National Aeronautics and Space Administration

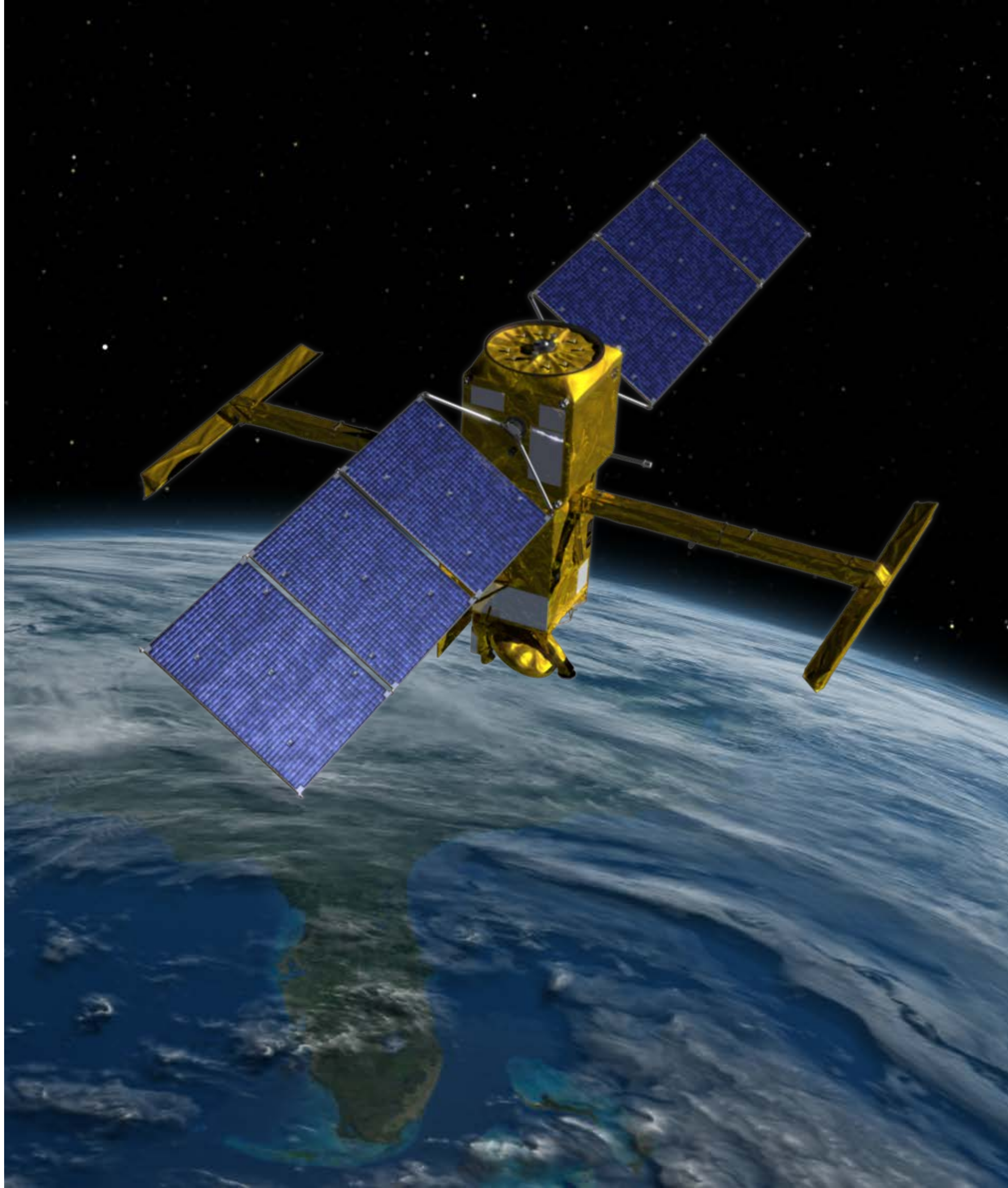
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

Surface Water and Ocean Topography Mission (SWOT) Mission

L2_HR River Processing

Science Team Meeting

Cassie Stuurman, 2021-09-15
with contributions from many others
(presented by Curtis Chen)



Outline

SWOT L2_HR River Processing

1. River processors review
2. Review river algorithms
   - i. Pixel assignment
   - ii. Aggregation to nodes
   - iii. Aggregation to reaches
3. Considerations for science users
   - i. Expected sources of error
4. Summary & questions

River Processors Review

There are three river processors:

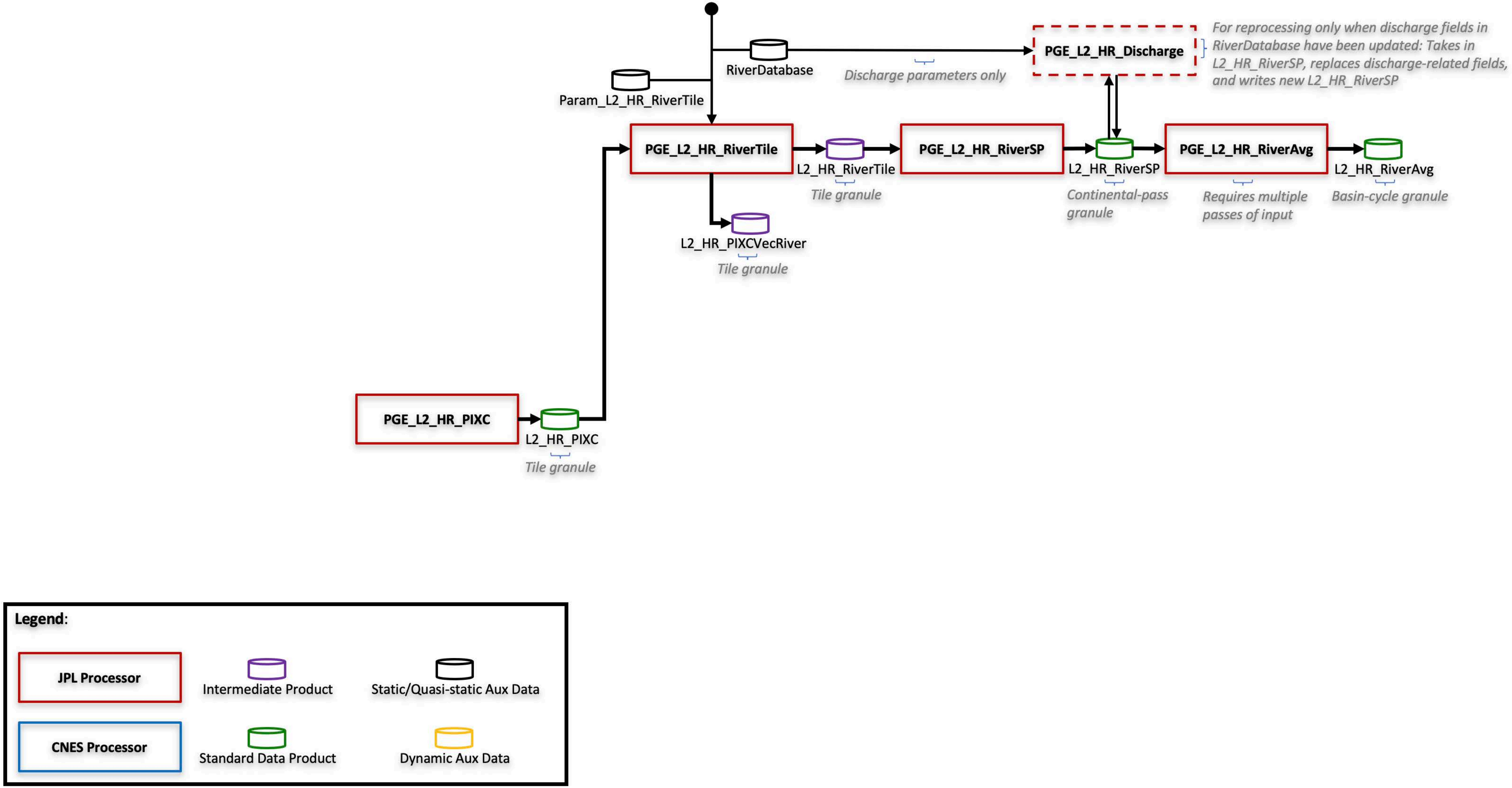
1. **RiverTile:**
   Input pixel cloud → River heights, widths, areas, & slopes

2. **RiverSP:**
   Input RiverTiles → Continent-sized RiverSP granule

3. **RiverAverage:**
   Input RiverSP granules from multiple orbit cycle passes
   → Averaged heights, slopes, and widths

KaRIn HR Flow

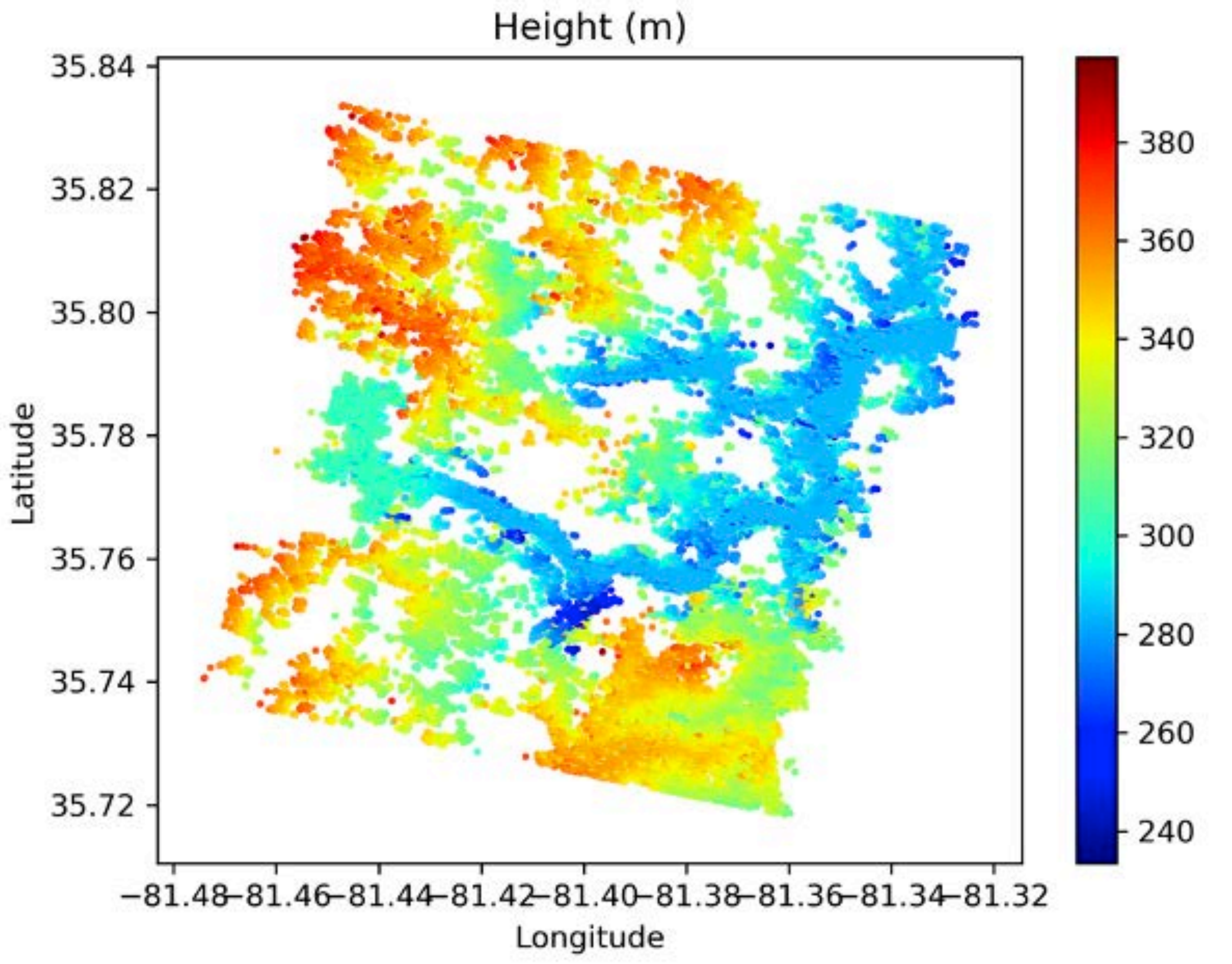
KaRIn HR Flow

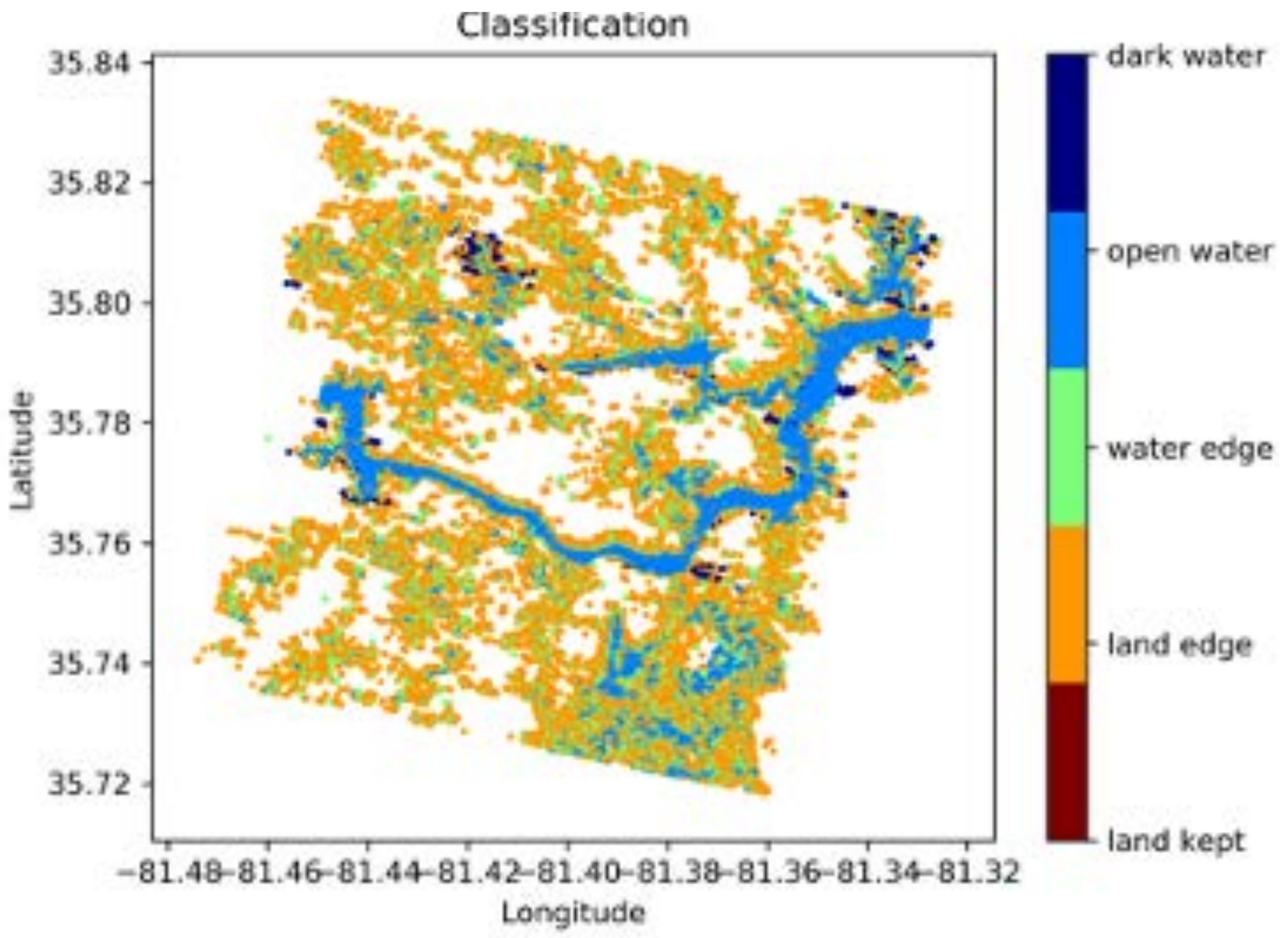


River Processor Inputs

Pixel Cloud Product

- InSAR-derived 3D pixel product
  - Latitude, longitude, height (*wrt ref ellipsoid*)
    - Also includes geoid & tide corrections that are removed during river processing
- Pixel type classes
  - Land, water body, water edge, etc.
- Considerations for science users:
  - Dark water
  - Measurement geometry/geolocation
    - associated sources of error

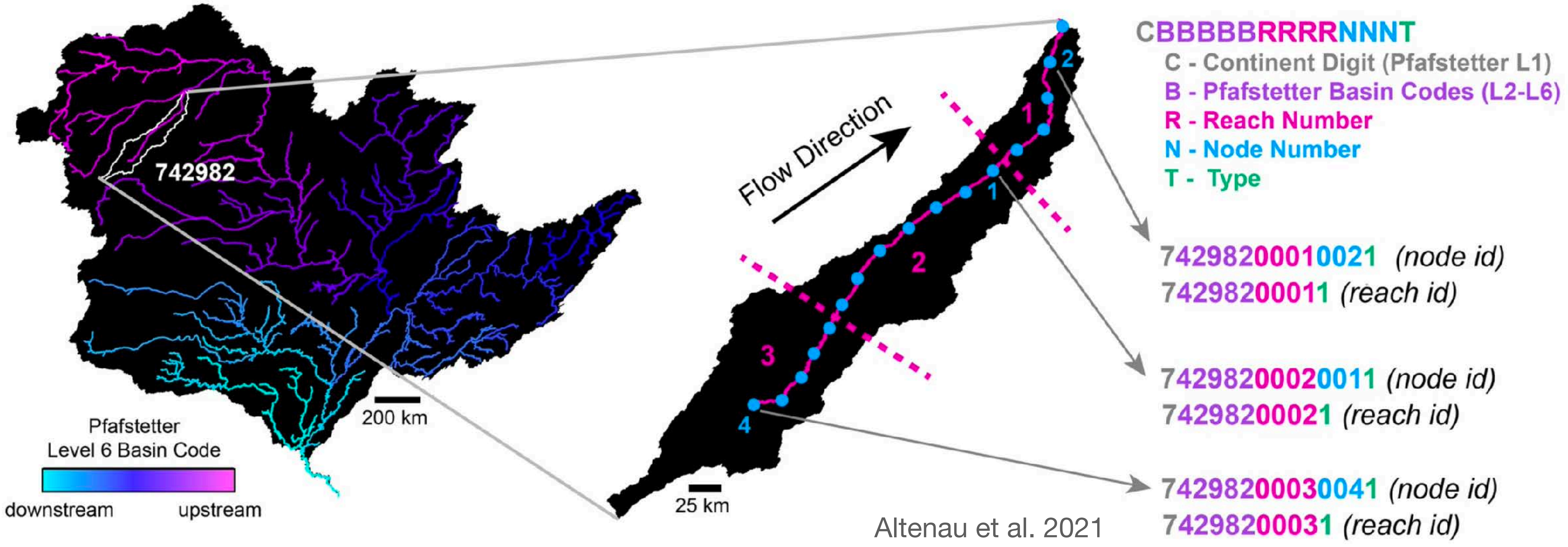




River Processor Inputs

SWORD

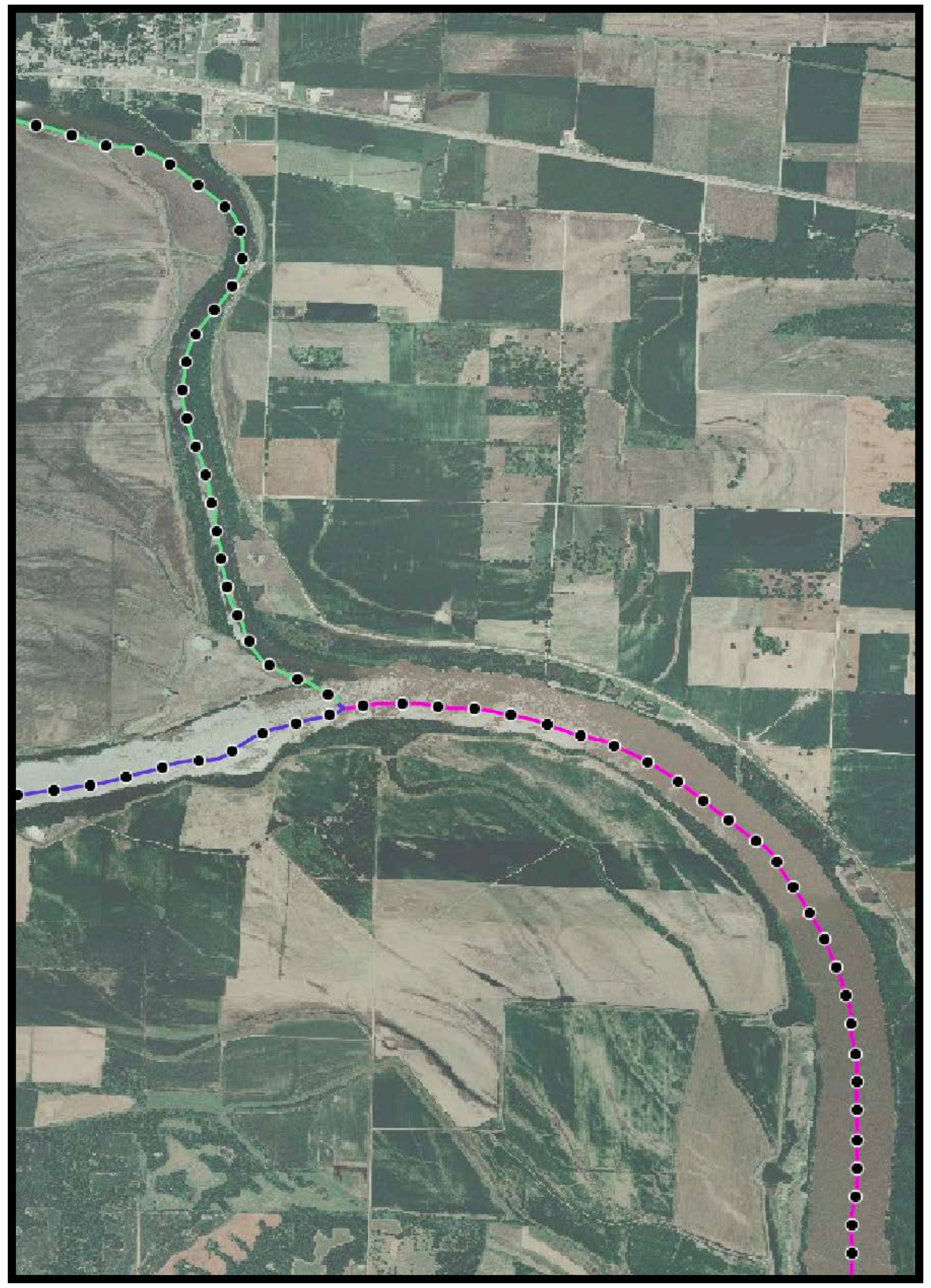
- SWOT River Database (SWORD)
  - A global prior river database (PRD) containing reach/node information, discharge parameters, dam/waterfall identifiers, multichannel information, and other prior science info for river algorithms as well as science users



Altenau et al. 2021

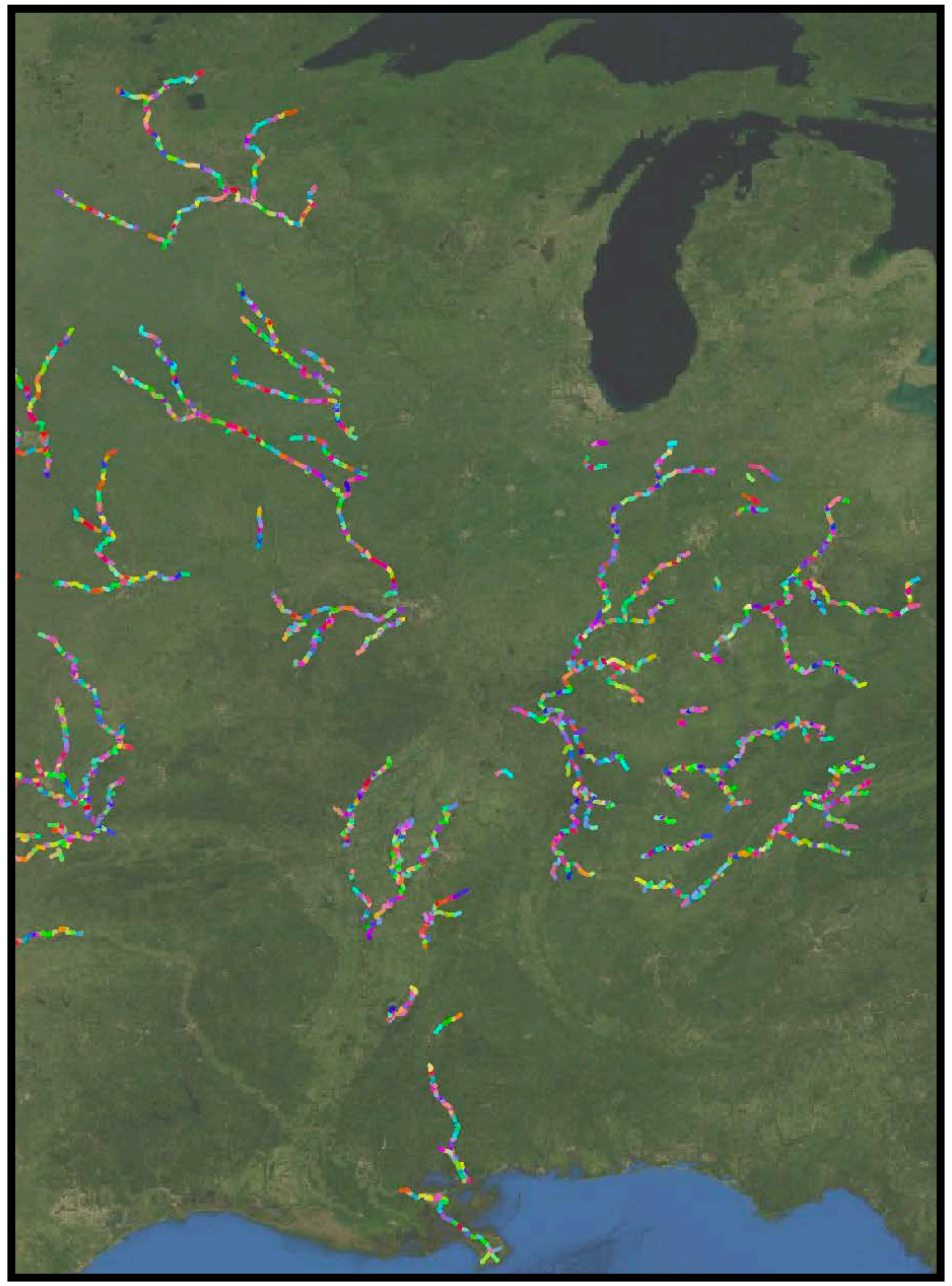
River Products

RiverSP product



L2_HR_RiverSP: reaches.shp; nodes.shp

RiverAvg product



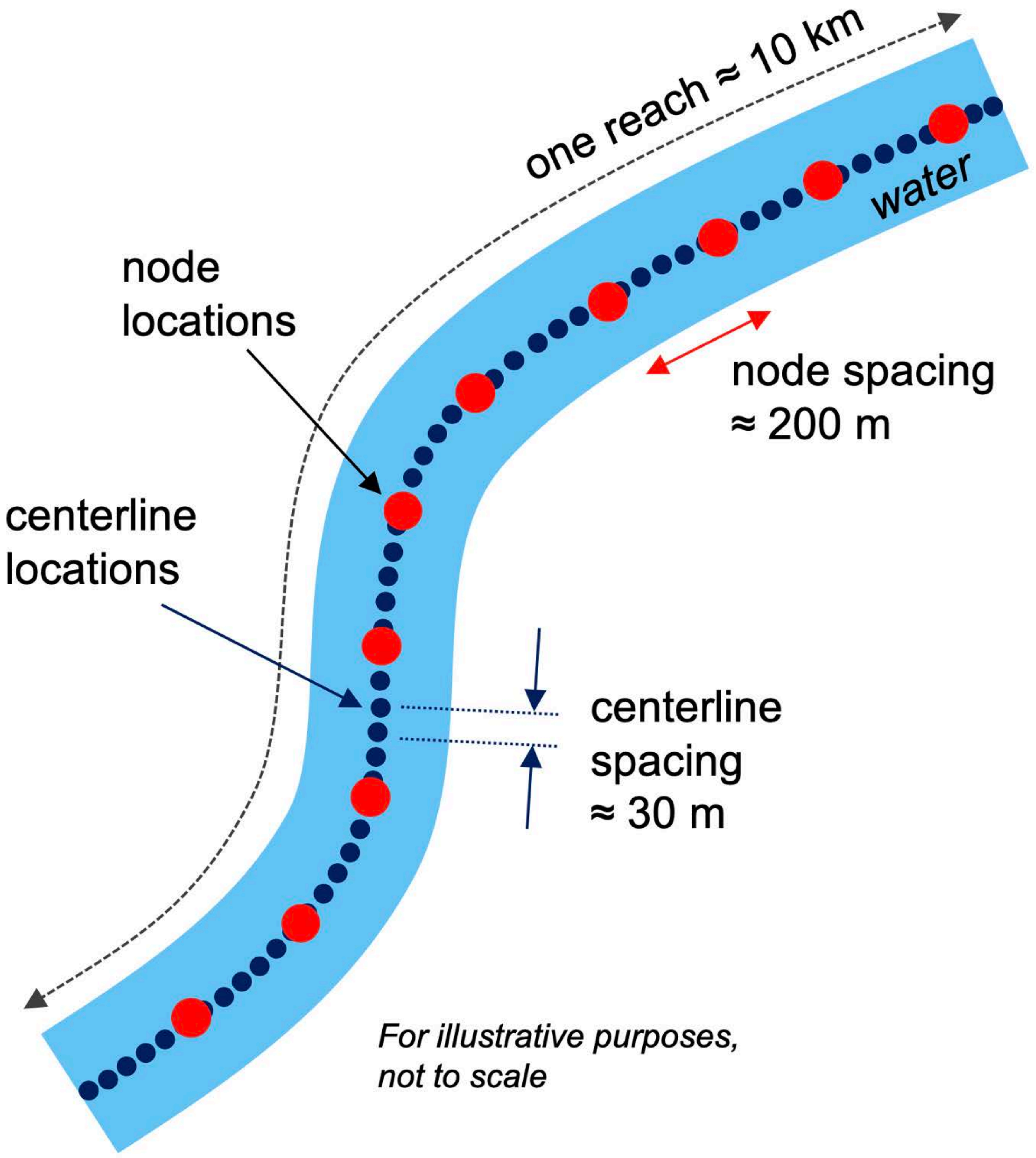
SWOT_L2_RiverAvg_007_74

Shown on LANDSAT basemap

River Algorithms

Fundamental concepts

- 3-D pixel clouds mapped to input river nodes from SWORD
- All water classes in tile are potential river pixels
- Various thresholds are used to distinguish river pixels from other water bodies in scene



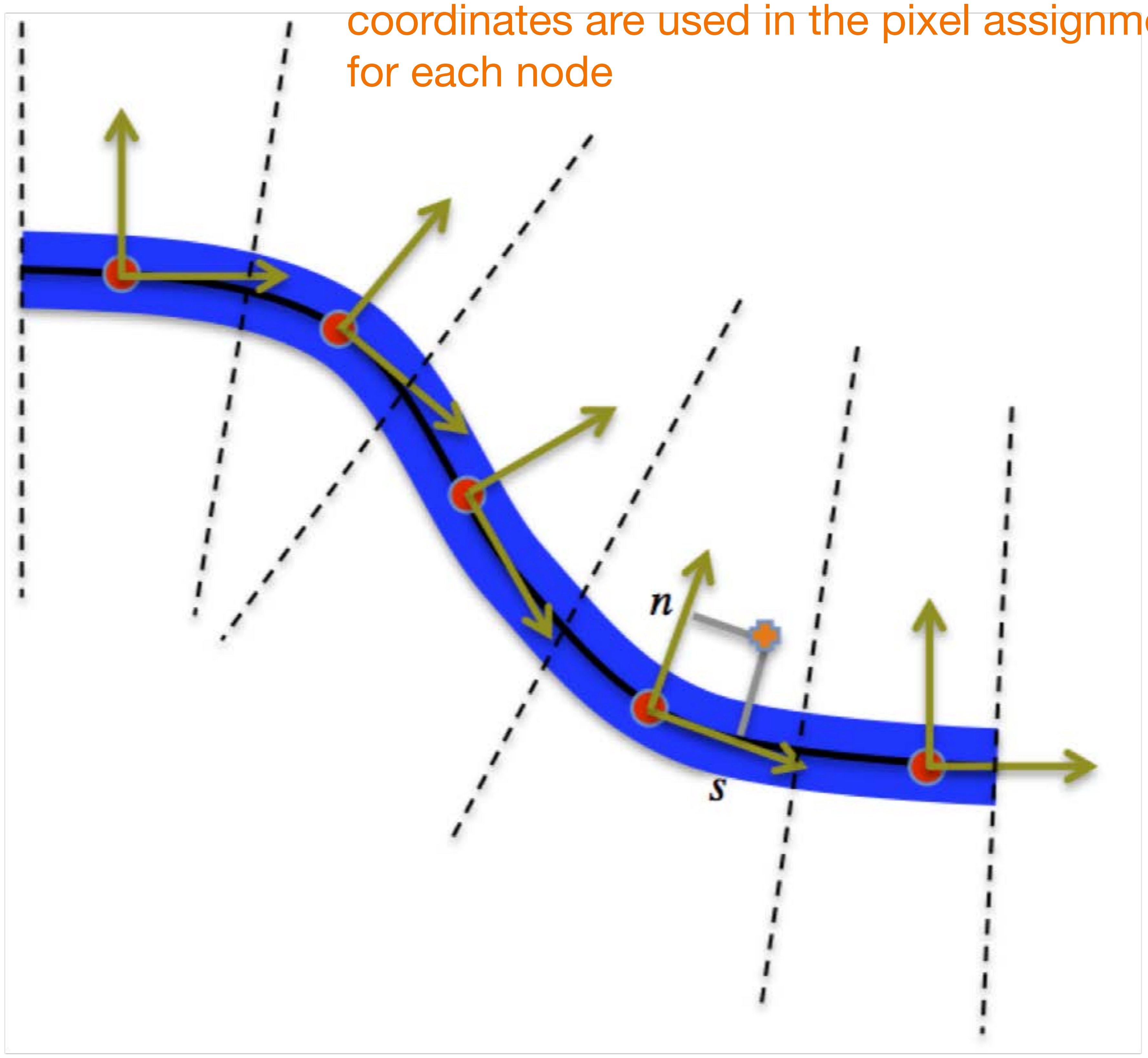
For illustrative purposes, not to scale

River Algorithms

Fundamental concepts

- 3-D pixel clouds mapped to input river nodes from SWORD
- All water classes in tile are potential river pixels
- Various thresholds are used to distinguish river pixels from other water bodies in scene
  - Defined in *s* (along-reach) and *n* (tangent)

Along-reach (*s*) and tangent-to-centerline (*n*) coordinates are used in the pixel assignment for each node



River Algorithms

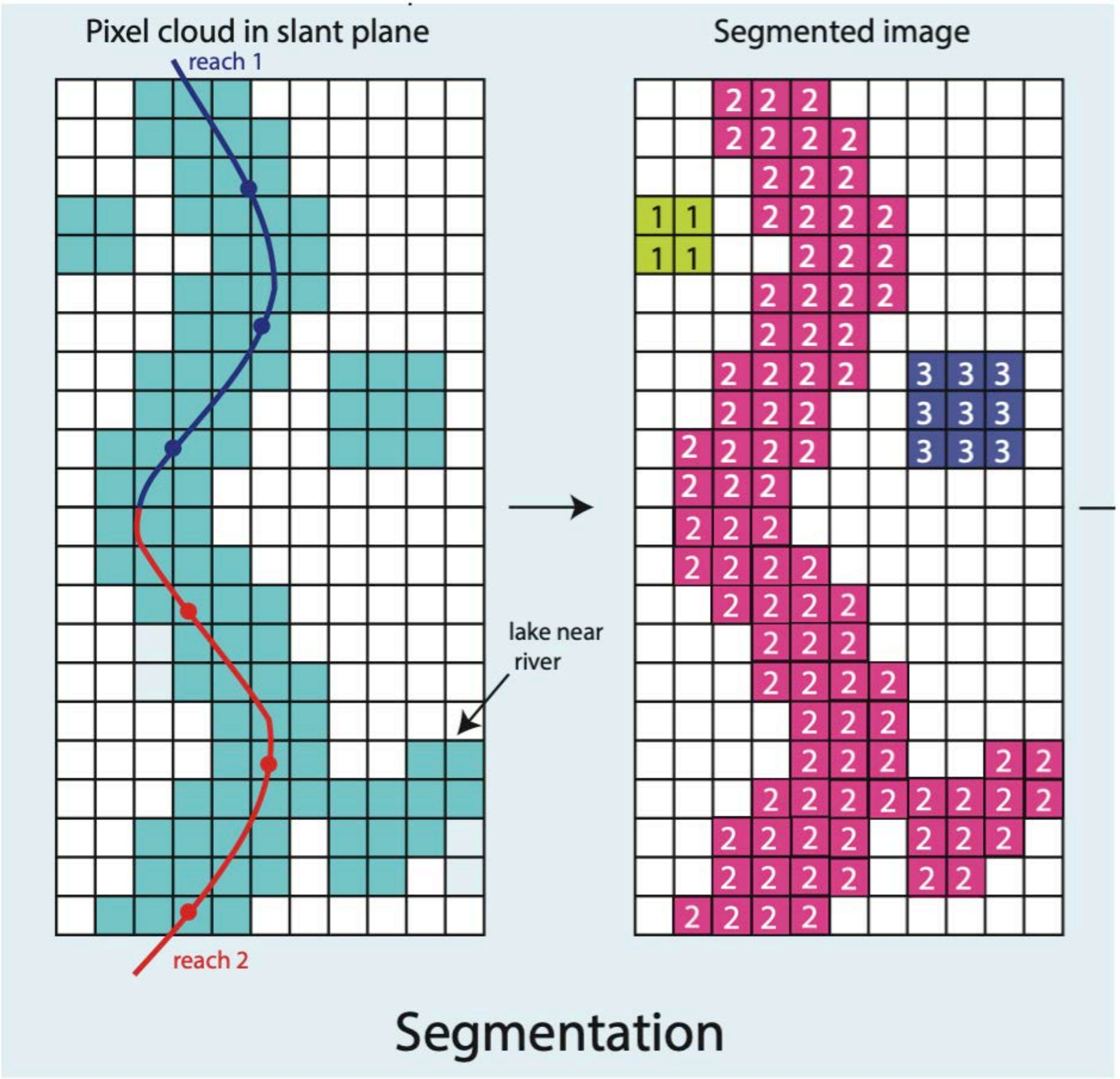
Fundamental Concepts

At a high level, river processing:

1. **Segments** input pixel cloud by contiguous water classes
2. Identifies **labels** associated with the channel centerline
3. **Maps** all pixels within channel label to nearest SWORD node
4. **Aggregates pixel information** to node information
5. **Aggregates node information** to reach information
6. **Outputs** products

Segmentation

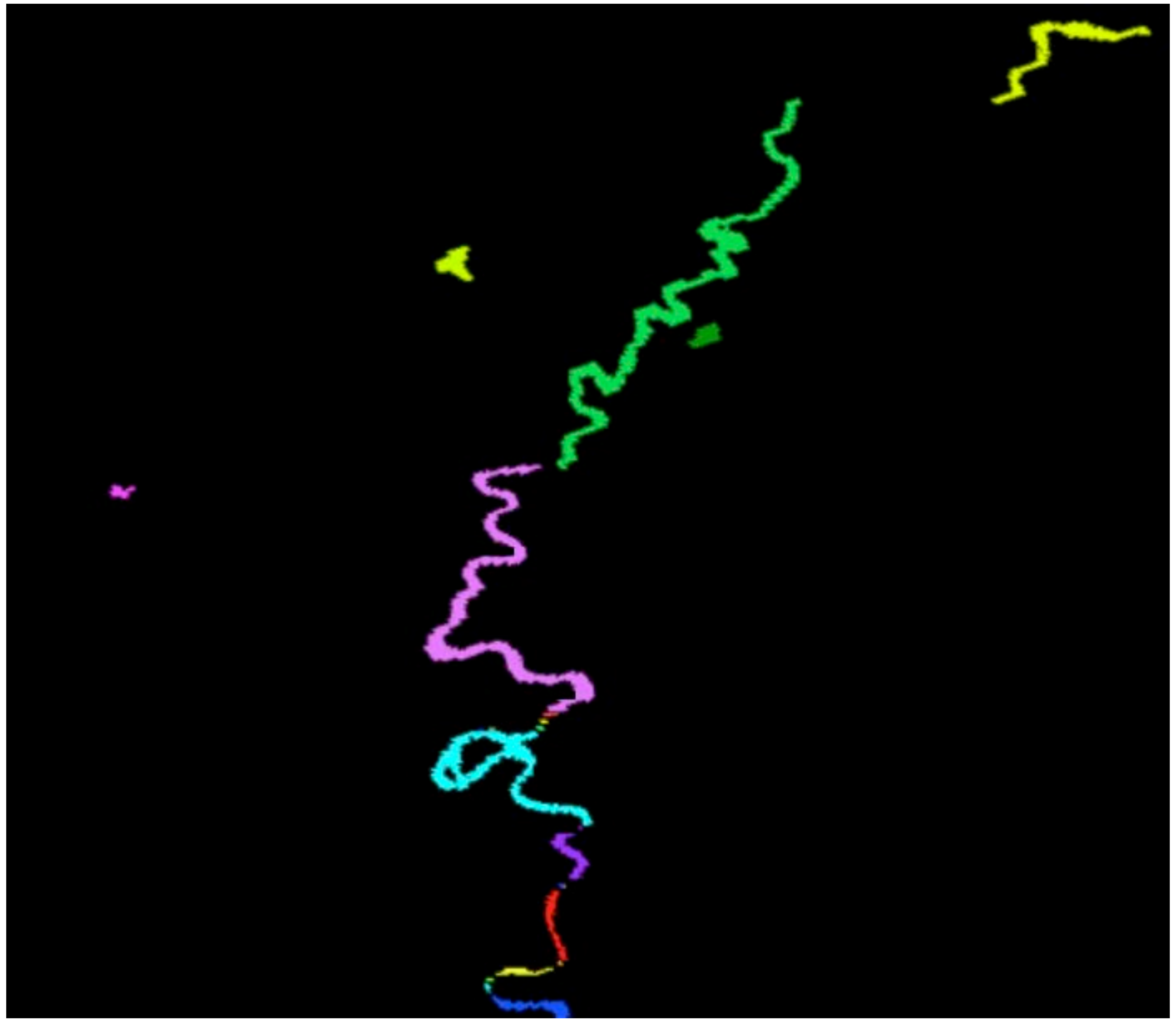
- Lakes connected to river in slant plane can be included within the in-channel segment
  - Hence the thresholds



Segmentation

- Lakes connected to river in slant plane can be included within the in-channel segment
  - Hence the thresholds
- For various reasons, the pixel cloud may be discontinuous even within the river channel

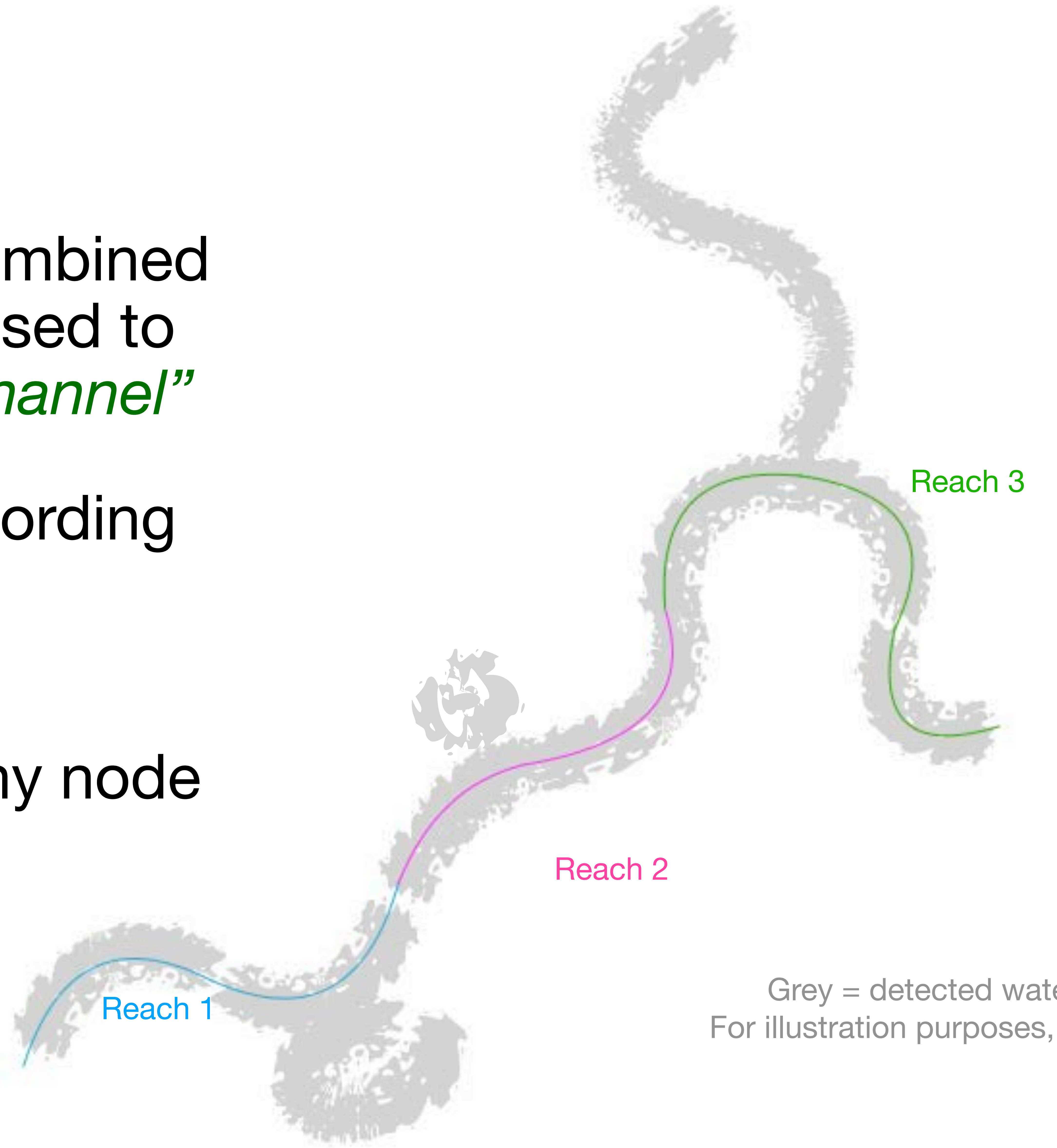
Segmentation on simulated data



Pixel assignment

- The segmentation information combined with prior width information are used to determine which pixels are "*in_channel*"
- Pixels are assigned to nodes according to their shortest distance
- Pixels outside the *"in_channel"* thresholds are not assigned to any node

Cartoon of RiverTile with multiple reaches

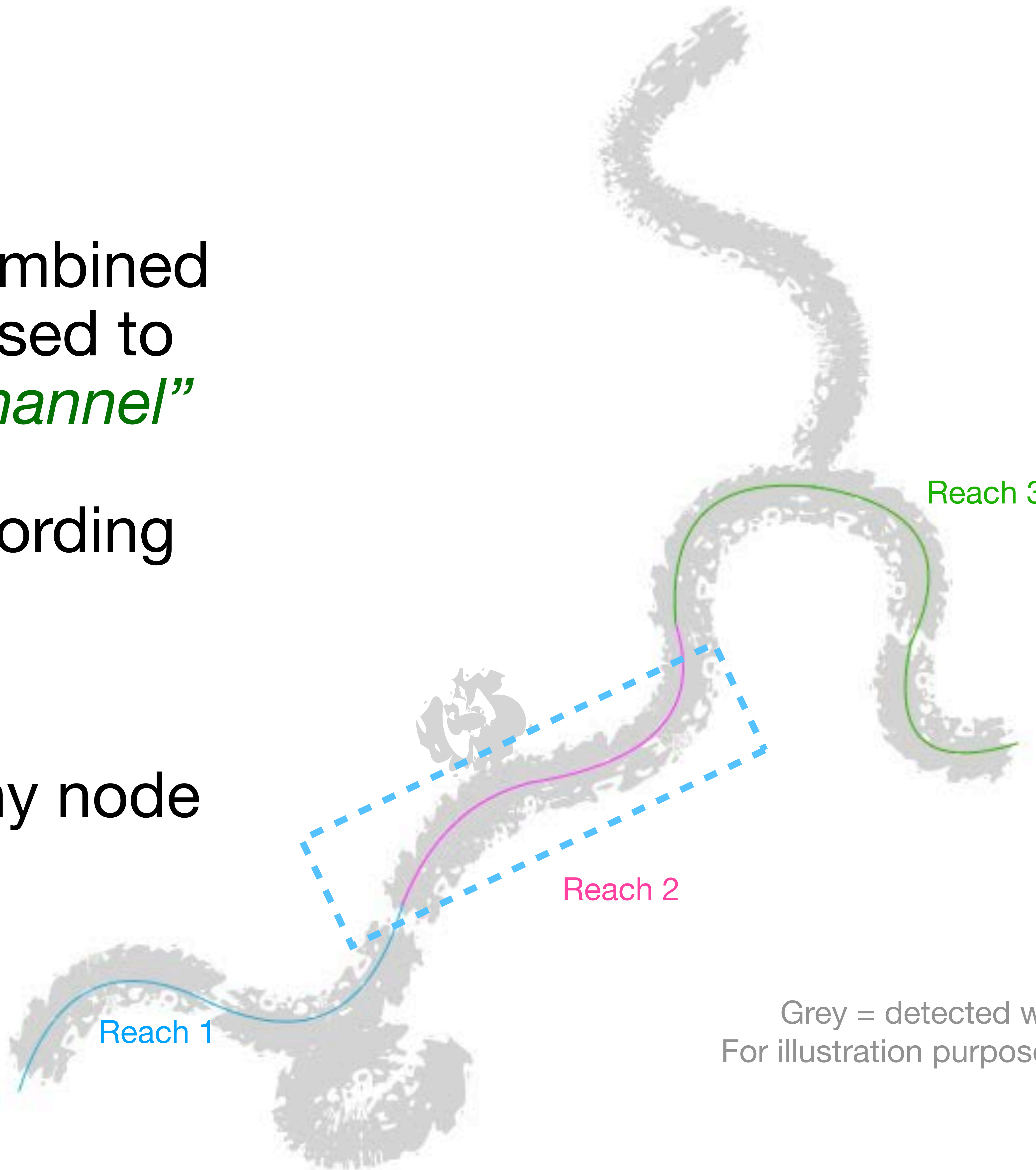


Grey = detected water pixels
For illustration purposes, not to scale

Pixel assignment

- The segmentation information combined with prior width information are used to determine which pixels are "*in_channel*"
- Pixels are assigned to nodes according to their shortest distance
- Pixels outside the "*in_channel*" thresholds are not assigned to any node

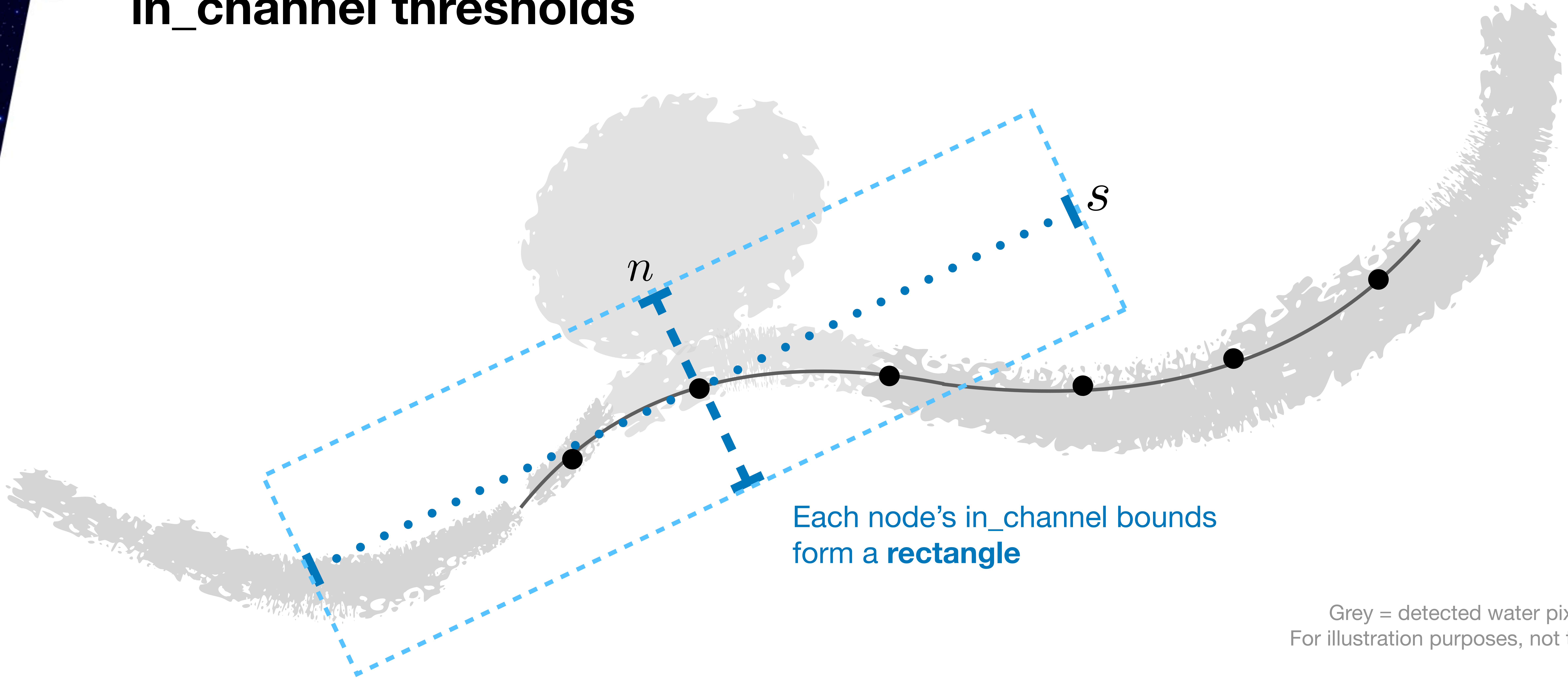
Cartoon of RiverTile with multiple reaches



Grey = detected water pixels
For illustration purposes, not to scale

Pixel assignment to nodes

in_channel thresholds



Each node's in_channel bounds form a **rectangle**

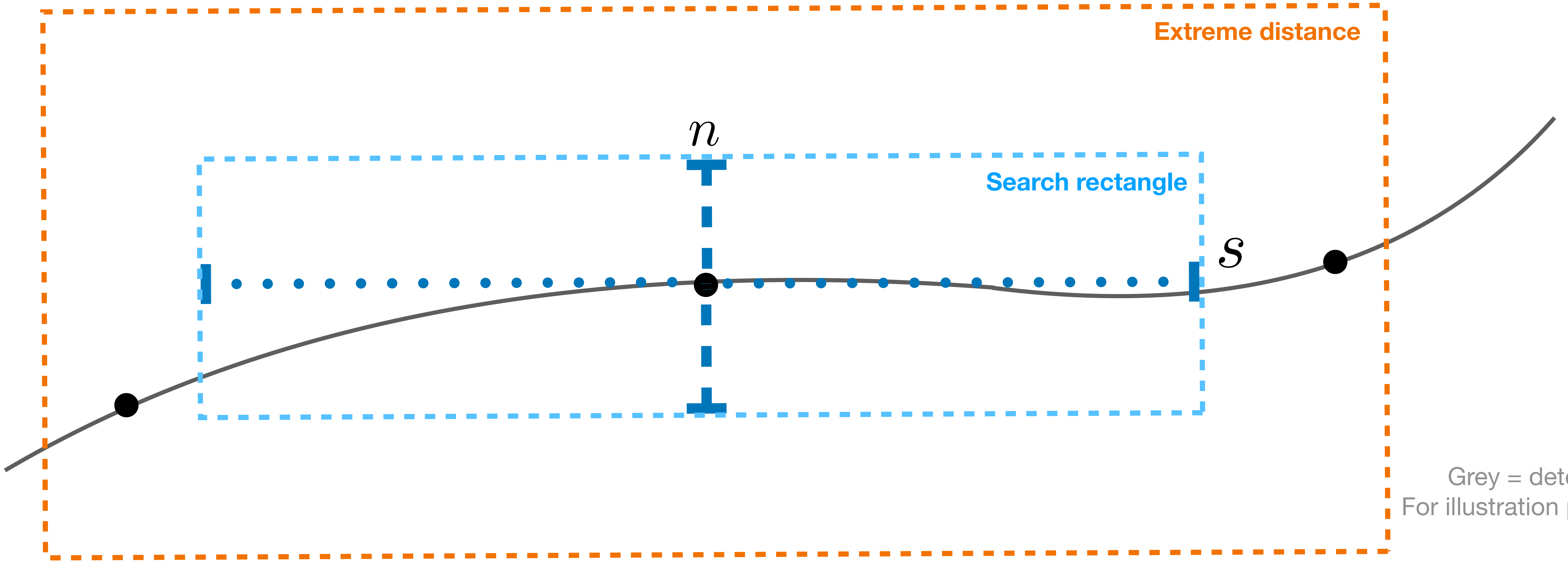
Grey = detected water pixels
For illustration purposes, not to scale

Pixel assignment to nodes

Thresholds

Two thresholds per node:

1. Search rectangle: the distance threshold in which pixels are *always* included
2. Extreme distance*:* the distance up to which pixels *within the segmentation label* are included

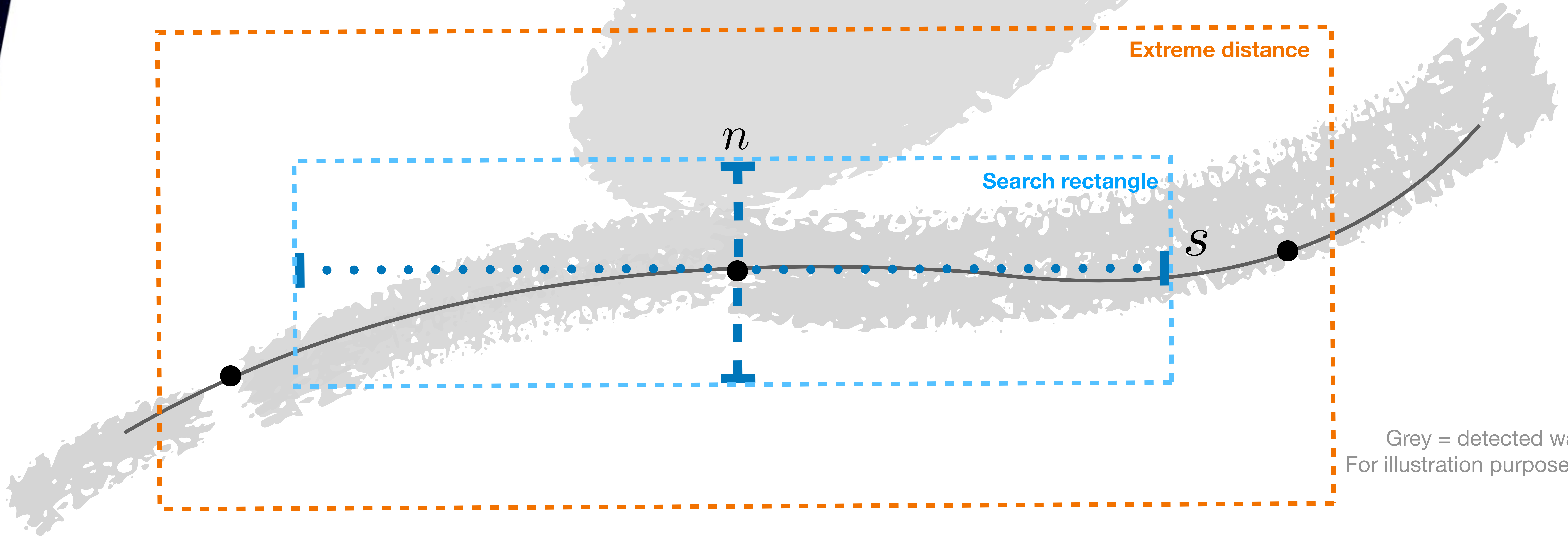


Grey = detected water pixels
For illustration purposes, not to scale

Pixel assignment to nodes
Thresholds

Two thresholds per node:

1. Search rectangle: the distance threshold in which pixels are *always* included
2. Extreme distance: the distance up to which pixels *within the segmentation label* are included



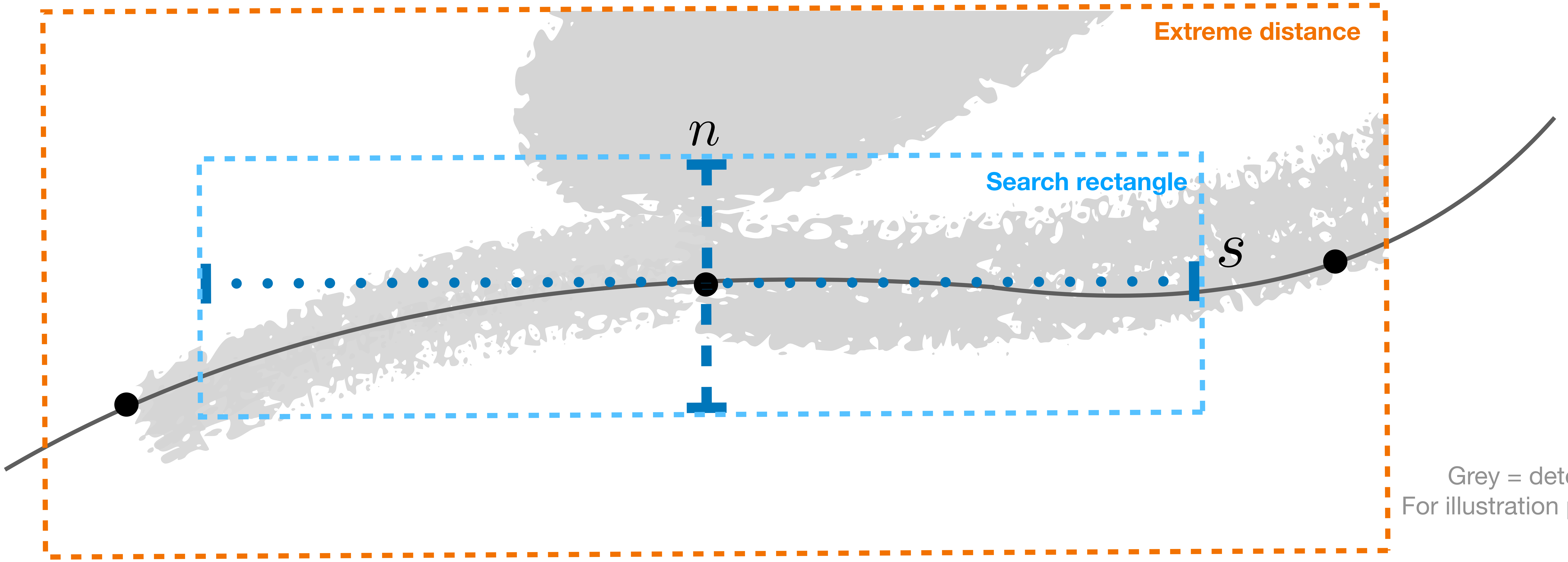
Grey = detected water pixels
For illustration purposes, not to scale

Pixel assignment to nodes

Thresholds

Two thresholds per node:

1. Search rectangle: the distance threshold in which pixels are *always* included
2. Extreme distance*:* the distance up to which pixels *within the segmentation label* are included



Grey = detected water pixels
For illustration purposes, not to scale

Pixel assignment to nodes

Thresholds

Two thresholds per node:

1. Search rectangle: the distance threshold in which pixels are *always* included
2. Extreme distance: the distance up to which pixels *within the segmentation label* are included

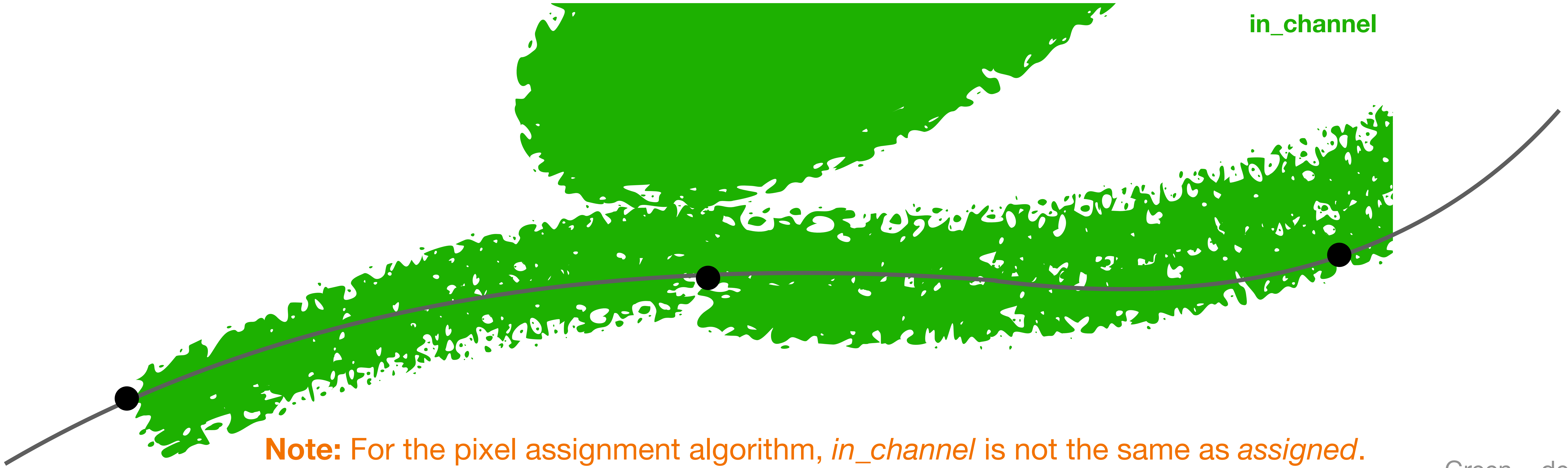
Green = detected water pixels
For illustration purposes, not to scale

Pixel assignment to nodes

Thresholds

Two thresholds per node:

1. Search rectangle: the distance threshold in which pixels are *always* included
2. Extreme distance*:* the distance up to which pixels *within the segmentation label* are included



Note: For the pixel assignment algorithm, *in_channel* is not the same as *assigned*.

Following this step, pixels multiply assigned to nodes will be adjusted and pruned for unique assignment to their nearest node.

Green = detected water pixels
For illustration purposes, not to scale

Pixel assignment to nodes

Thresholds

Two thresholds per node:

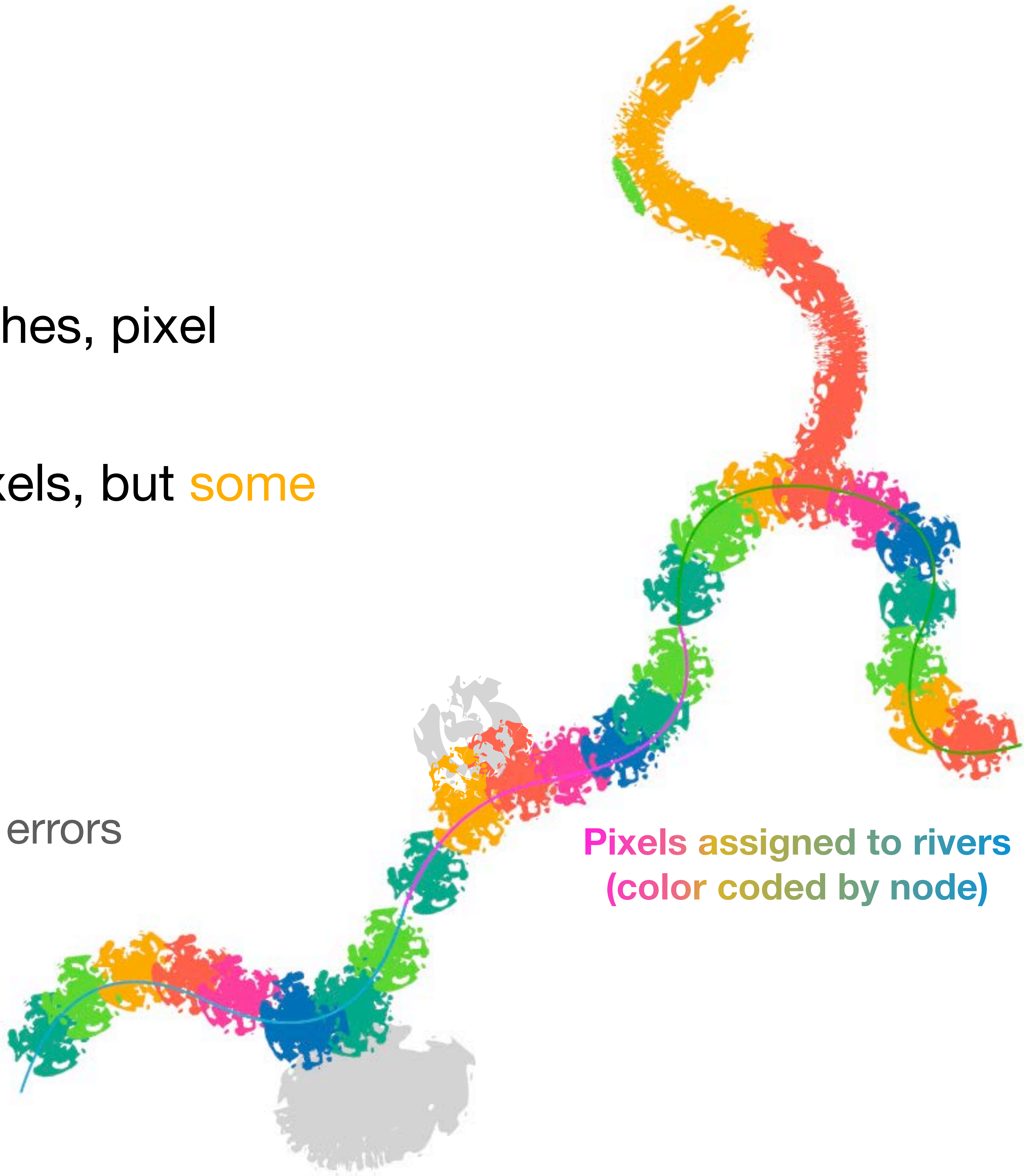
1. Search rectangle: the distance threshold in which pixels are *always* included
2. Extreme distance: the distance up to which pixels *within the segmentation label* are included

Grey = detected water pixels
For illustration purposes, not to scale

Pixel assignment

Unique assignment

- After iterating through all nodes in all reaches, pixel assignment now finished
- Most assigned pixels are river channel pixels, but some pixels may be assigned incorrectly
    - Mid-size tributaries (not in SWORD)
    - Lake-near-river pixels
    - Missing pixels due to dark water or other errors



Pixels assigned to rivers (color coded by node)

Pixel assignment

Things to be aware of

- Pixel assignment often errs on the side of **keeping** pixels
  - False positives due to non-channel waterbodies near SWORD centerline are common
    - e.g. unknown lakes-near-river or artificial reservoirs, moderate width tributaries (not in SWORD), etc.
  - Coherence time/azimuth smearing can also result in increased false positives
    - Somewhat mitigated by water fraction approach
  - Mean/median area error (%) **often positive** in our simulations
    - i.e. river processor tends to over-estimate area

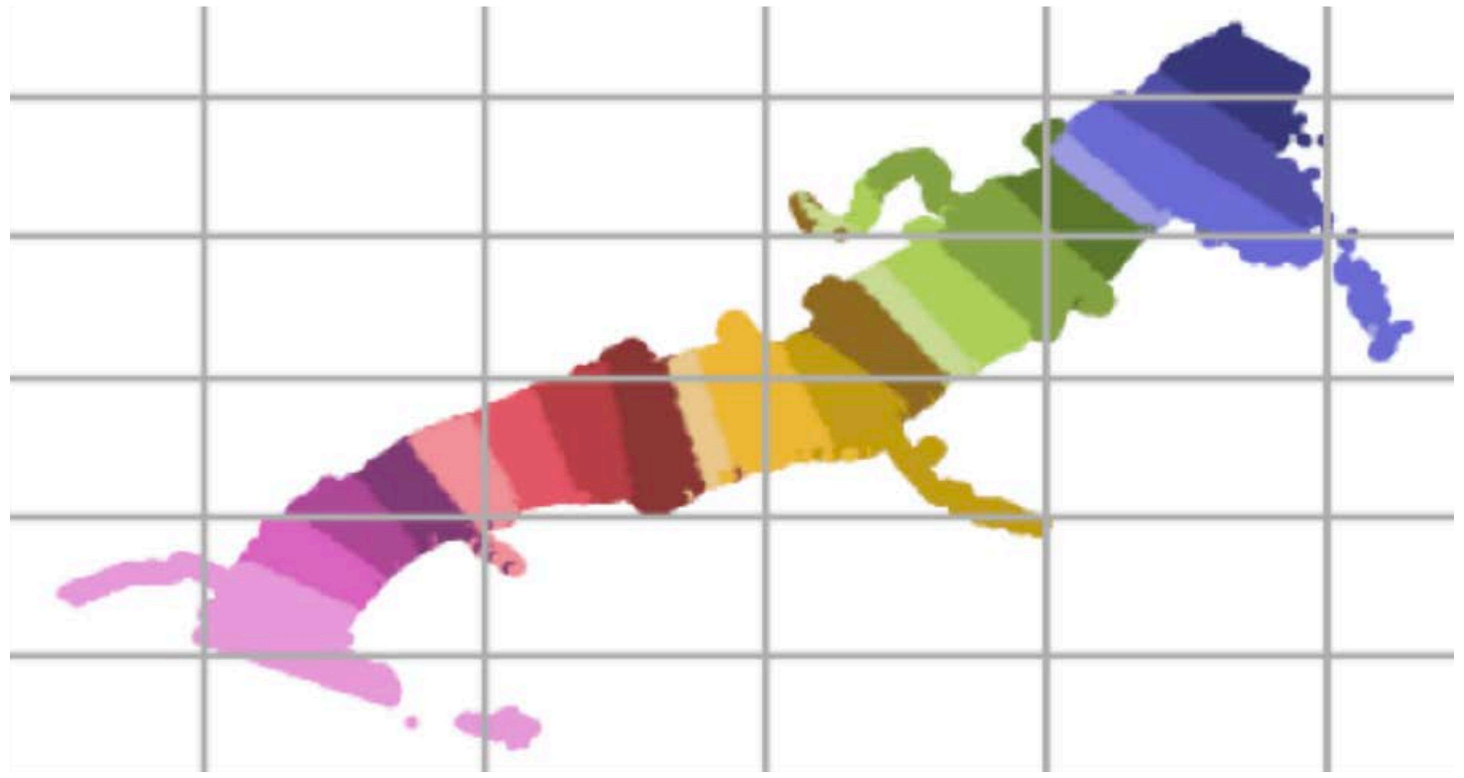
Example pixcvec output

Pixel assignment

Things to be aware of

- Pixel assignment often errs on the side of **keeping** pixels
  - False positives due to non-channel waterbodies near SWORD centerline are common
    - e.g. unknown lakes-near-river or artificial reservoirs, moderate width tributaries (not in SWORD), etc.
  - Coherence time/azimuth smearing can also result in increased false positives
    - Somewhat mitigated by water fraction approach
  - Mean/median area error (%) **often positive** in our simulations
    - i.e. river processor tends to over-estimate area

Example pixcvec output



- Pulling in bad pixel heights can contaminate node-level information
  - Node-level wse/area outliers are relatively common
  - Design philosophy is to **include, but flag** bad nodes, and weight accordingly for reach-level aggregation stats

Pixel Aggregation to Nodes

- Following assignment, pixel information is aggregated to node-level information
  - Each pixel from the input pixel cloud has a height, area, water fraction, and associated uncertainties

Pixel Aggregation to Nodes

- Following assignment, pixel information is aggregated to node-level information
    - Each pixel from the input pixel cloud has a height, area, water fraction, and associated uncertainties

For **area** aggregation, account for pixel area and estimated water fraction to find total area (sum of all estimated pixel areas)

$$\hat{A}_{f,\alpha} = \sum_x A(x)\hat{\alpha}(x)I_f(x)$$

Pixel Aggregation to Nodes

- Following assignment, pixel information is aggregated to node-level information
  - Each pixel from the input pixel cloud has a height, area, water fraction, and associated uncertainties

For **area** aggregation, account for pixel area and estimated water fraction to find total area (sum of all estimated pixel areas)

$$\hat{A}_{f,\alpha} = \sum_x A(x)\hat{\alpha}(x)I_f(x)$$

For **height** aggregation, take the *mean* height using weights based on the phase noise variance (i.e. inverse variance weighting)

$$\hat{h}_f = \sum_x w(x)h(x)I_{f,h}(x) \qquad w(x) = \frac{\sigma_h^{-2}(x)}{\sum_x \sigma_h^{-2}(x)I_{f,h}(x)} = \frac{d_h^{-2}(x)\sigma_\phi^{-2}(x)}{\sum_x d_h^{-2}(x)\sigma_\phi^{-2}(x)I_{f,h}(x)}$$

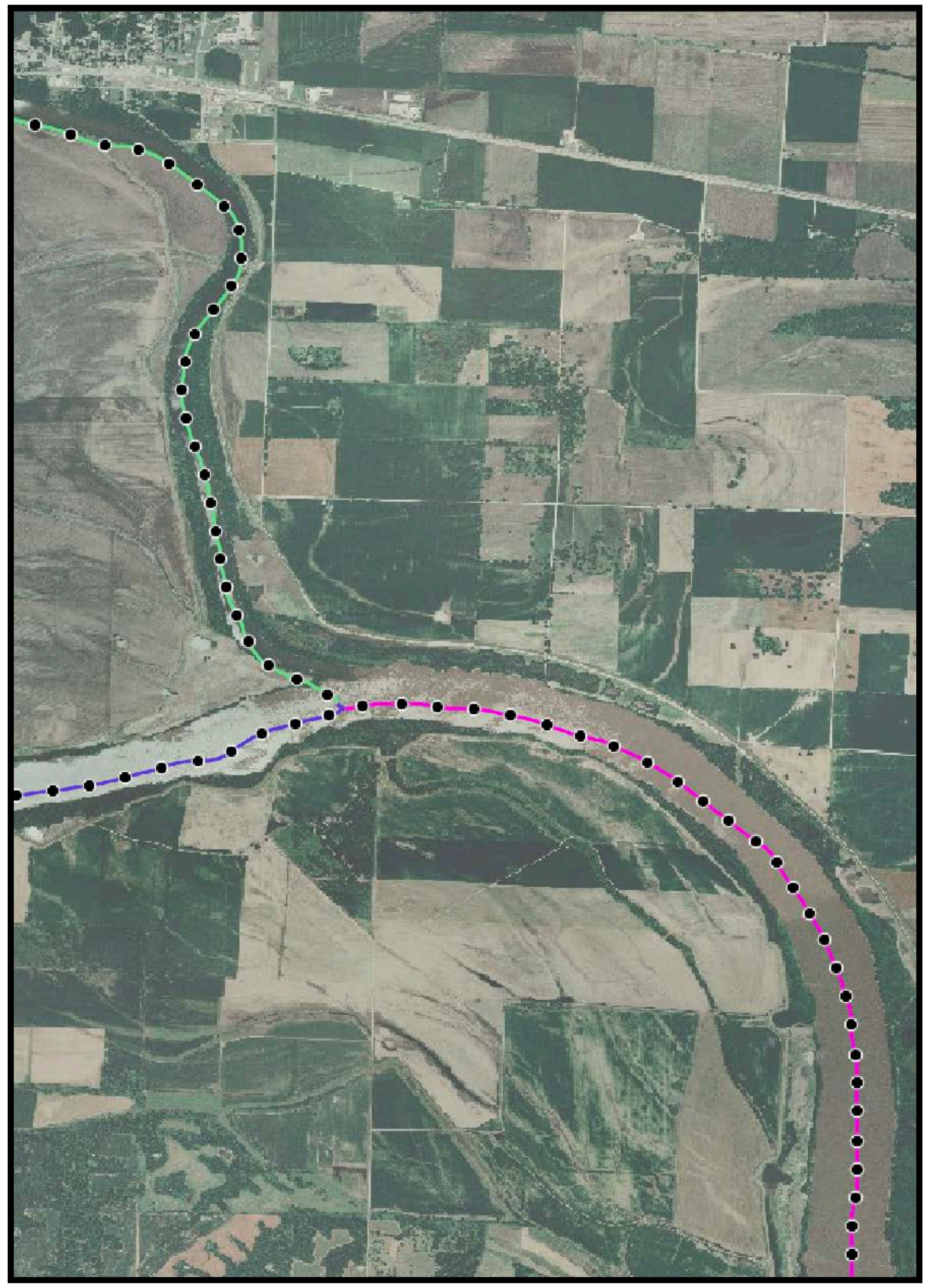
Node aggregation to reaches

For the reaches.shp output product, combine node-level height and area data into reach-level heights, areas, and slope

- Reach-level area is the sum of all node areas
- Reach-level wse is the estimated height at the along-reach center defined by SWORD
  - Height is estimated using estimated reach slope
- Reach slope is currently calculated as a weighted linear fit to the reach using pixels from only that reach
  - May change due to frequency of non-linear reaches in simulation dataset; exploring option to include information from adjacent reaches
  - Slope is only available at the reach level (not for individual nodes)

Reach and node heights are reported relative to geoid with corrections for solid Earth, load, pole tides

RiverSP shapefiles



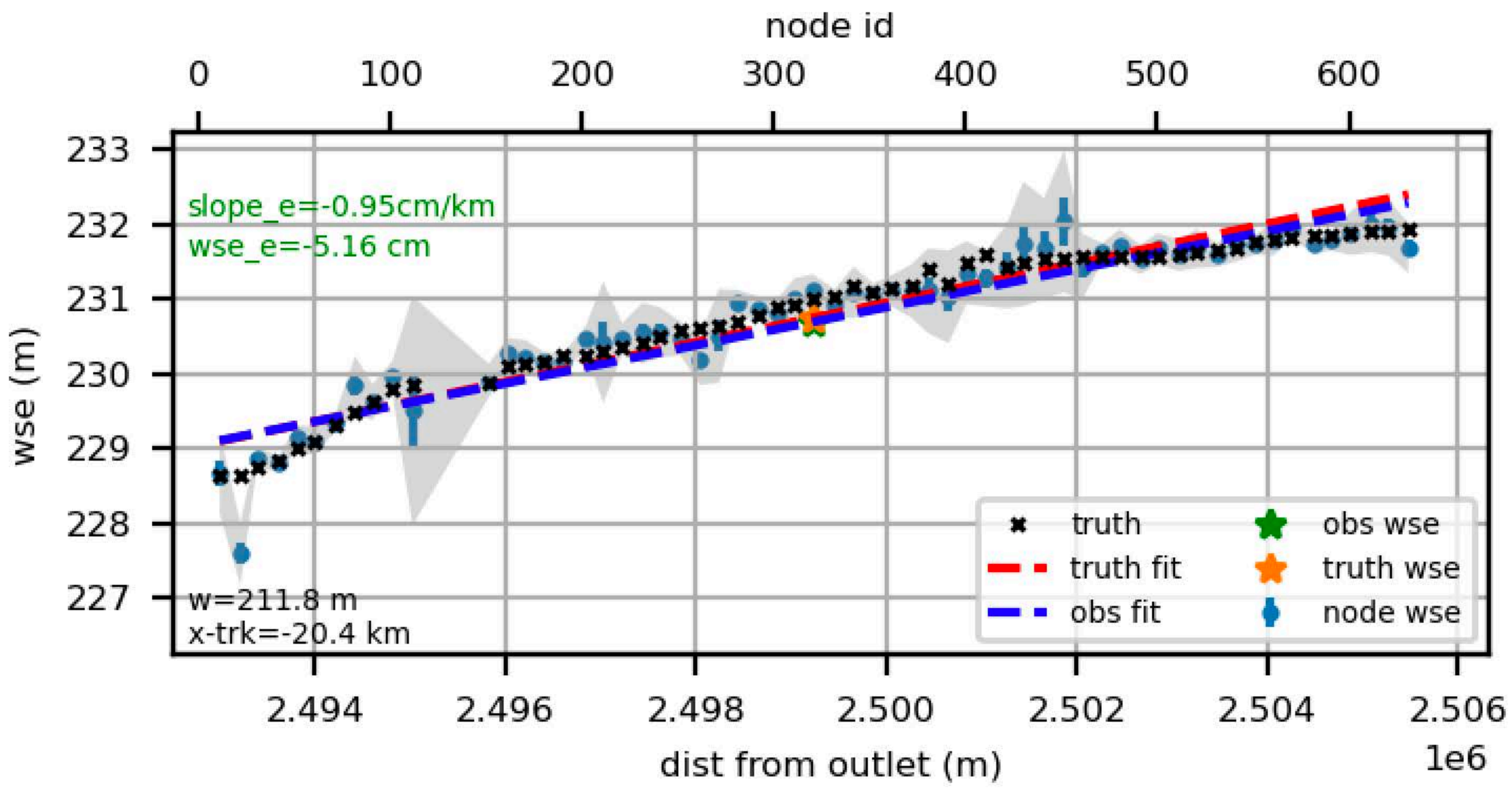
L2_HR_RiverSP: reaches.shp; nodes.shp

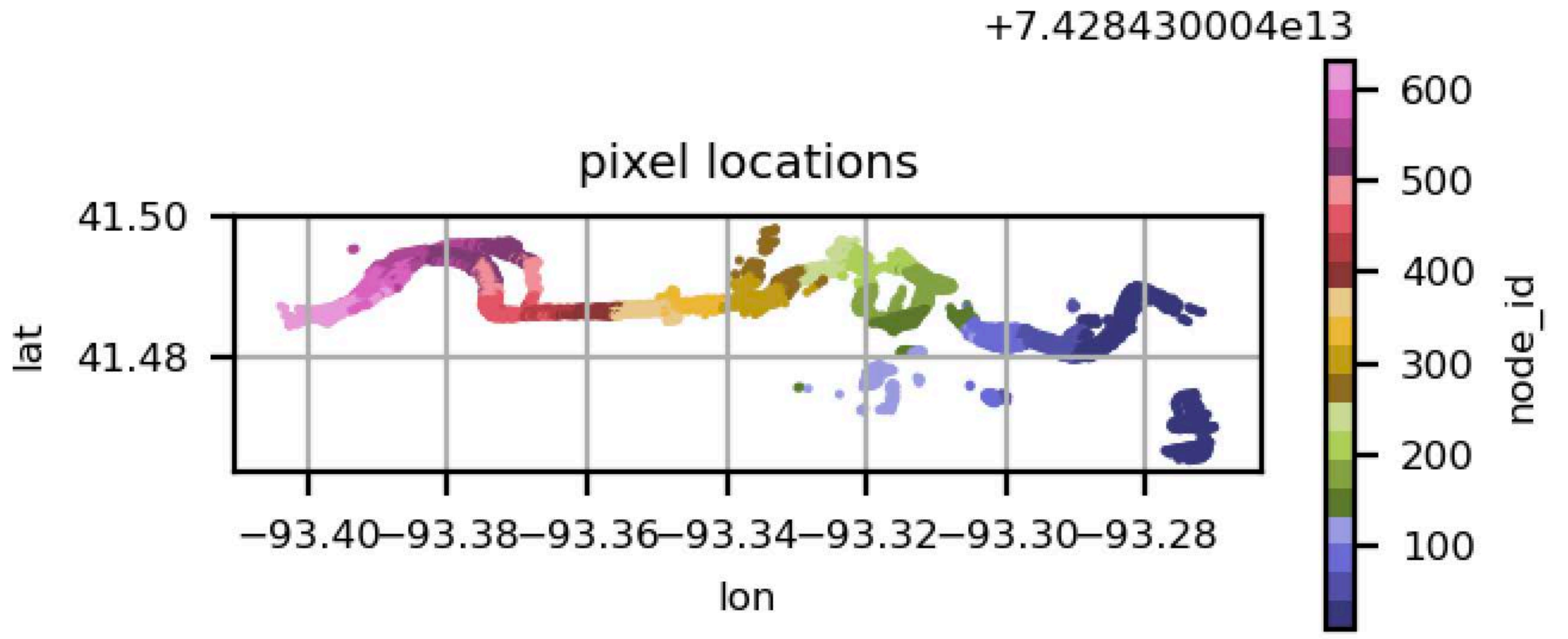
SWOT

Node aggregation to reaches

Example result

Simulated node-level outputs with reach-level slope fit

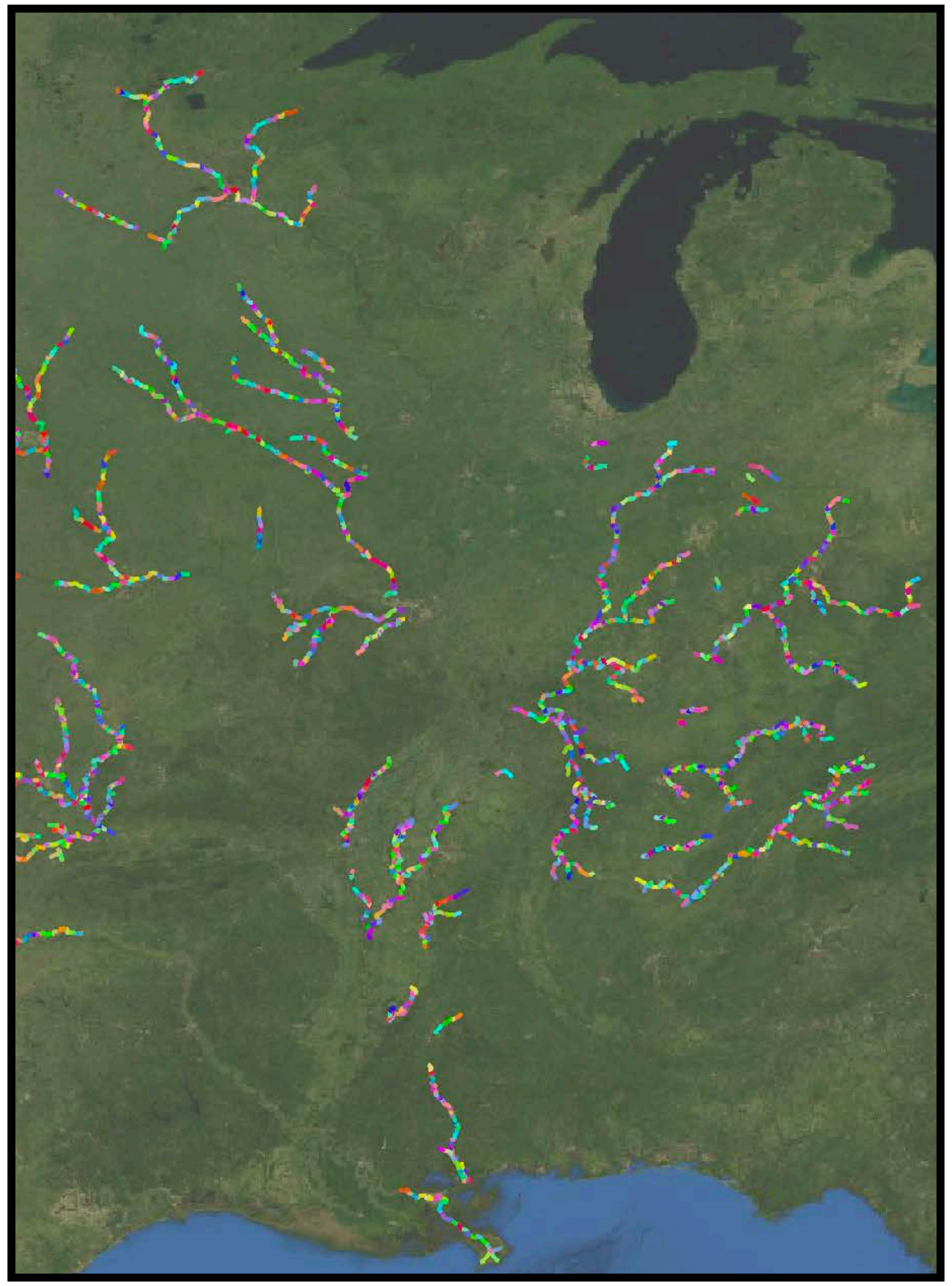




River Average Product

- The RiverAverage product provides basin-scale (Pfafstetter 2) SWOT-derived reach information averaged over the ~21-day repeat cycle
  - Observations per cycle range from 1-36, depending primarily on latitude
  - Reach-level information only (no nodes)
- Aim is to provide temporally averaged data for large scale hydrological models

RiverAvg product

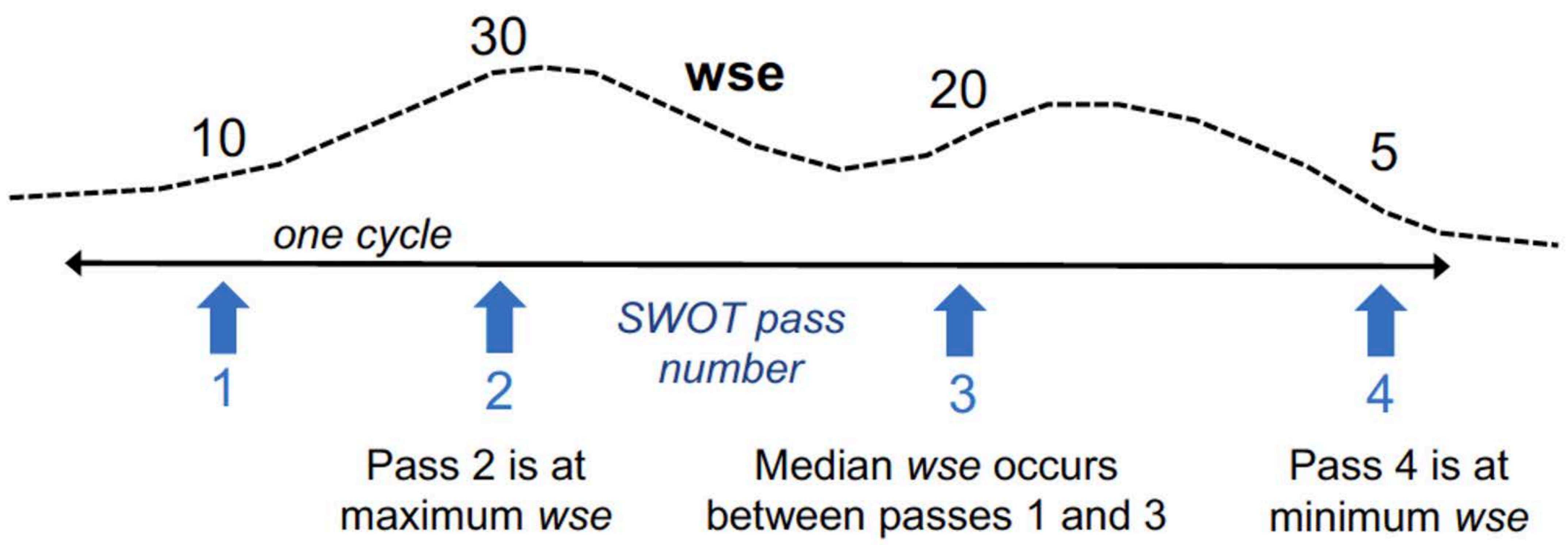


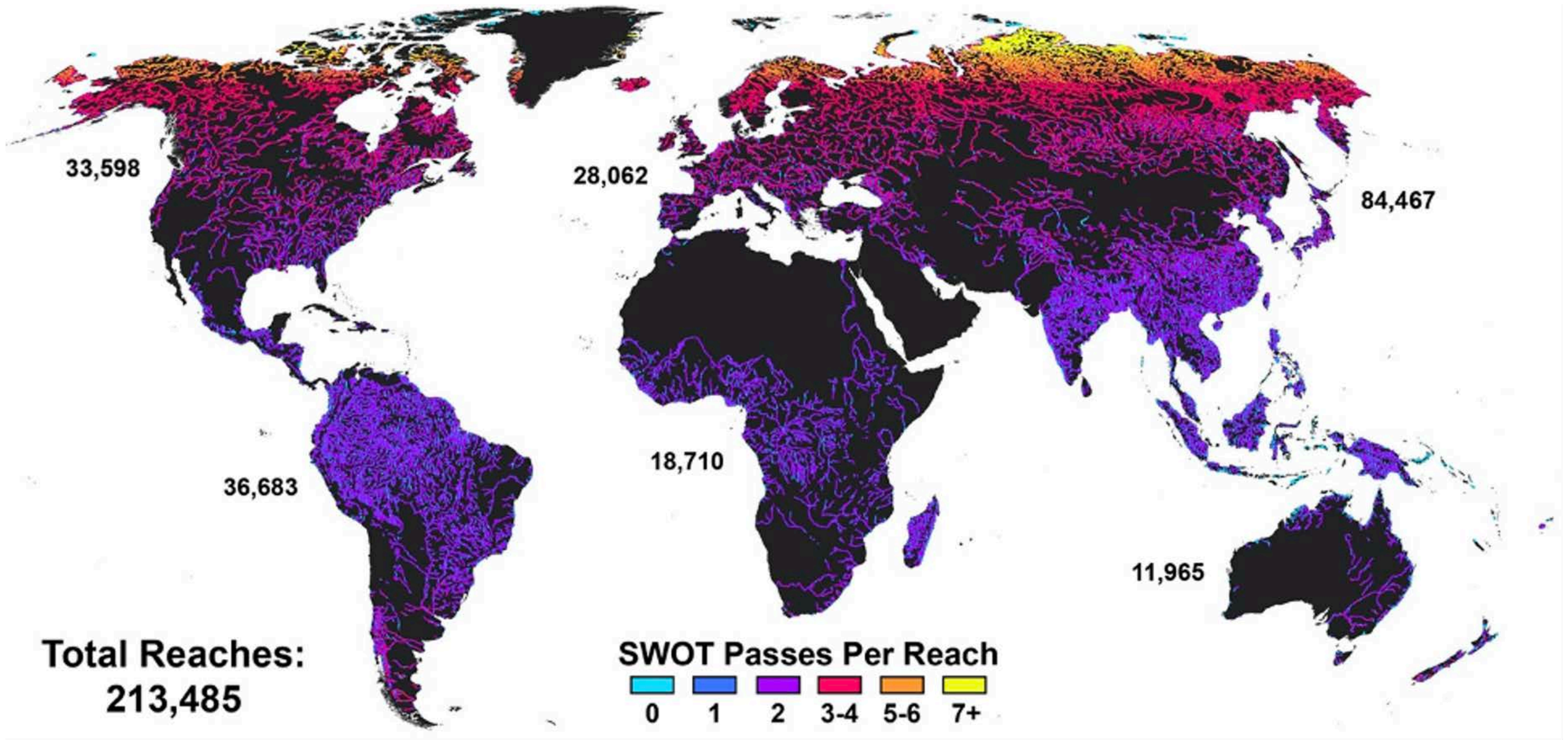
SWOT_L2_RiverAvg_007_74

River Average Product

Note for users: RiverAvg product convention is “wse forward”

- Report values & orbits corresponding to **min, med, and max *wse*** for all passes with valid wse
- Area, width, slope, and discharge values are provided in reference to the **passes** associated with min, med, and max ***wse***
- Mean values for wse, area, width, and slope are the simple means over all orbits





Summary

SWOT River Processing

- River processing takes input pixel cloud and SWORD data to generate **reach** and **node** single-pass shapefiles
  - Pixel assignment distinguishes river channel pixels from land & other waterbodies
- Cycle-average product gives reach-level information per 21 day orbit cycle
- **Considerations**:
  - Pixel assignment errors can cause height, slope, or area errors
    - Bad nodes, various node flags
  - Upstream factors
    - Dark water, phase unwrapping, water-water layover

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov