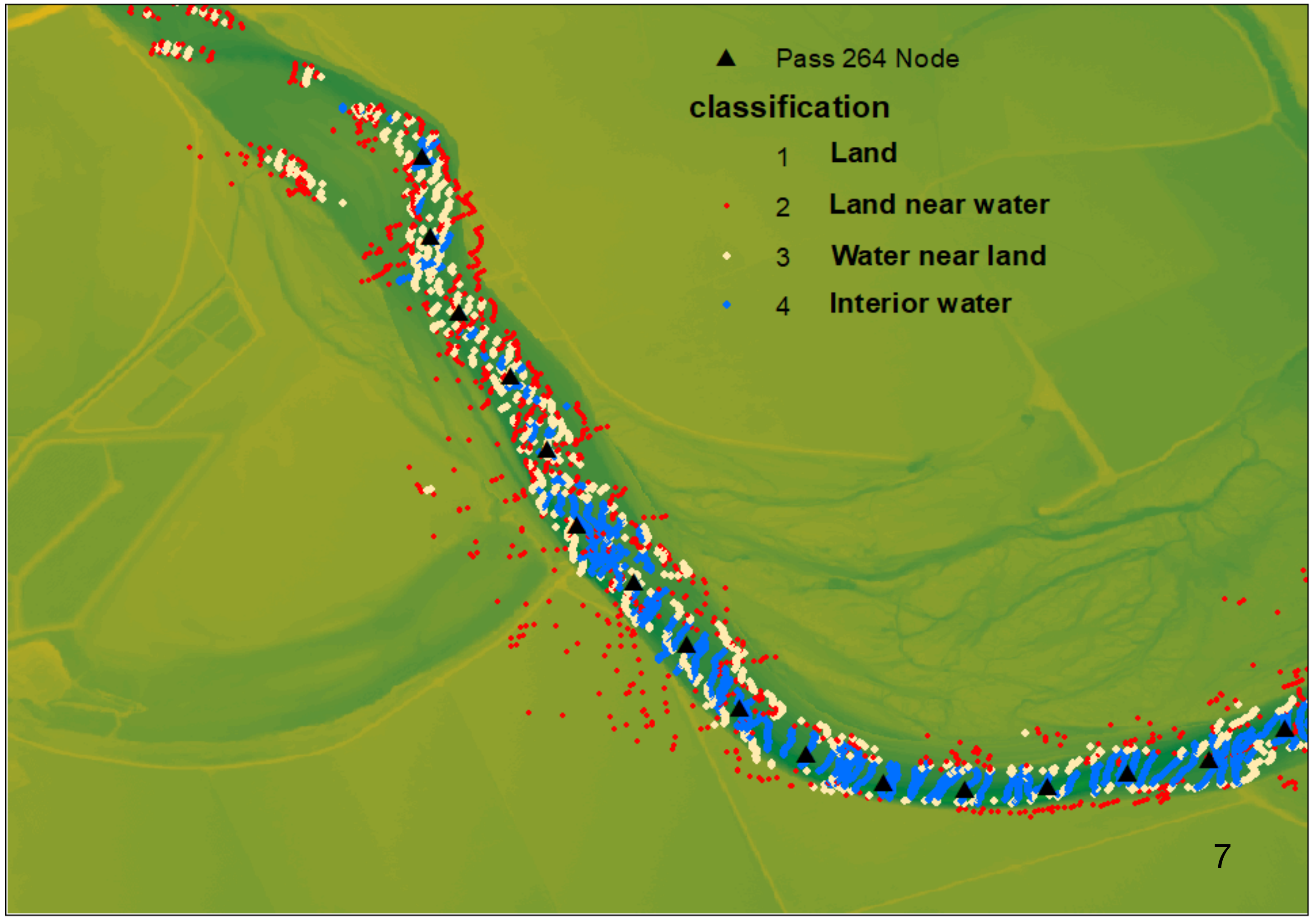
- Water: 10 dB. Land: -5 dB
- Medium pixel cloud
- Layover errors simulated physically
- Wet troposphere and instrument (e.g. roll) errors simulated statistically.
- No dark water. No riparian vegetation.

Hydrology Simulator produces the Pixel Cloud (PixC)

Example PixC in the Near Range

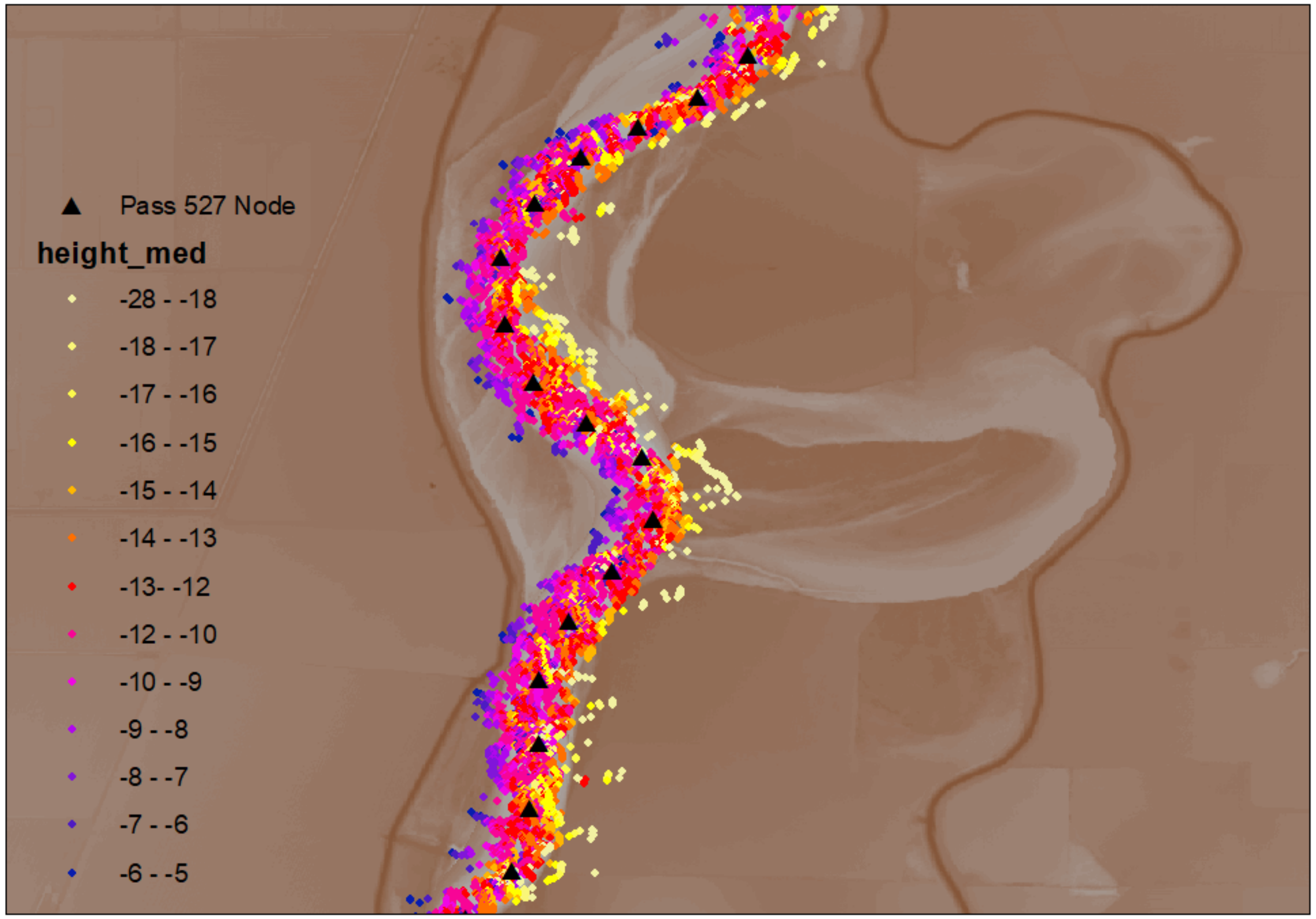
This example at ~20 km cross-track distance shows some gaps in pixC coverage of the river, due to larger pixel sizes in the cross-track direction. Heights for classes 2 & 3 are generally precise.

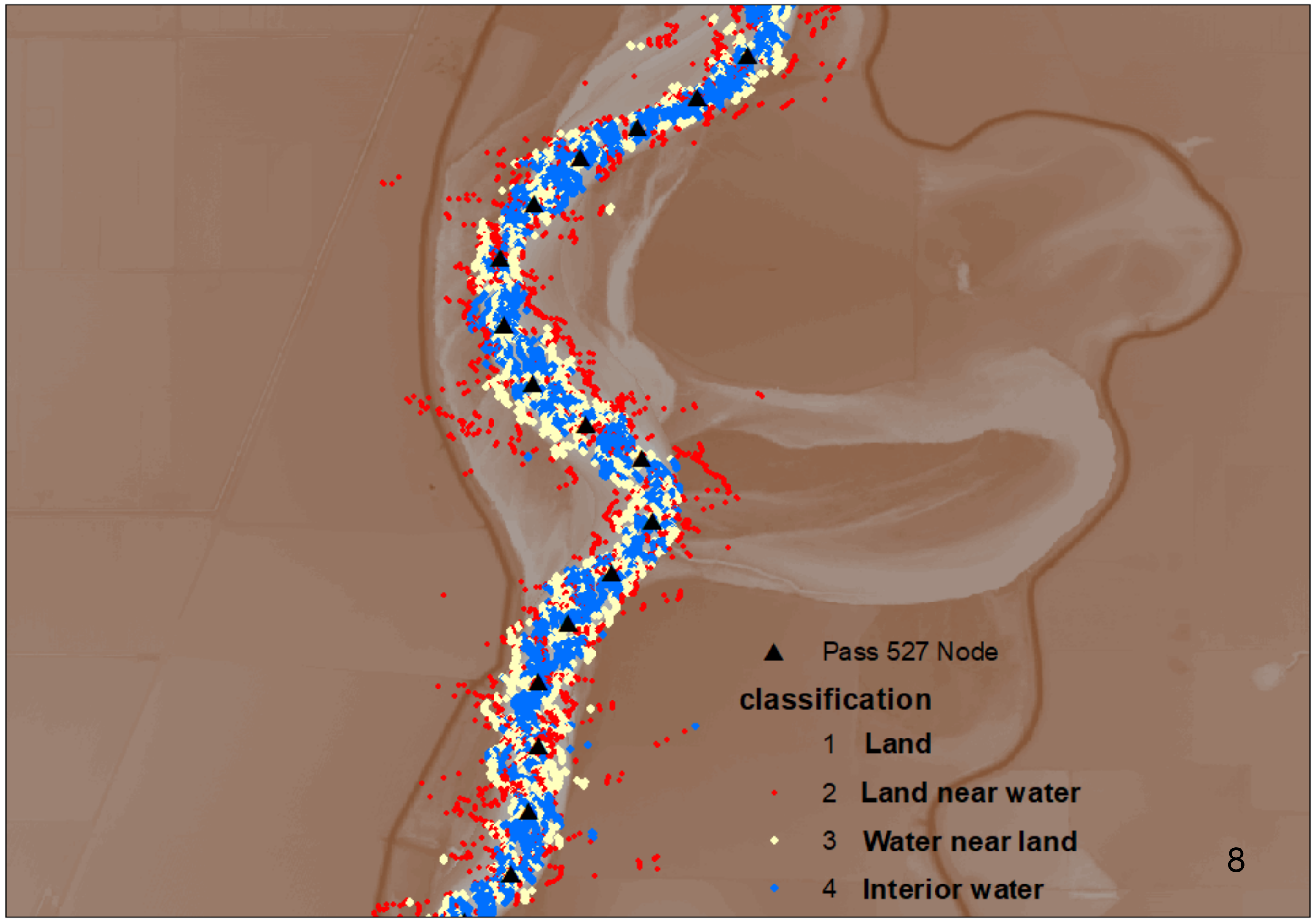




Example PixC in the Far Range

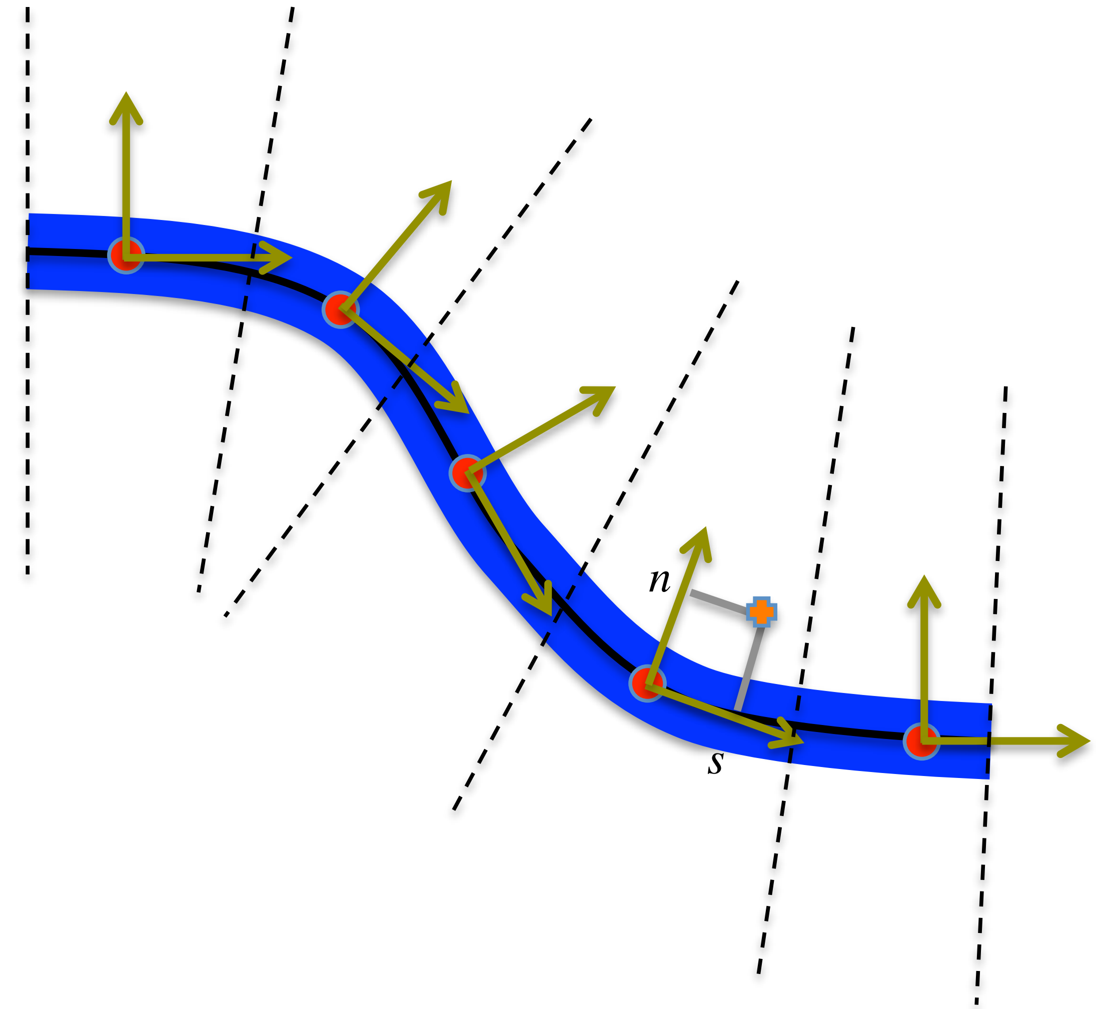
This example at ~60 km cross-track distance shows dense pixC coverage of the river, due to smaller pixel sizes in the cross-track direction. Heights for classes 2 & 3 are less precise.

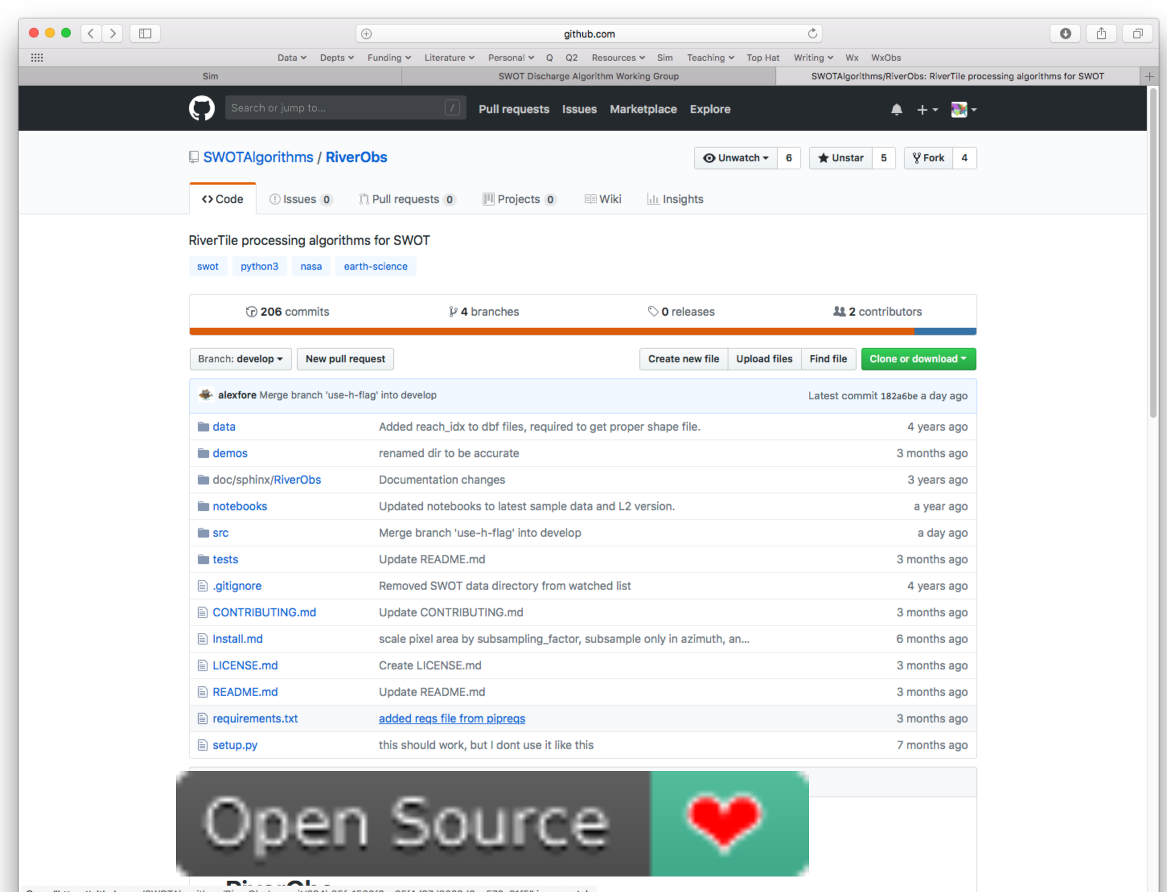




RiverObs maps PixC onto centerline

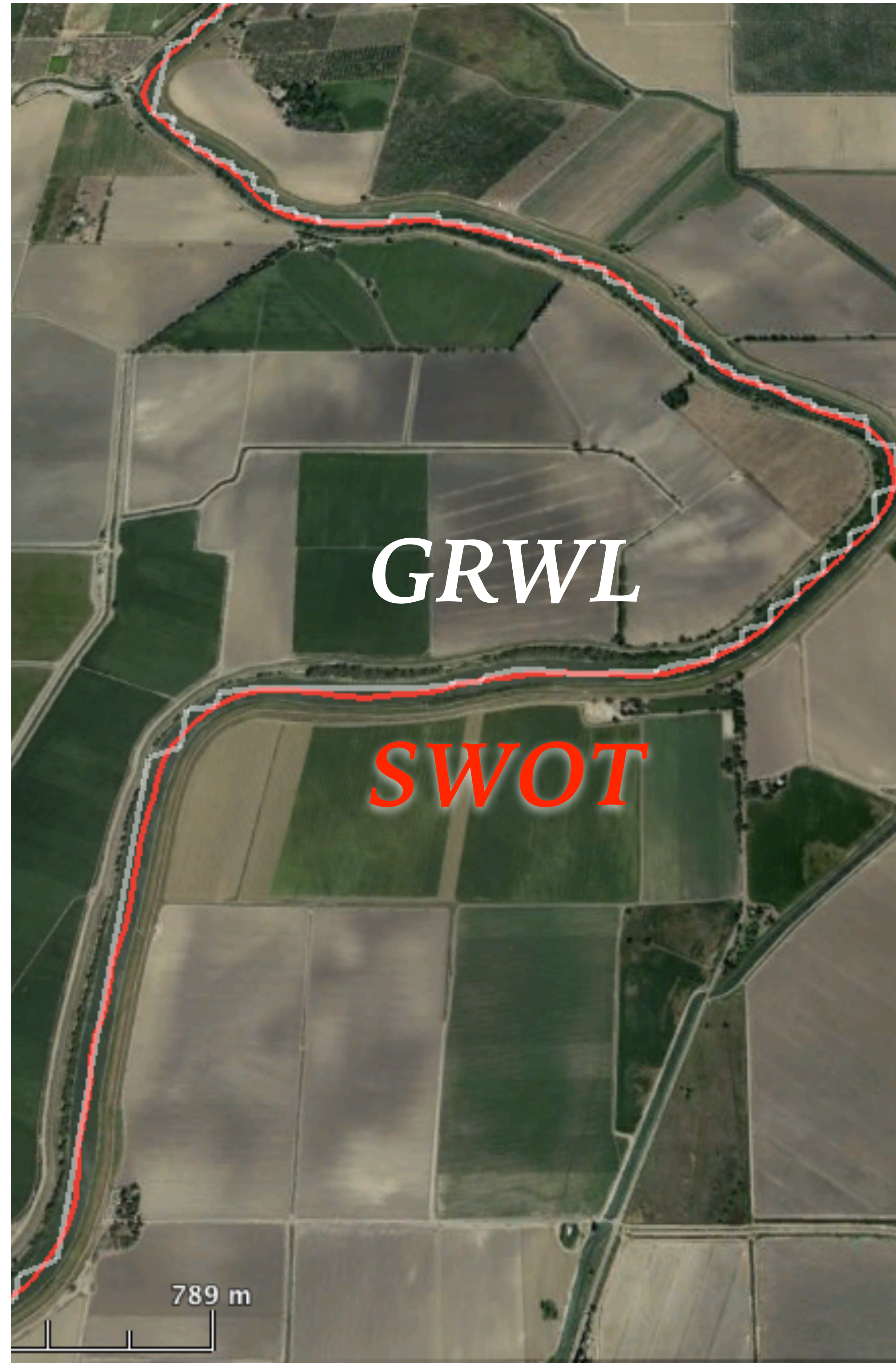
- RiverObs maps pixels onto a river centerline.
- RiverObs is open source, and available for download at: github.com/SWOTAlgorithms/RiverObs
- RiverObs is the core of the river “tile processor” in the official processing chain
- Originally by Ernesto Rodriguez. Now developed collaboratively.
- RiverObs version used to produce this dataset is available (not master branch), but requires v2 of the a priori database, which is not available globally. Contact durand.8@osu.edu with questions.





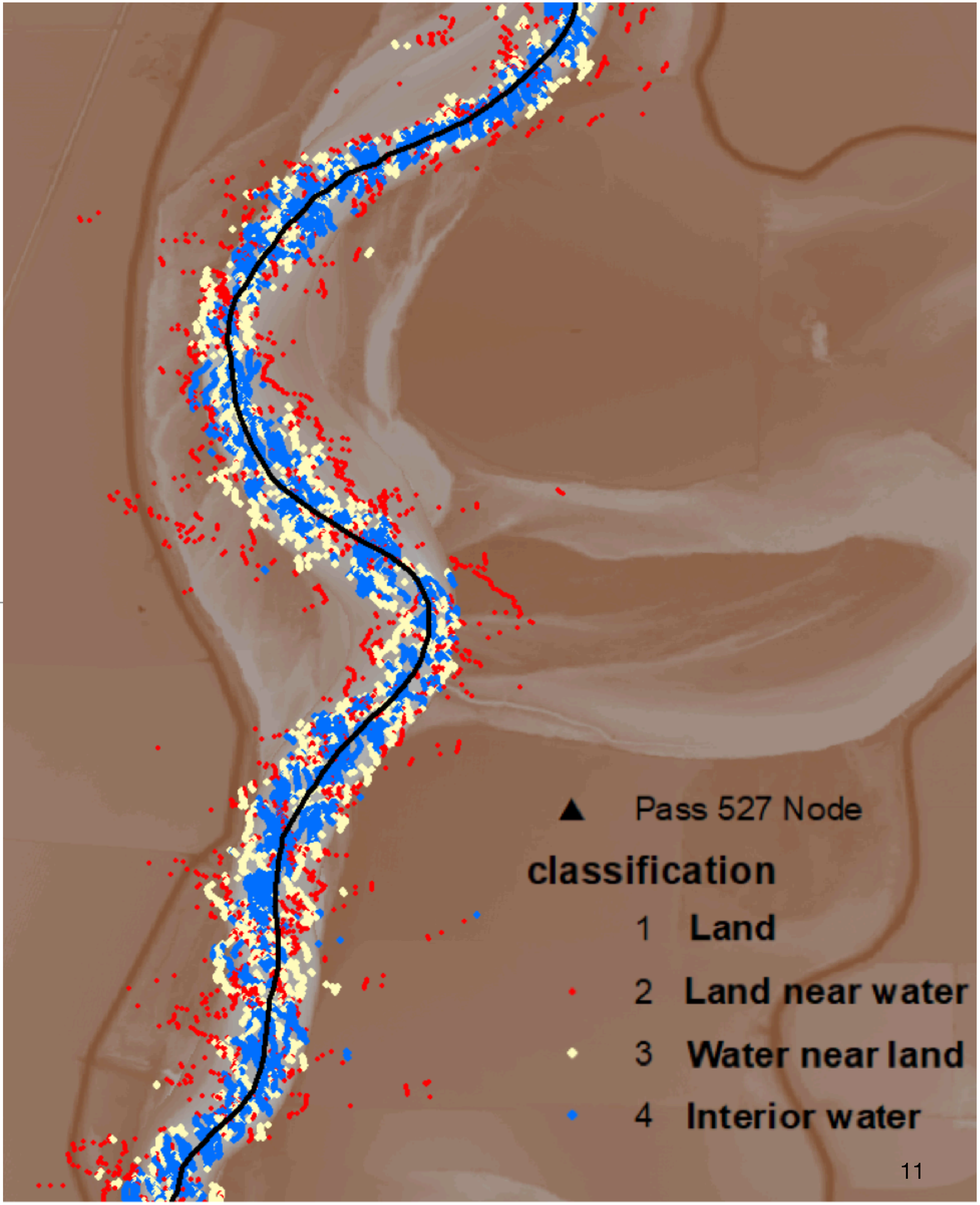
The a priori dataset and RiverObs

- Initial centerline defined as points along Global River Width from Landsat (GRWL). See talk by Tamlin Pavelsky, Day 2 Splinter, 2pm.
- Centerline refined offline using RiverObs run on merged low-flow PixC (30 m posting)
- Nodes are defined every 200 m
- Reaches are computed by aggregating nodes to ~10 km based on SWOT overpasses, tributaries, features. Here we used sinuosity [*Frasson et al.*, 2017].
- Cross-sectional area and discharge parameters also stored in the a priori database



Mapping pixels to centerline nodes

Each pixel is mapped to nearest node located at 200 m intervals along the centerline

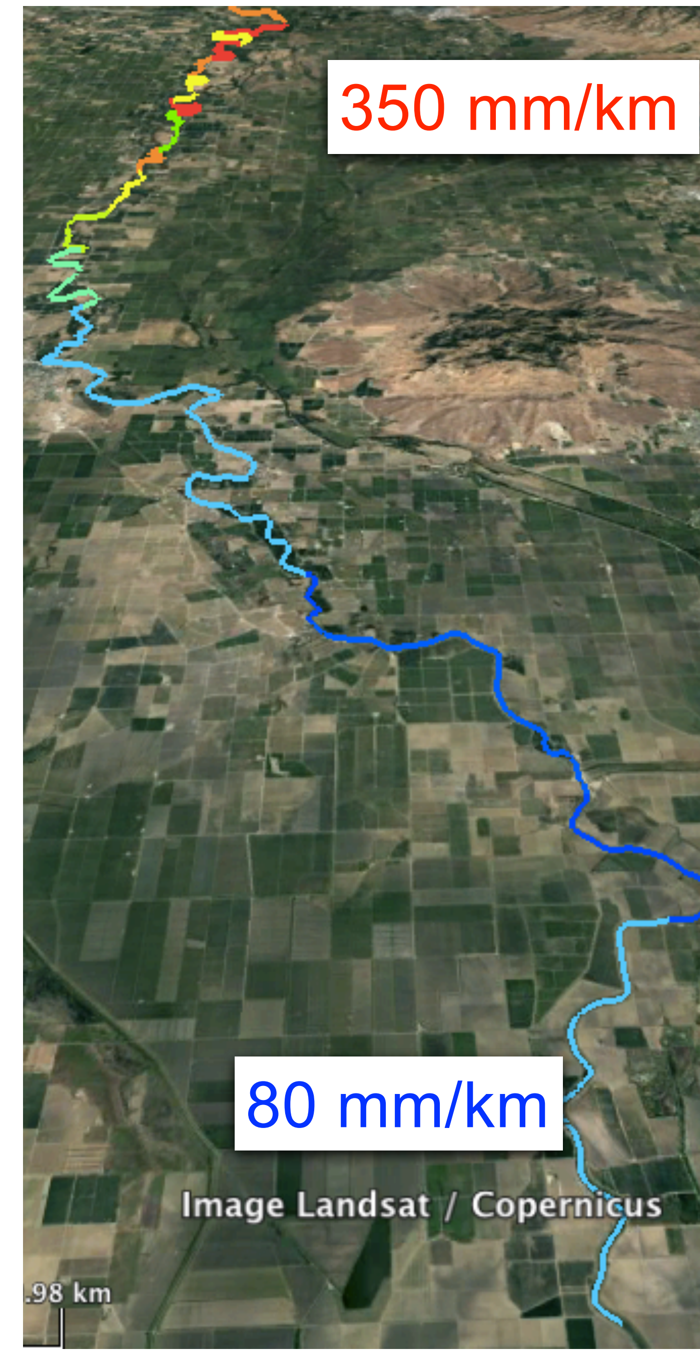


RiverObs: From pixels to centerline

- Pixels are mapped to nodes in the a priori node database
- To compute node elevations, only so-called “interior water”, and “water-near-land” are used. This avoids ~10 cm bias (equal to entire reach error budget!) for the Sacramento
- To compute width elevations, a third class (“land near water”) is used in addition
- Currently, laid-over pixels are used to compute node heights. Their exclusion generally makes things worse

Node Heights

Reach Slopes

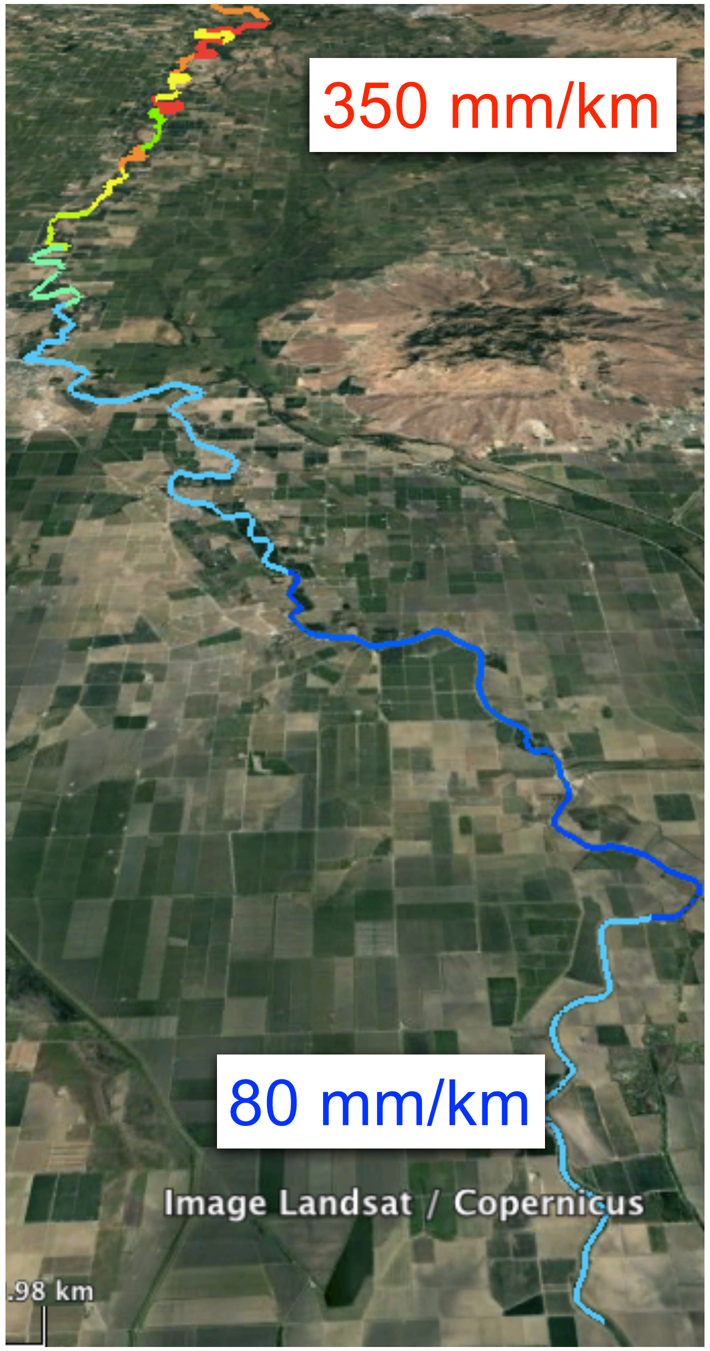


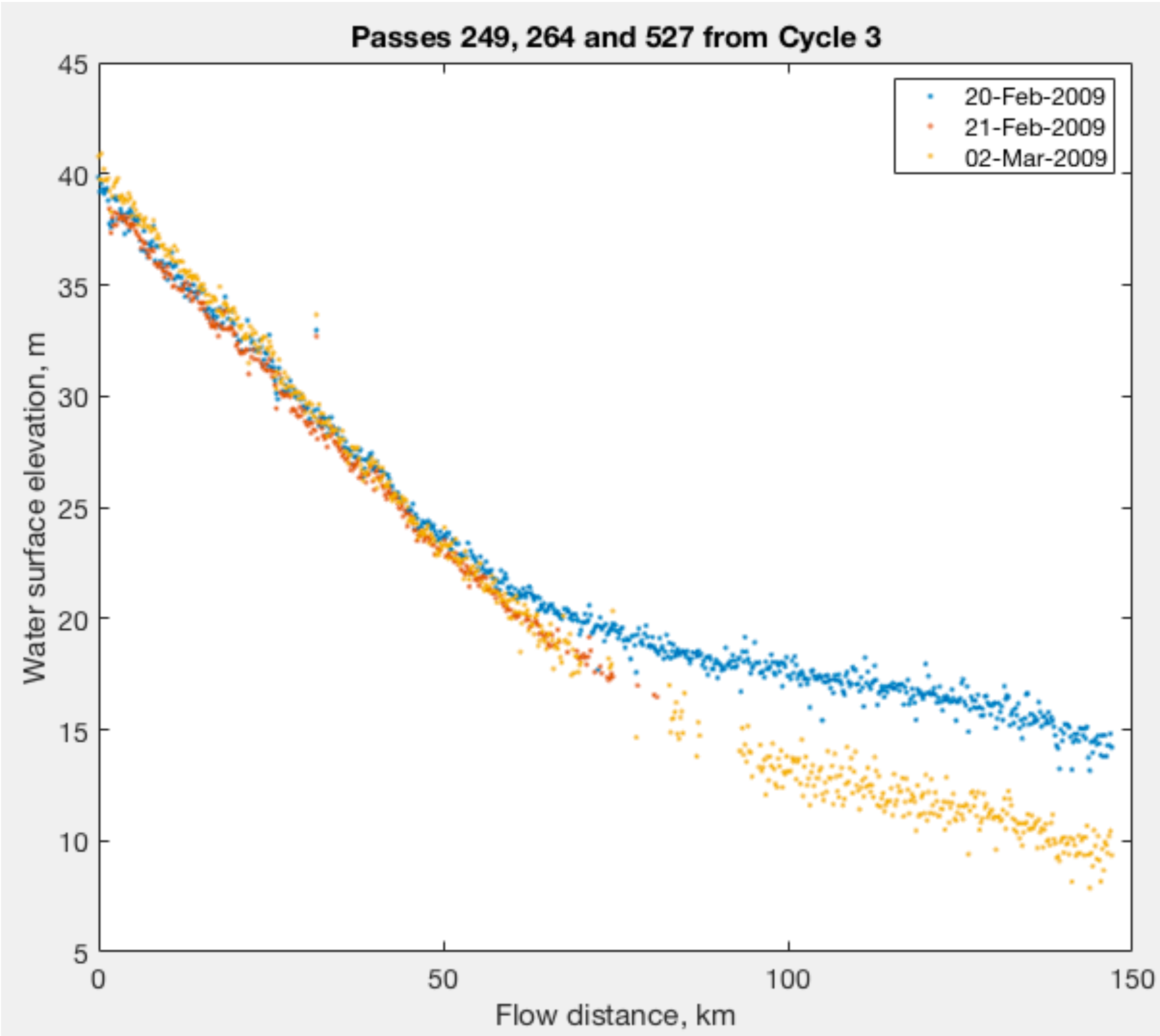
RiverObs: From pixels to centerline

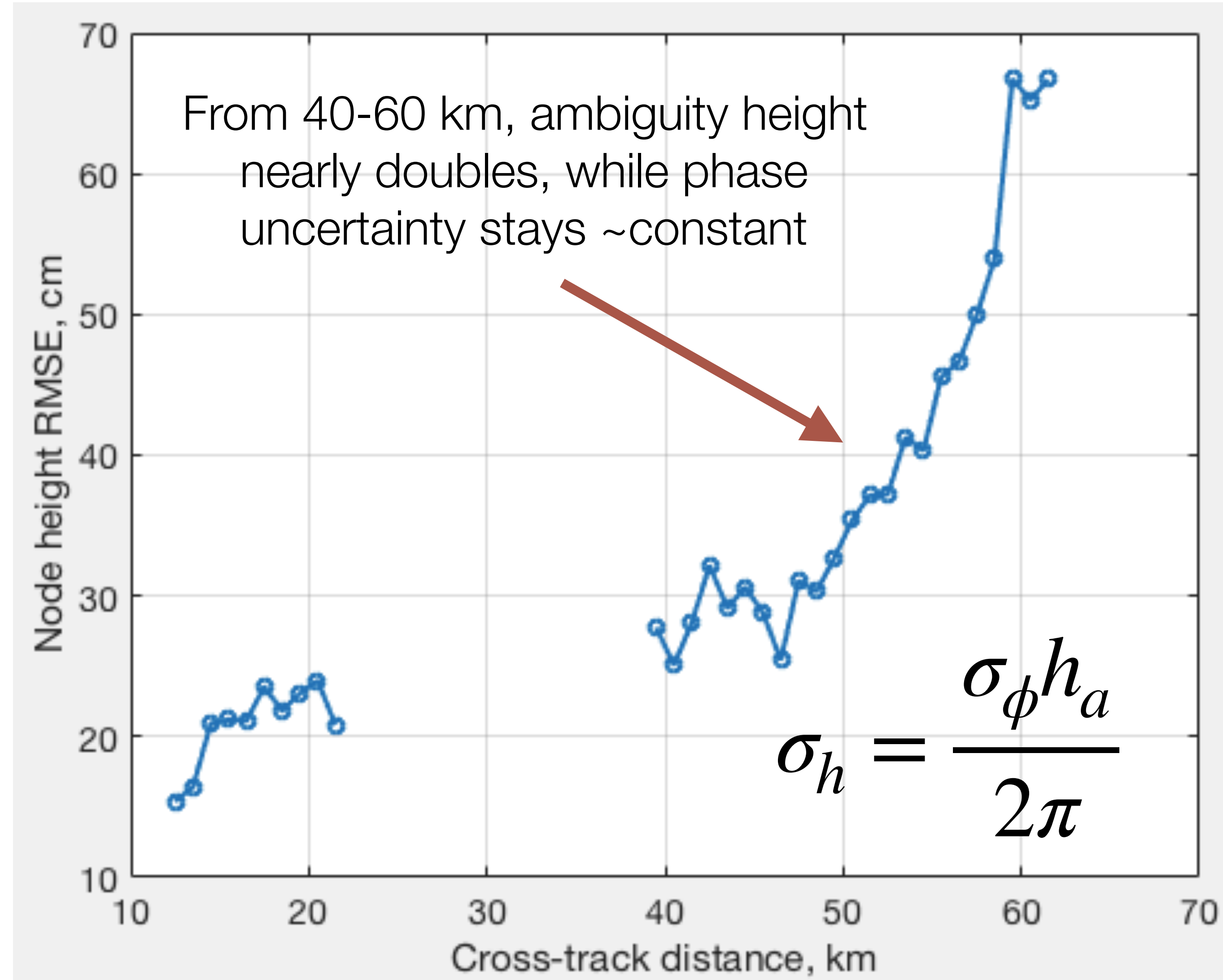
- Reach average height and slopes are computed using a first-order fit to the height data vs downstream flow distance
- RiverObs writes out data elements. These element definitions are being finalized.
- An unofficial beta test dataset will be announced once baseline data elements are final. This version is “pre-beta” and is also available.

Node Heights

Reach Slopes





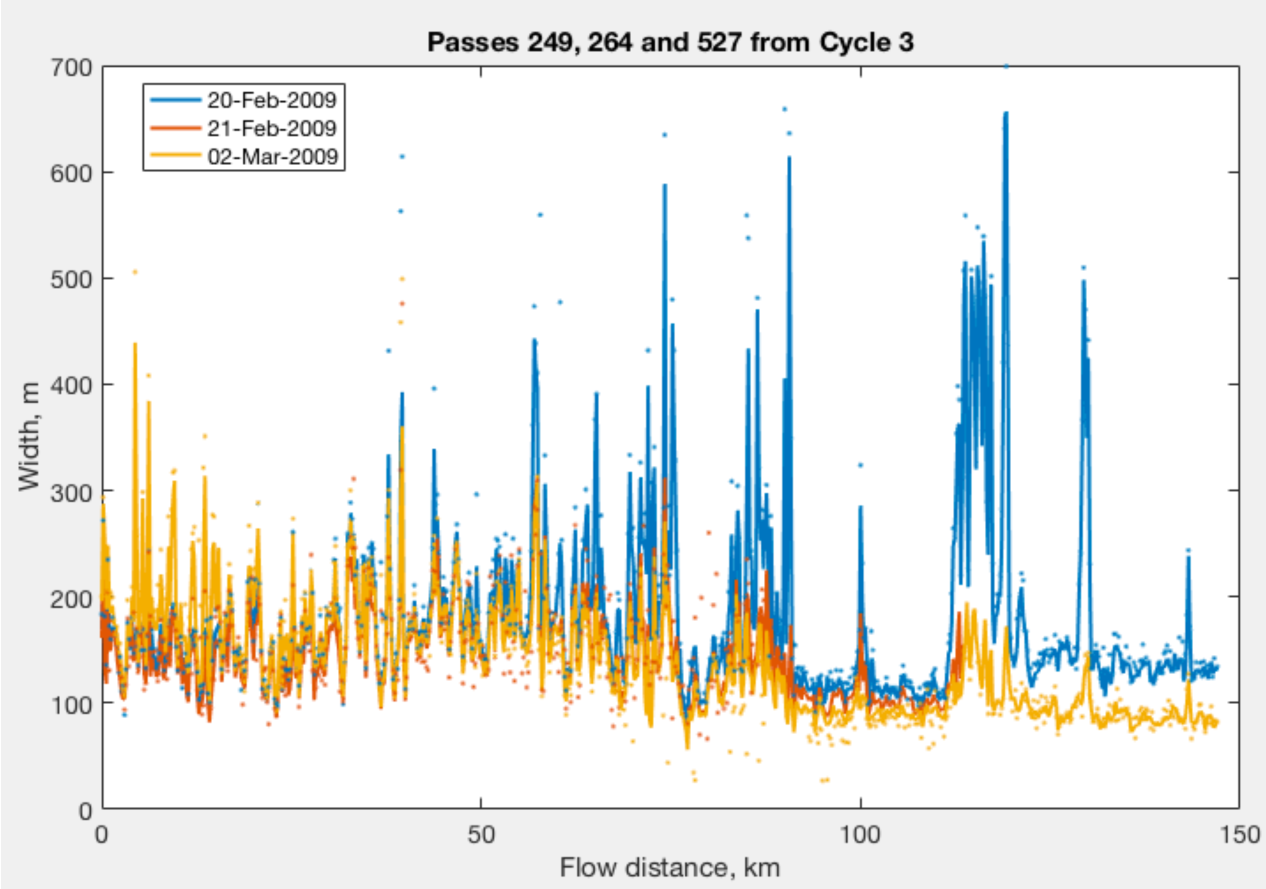


*modulated by river-track orientation

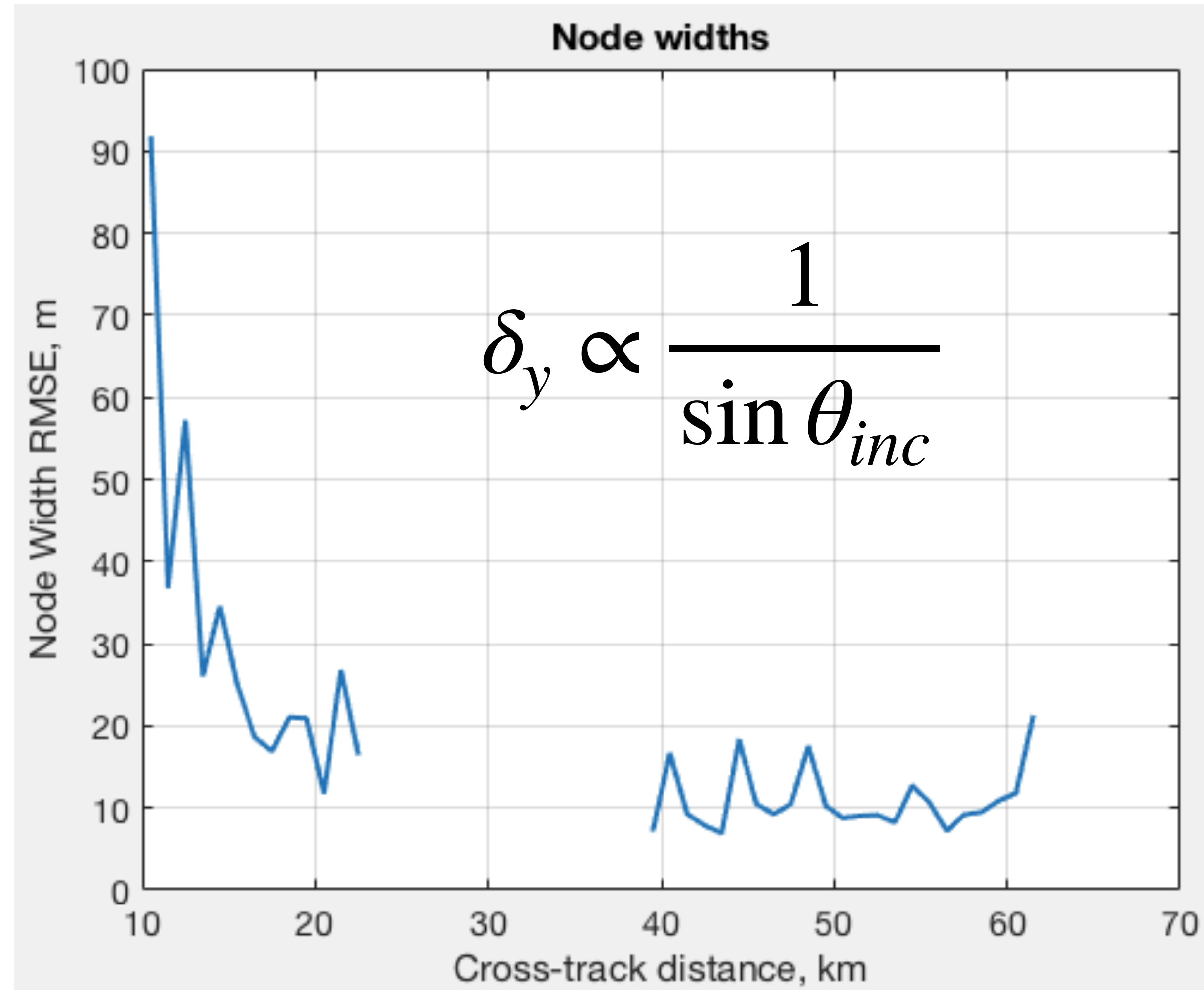
Across all nodes, height RMSE = 38 cm

Node height: Variability and error

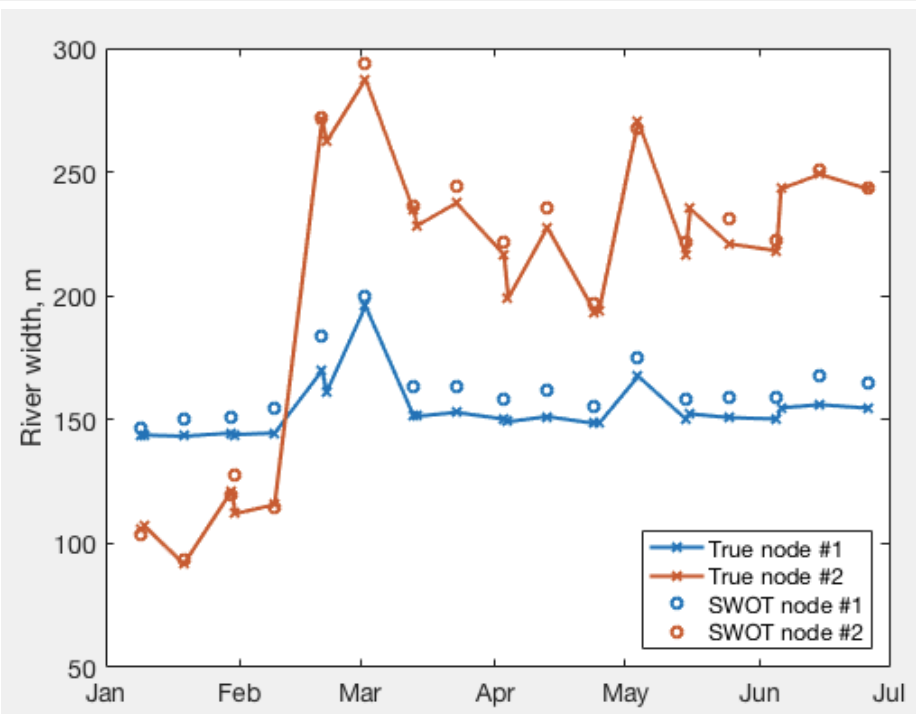
Errors governed by cross-track distance in this simulation, via the interplay between pixel size*, signal-to-noise, and ambiguity height



From 20 km to 10 km, incidence angle decreases from ~1.5° to ~.75°, ~doubling pixel size from 25 m to 55 m. SNR decreases as well.



$$\delta_y \propto \frac{1}{\sin \theta_{inc}}$$



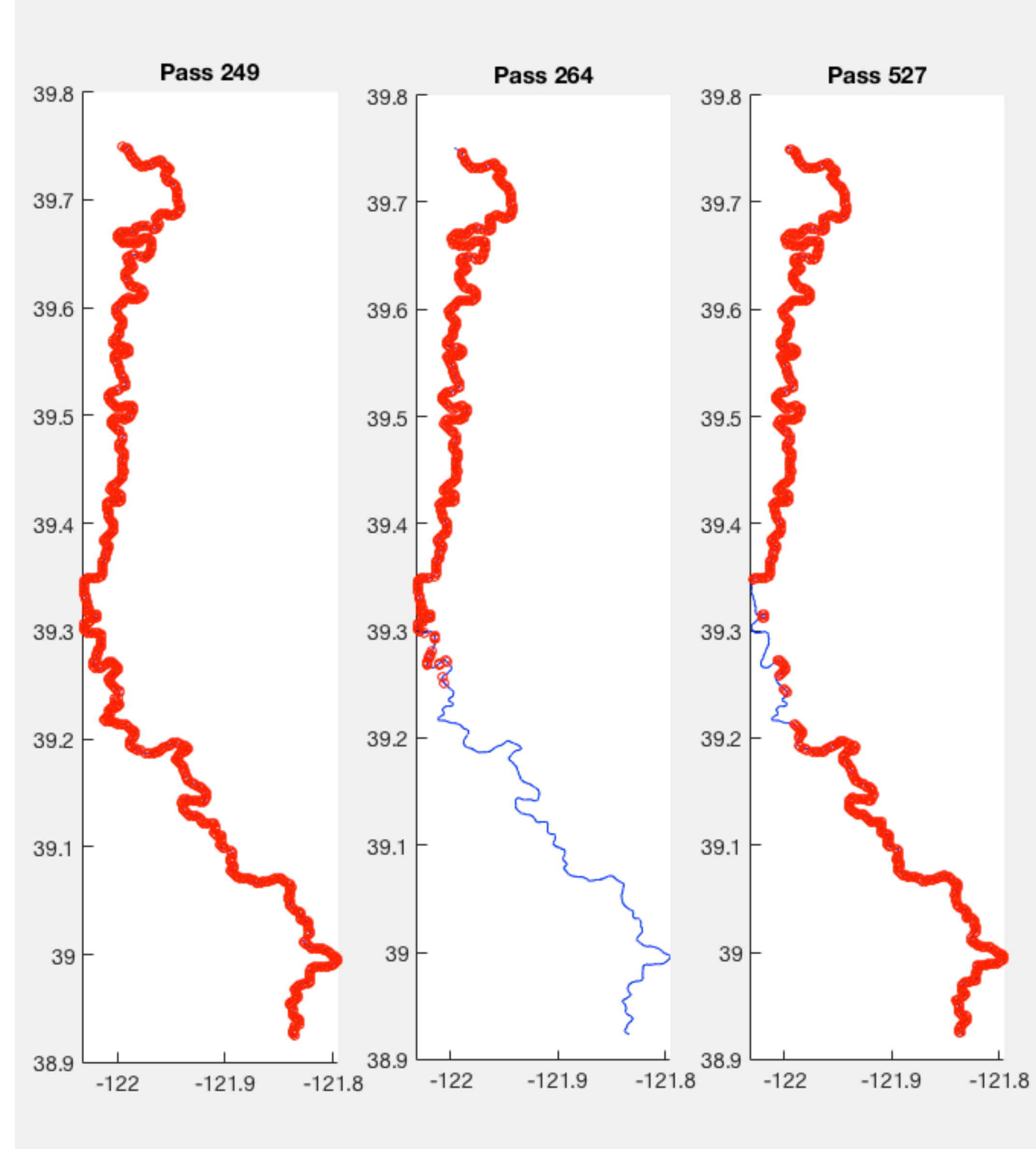
Across all nodes, width RMSE = 18 m

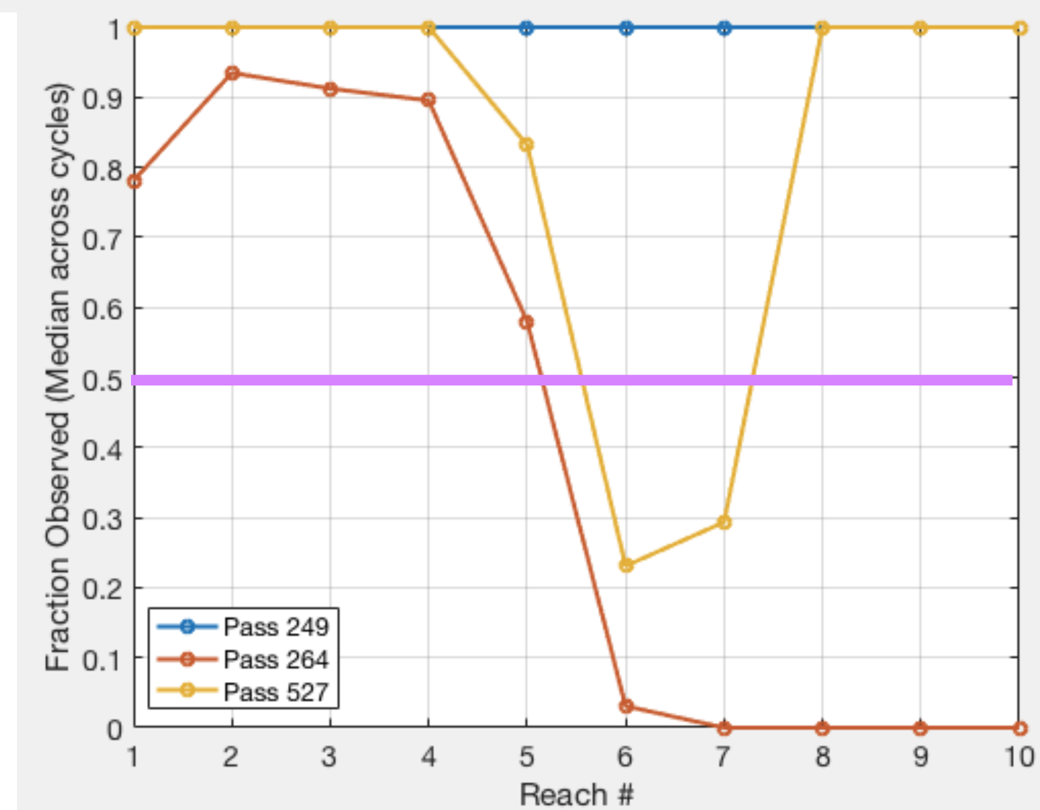
Node width: Variability and error

Errors governed by cross-track distance via the increase in cross-track pixel size, and drop-off in SNR

Spatial coverage for nodes and partial reach-observation

- We do not produce node elevations or widths if there are <100 pixels mapped to the node
- We do not produce reach-average data products if <50% of nodes are observed
- Pass 249: all reaches fully observed.
- Pass 264 even best reaches are far in the near swath (<20 km) where pixels are large. Reaches are always partially observed. Downstream not observed at all.
- Pass 527 has three reaches that are partially observed



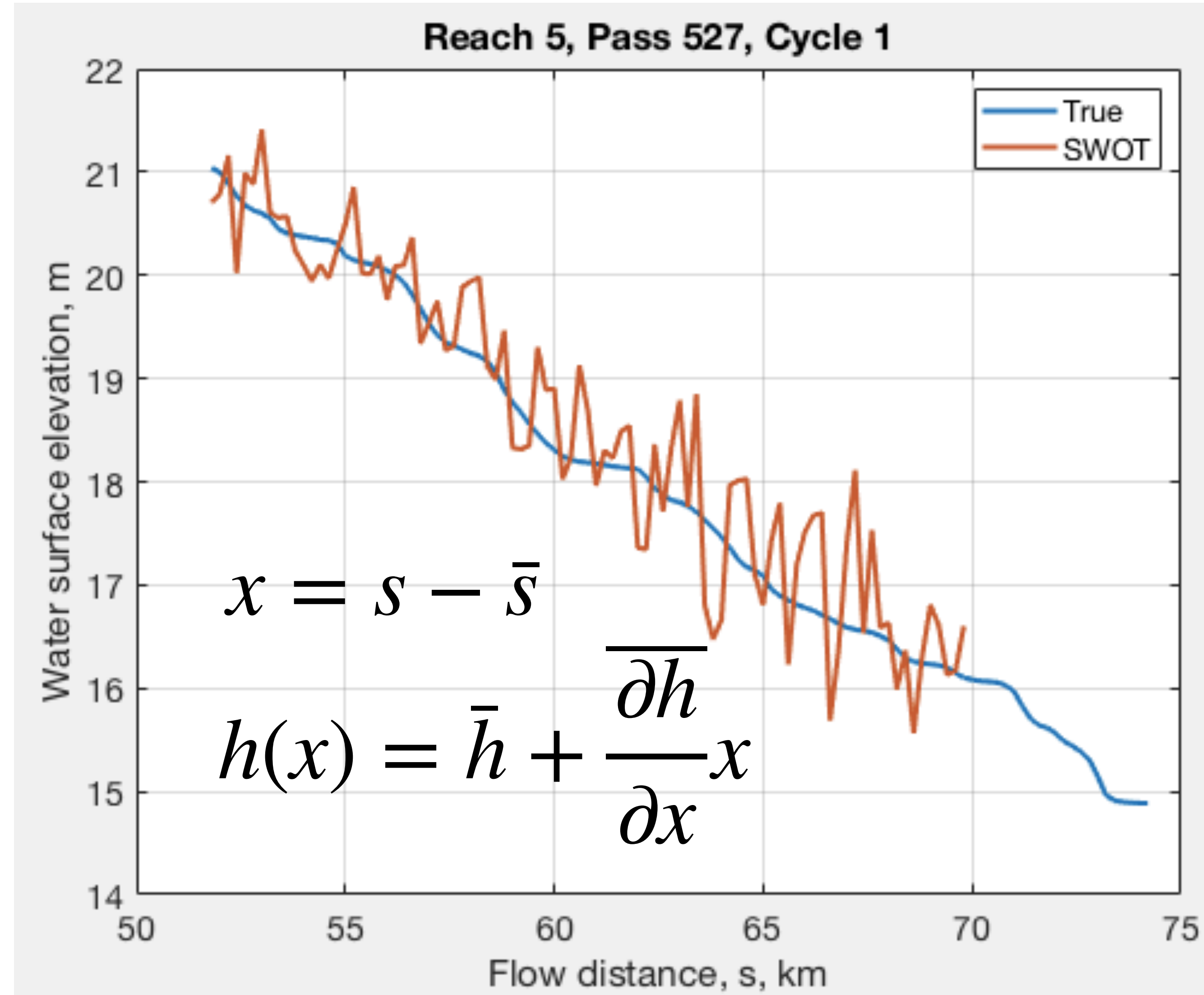


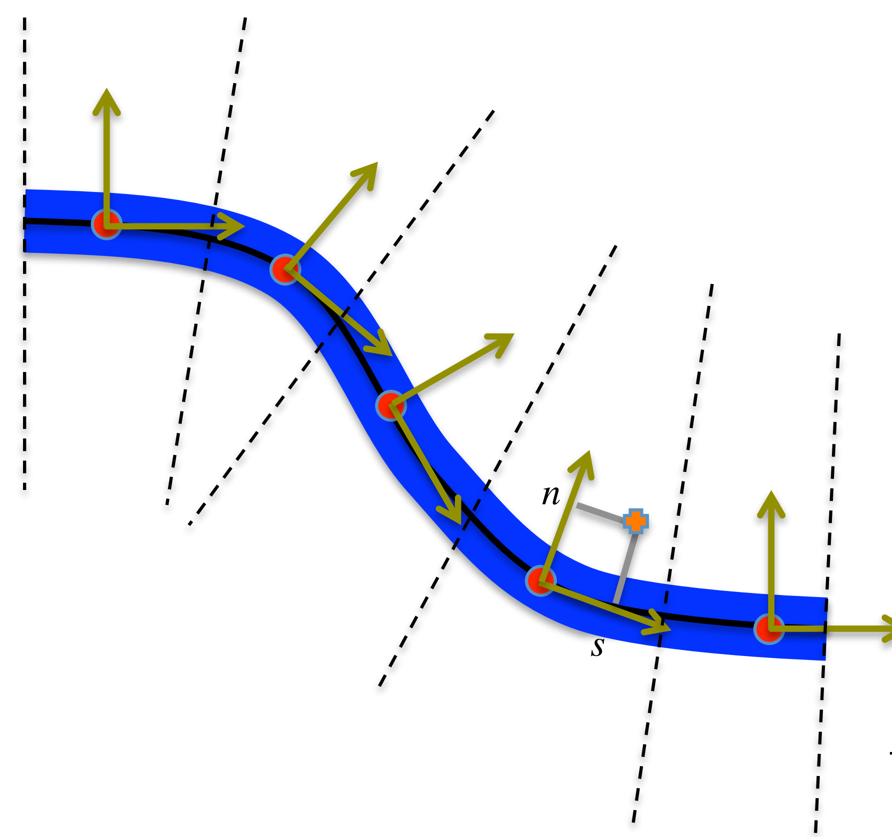
~83% of the nodes in this reach are observed for Pass 527

RiverObs computes average height and slope from a linear fit to the data

RiverObs bug alert! The way the software is coded, it is not robust to partial reach observation: led to bias of ~70 cm for this reach.

Rui produced a fix (June 21) that has not yet been fully incorporated on GitHub, though is incorporated in the pre-Beta data products we have shared.

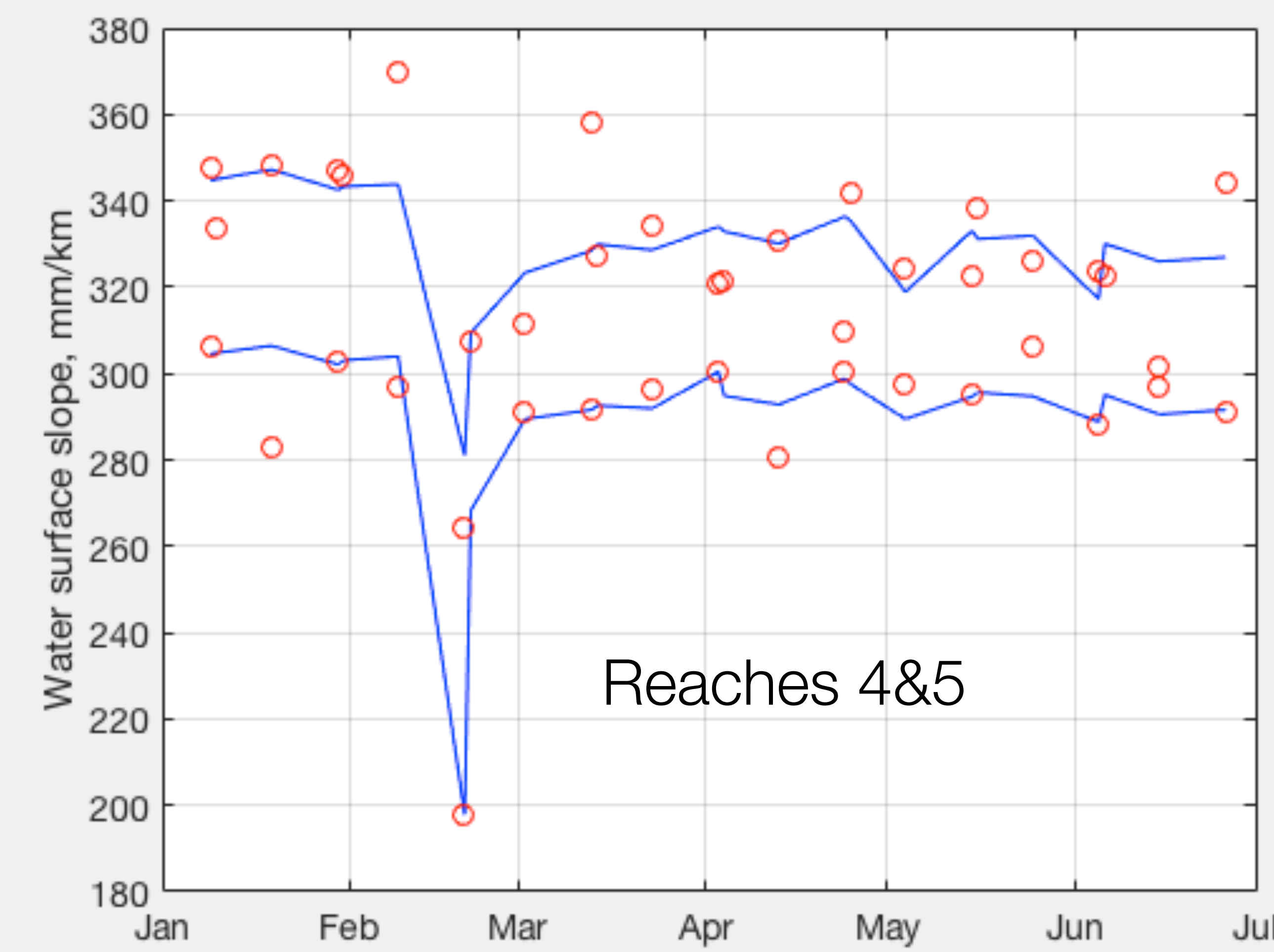


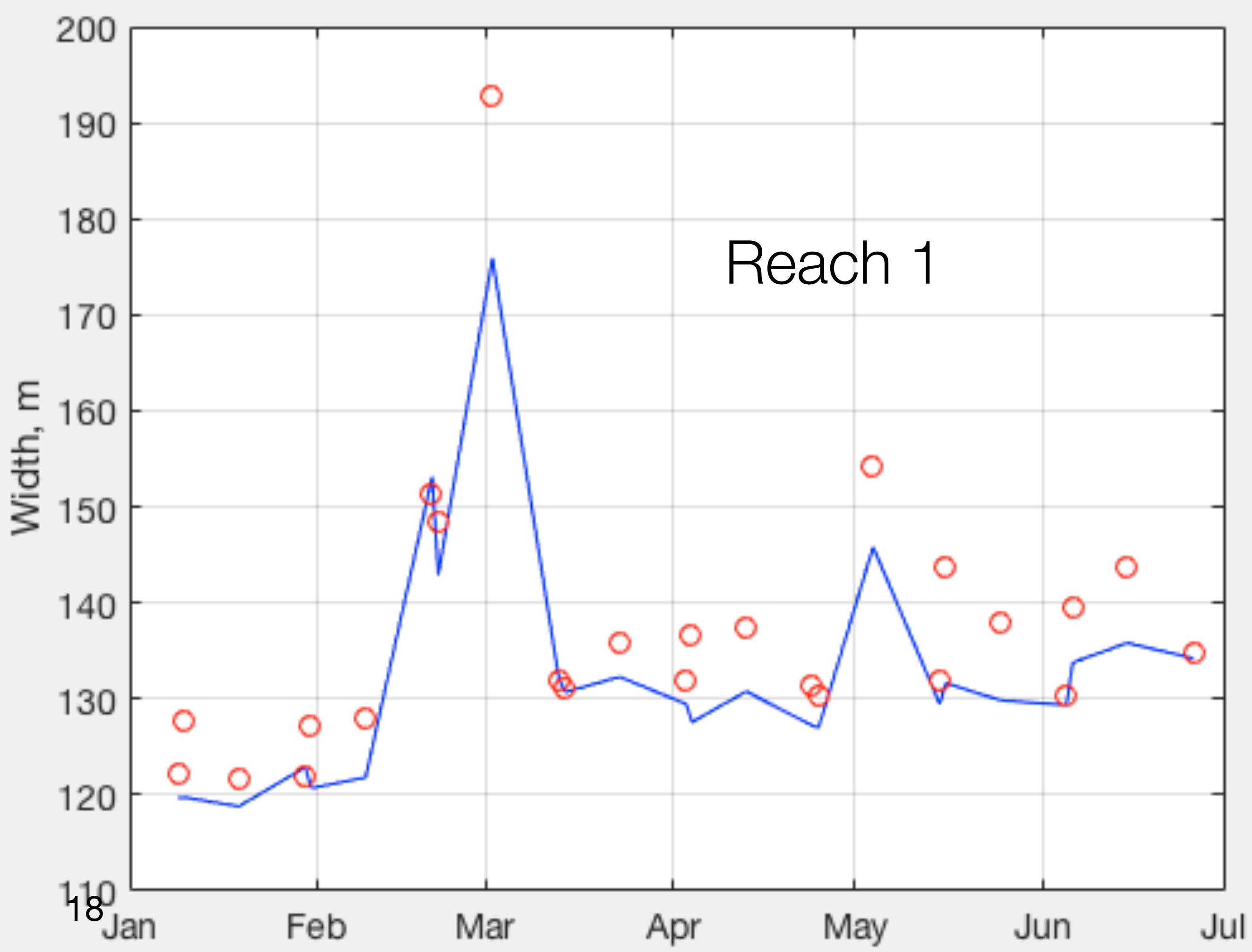


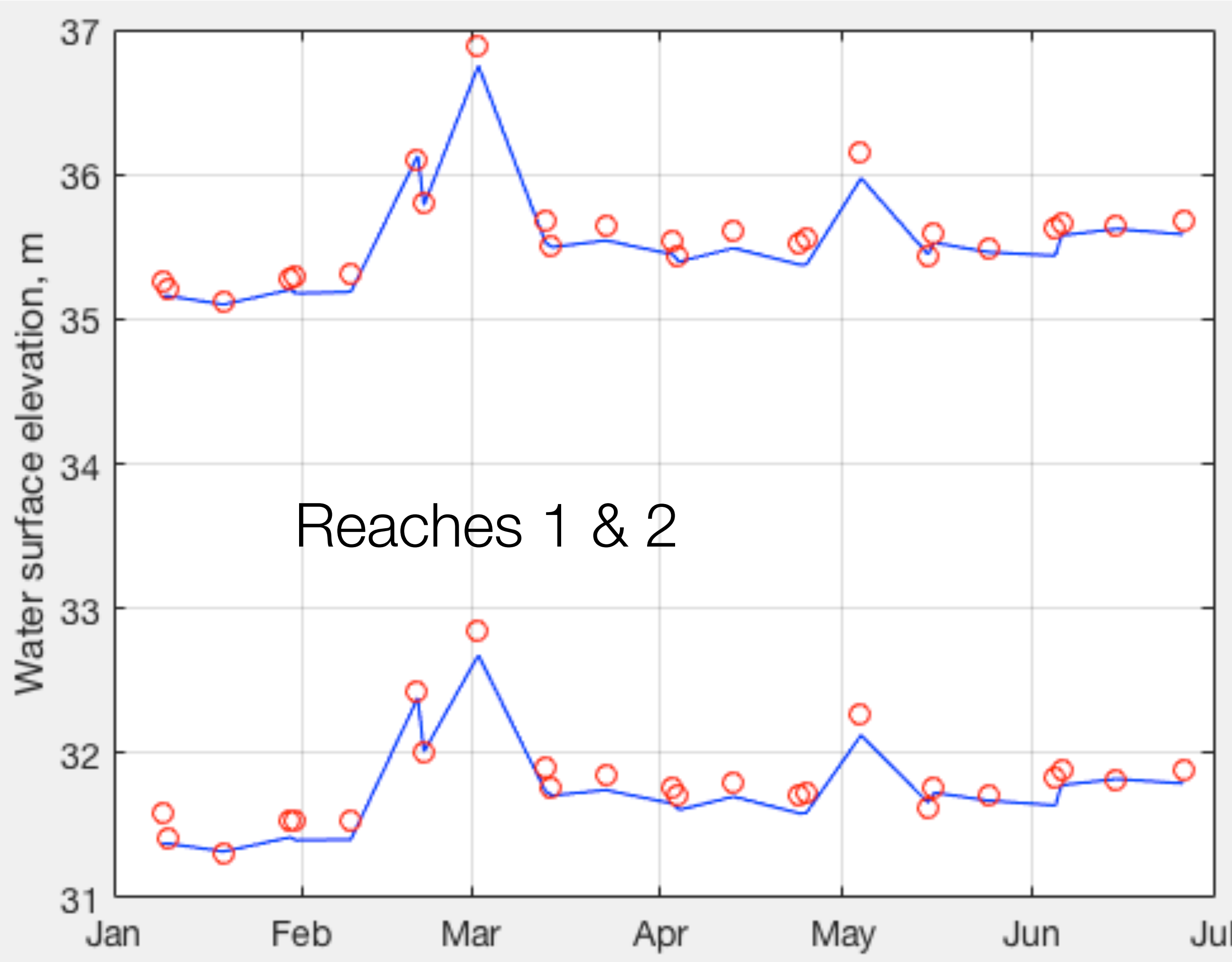
Partially-observed reaches

Reach data: Example timeseries

- Example reaches shown for height, width and slope
- The data resolve many of the smaller changes in the observables

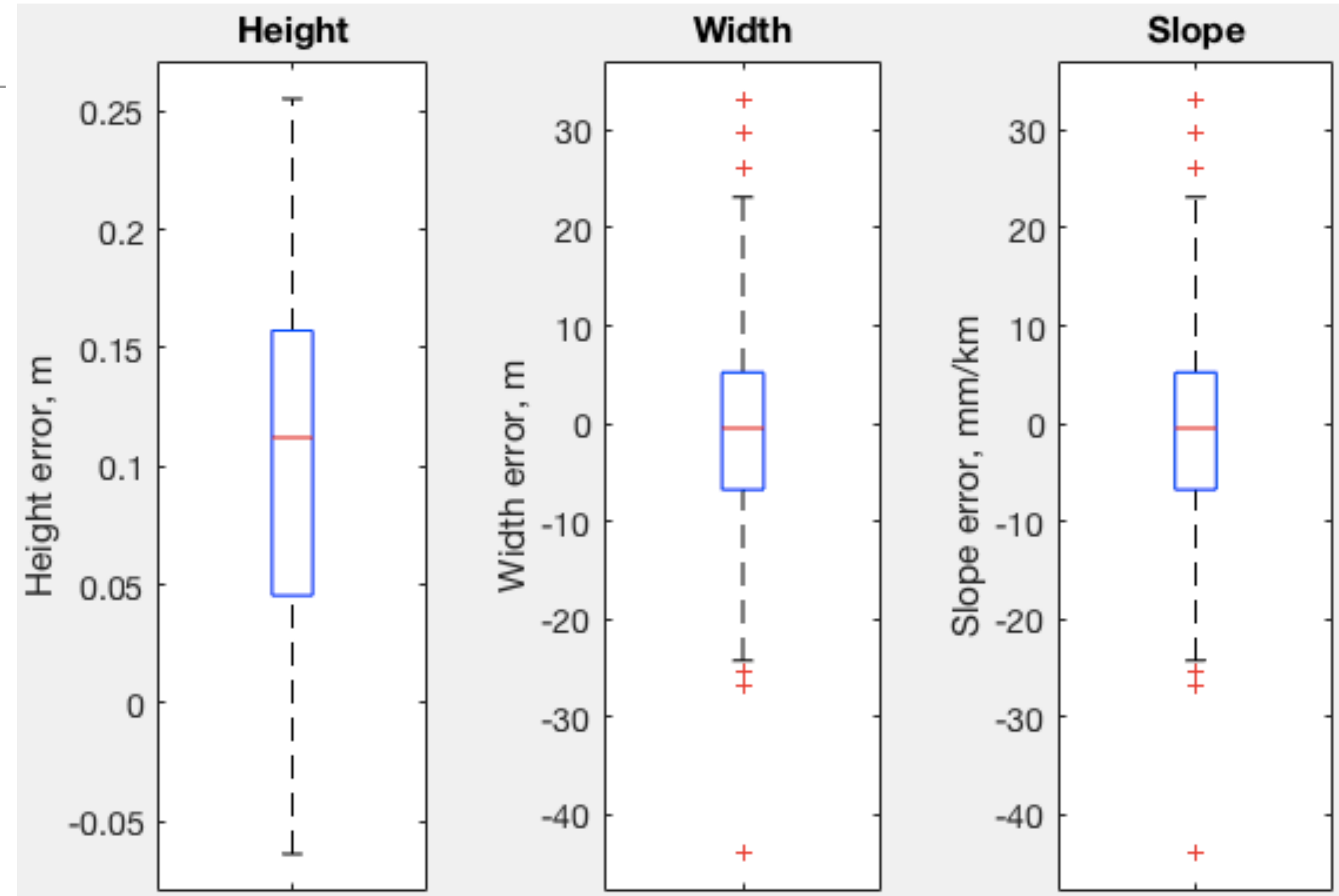






Reach data overview: summary errors

- Height errors: **12.5 cm RMSE**. Nearly all of this is bias, and most of the bias is due to bias in the (simulated) wet troposphere*
- Width errors: **4.3 m RMSE**. Much of this is a slight high bias. Caveat: errors in riparian vegetation and dark water not included
- Slope errors: **10.5 mm/km RMSE**. Most of this is random.



Note: Reach lengths averaged 14.5 km, and ranged from 8.4 to 26 km.

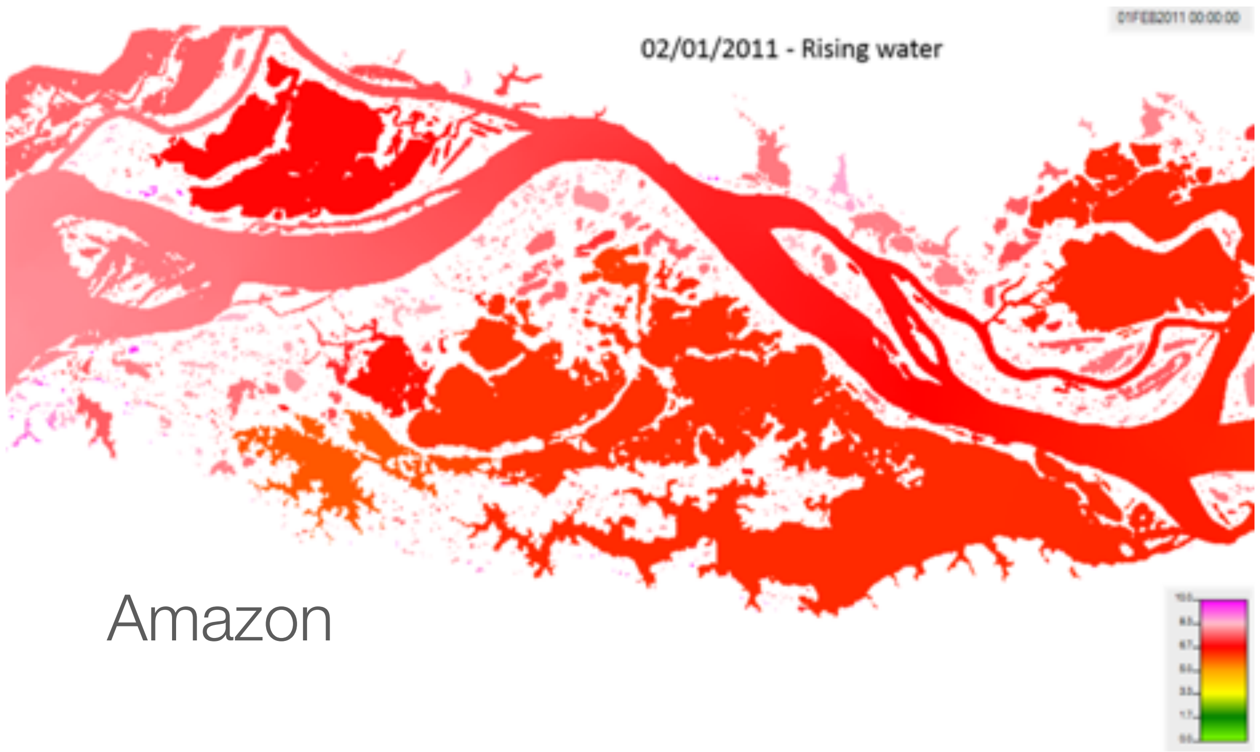
* The source of this bias has been identified. It will be fixed before final release.

Summary

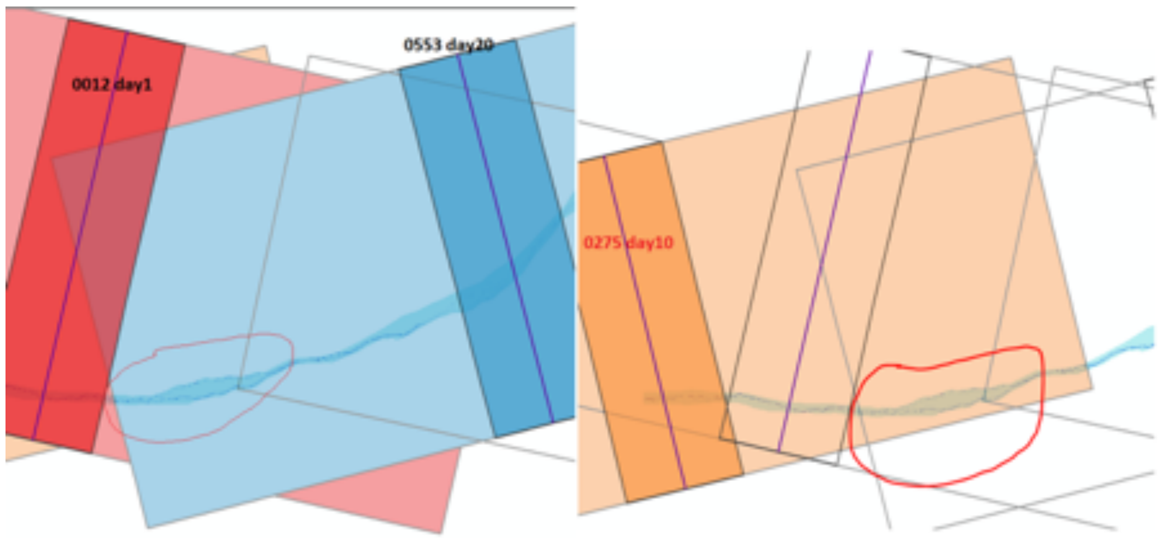
- Current: Beta data products useful for exploring what SWOT data will look like. Does not include dark water or riparian vegetation errors.
- Node width accuracy exceeded expectations! Caveat: not all errors are yet taken into account. Finally: May be able to improve them using height.
- Future: Beta will be ready for download in two months, by August 31. “Official” element names, and as many data elements as possible. Future versions will include cross-sectional area and discharge parameters.

Future possible unofficial example datasets

- The **St. Lawrence River** (courtesy Jean-Michel Fiset, Environment & Climate Change Canada)
- The **Tanana River** (courtesy Tamlin Pavelsky, Elizabeth Altenau)
- The **Garonne River** (courtesy Kevin Larnier, Sylvain Biancamaria)
- The **Platte River** (courtesy Brett Sanders, Kostas Andreadis). Multiple separable channels
- The **Amazon River** (courtesy Rodrigo Paiva)
- The **Po River** (courtesy Alessio Domenghetti). WRR, 2018.



Amazon



Platte

Simulations run for all of these rivers. Analysis and way to make pixC available in process.

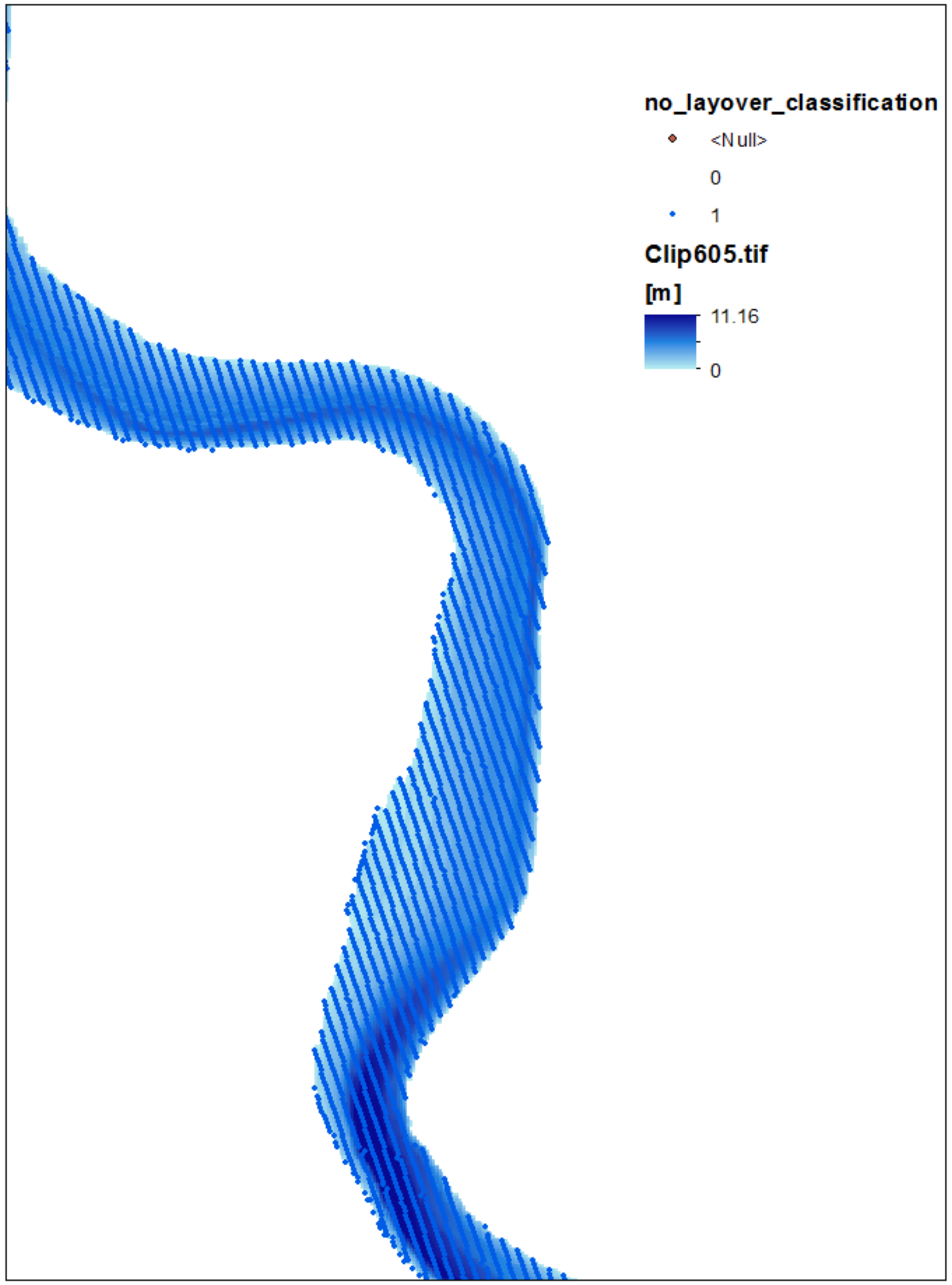
Snake River, near Jackson, Wyoming

Extra Slides

Snake River, near Jackson, Wyoming

Additional limitations

- The continuous classification algorithm sometimes produces negative widths for pixels. This *rarely* does lead to negative widths at nodes. Currently set to fill value
- We currently run RiverObs using an option to “trim” first and last nodes in domain. This is a pragmatic choice that needs to be addressed in future
- Unclear that enhanced slope data element being correctly computed for partially-observed reaches. Fix coming soon.

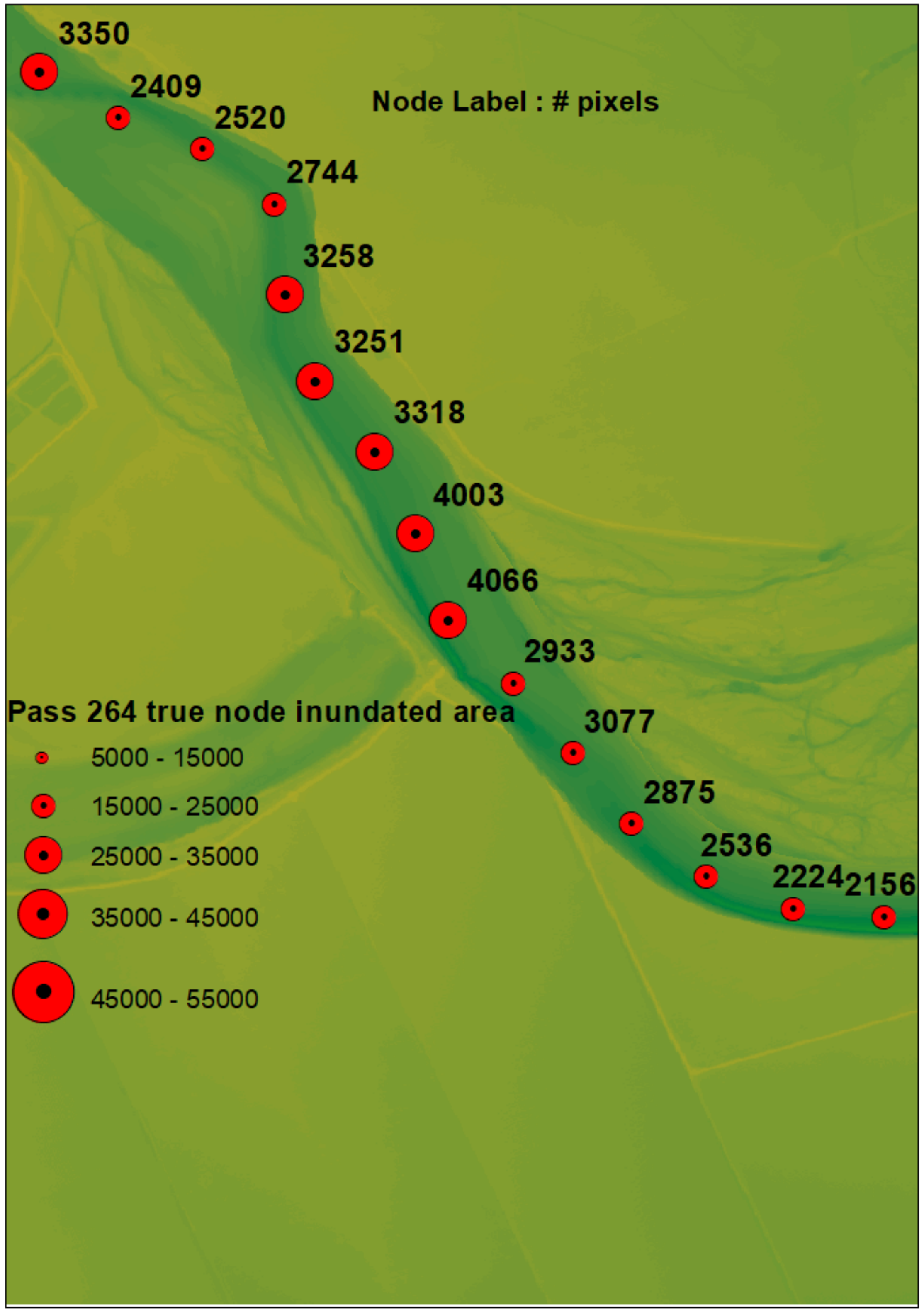


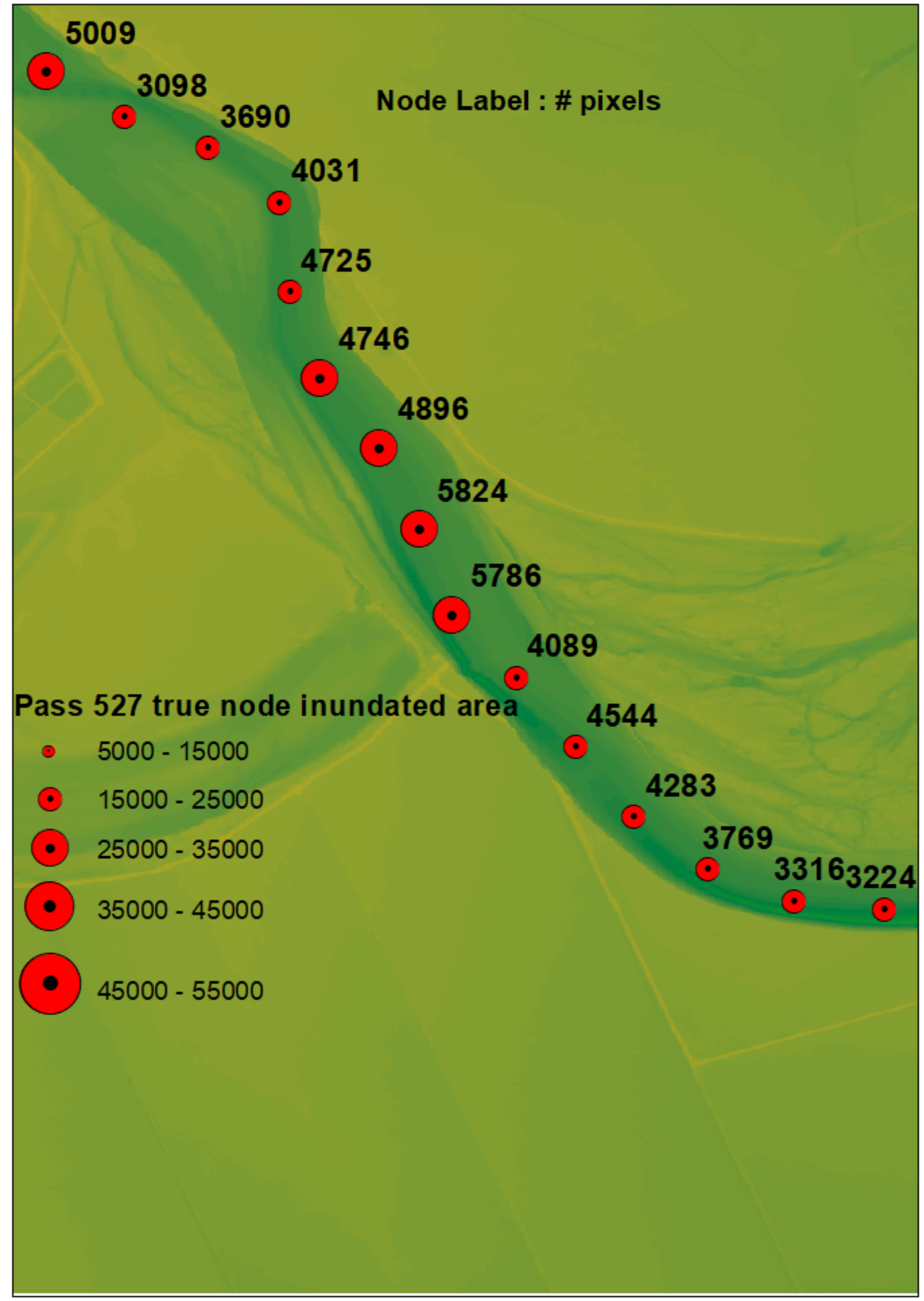
Node average width is node planform area divided by node spacing (200 m).

Node inundated area is computed by integration of the fractional water classification over all pixels assigned to a node in classes: water near land, land near water, and interior water

Computing true width

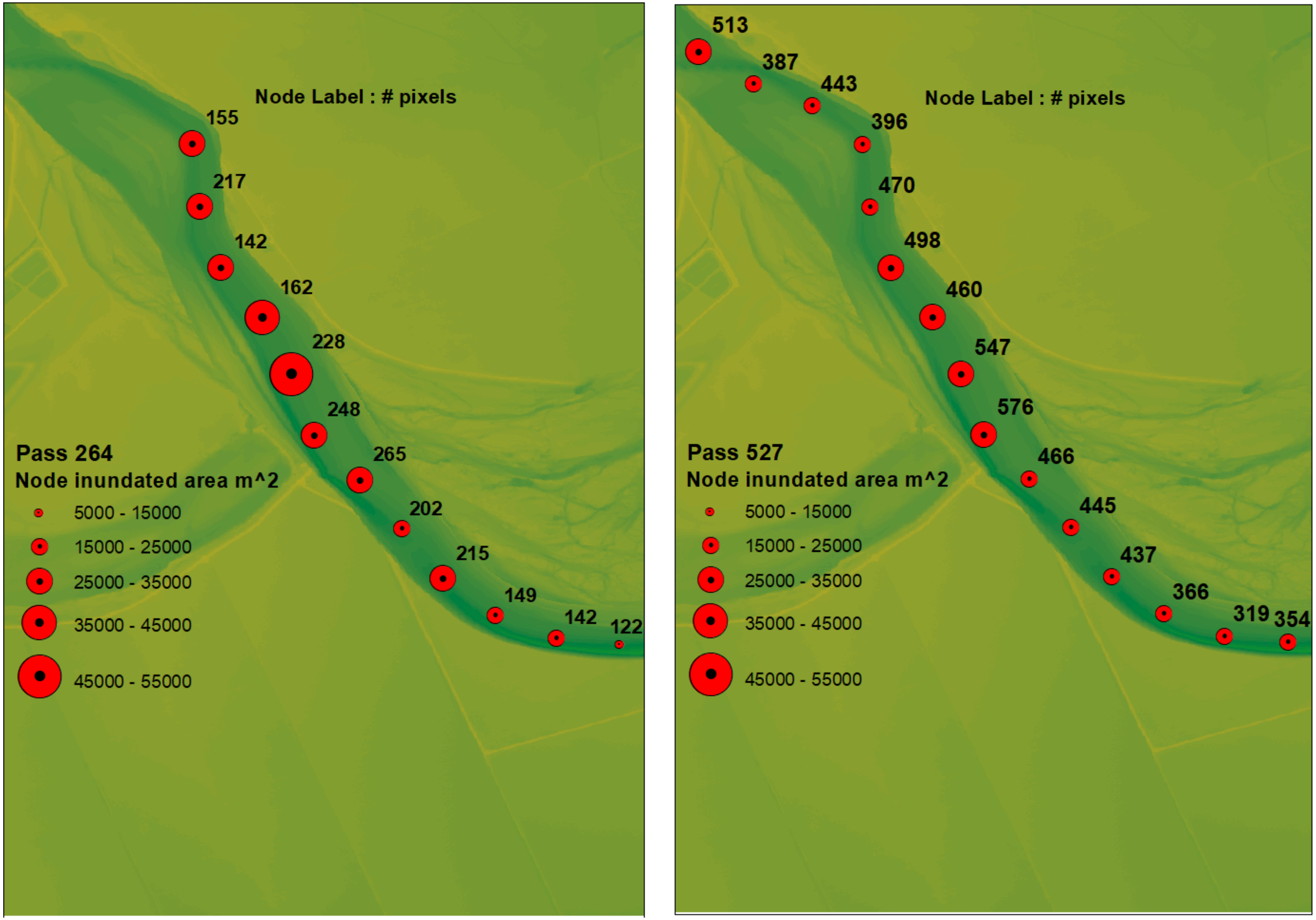
… from the “no layover” pixel cloud.





Node areas and number of pixels

These are from cycle 1; flow is nearly identical. Pass 264 vs 527 has true pixel areas 7.9 m² vs 5.28 m².



Node areas and number of pixels

These are from cycle 1; flow is nearly identical.