

# GENERALIZATION OF BRAIDED STREAMS

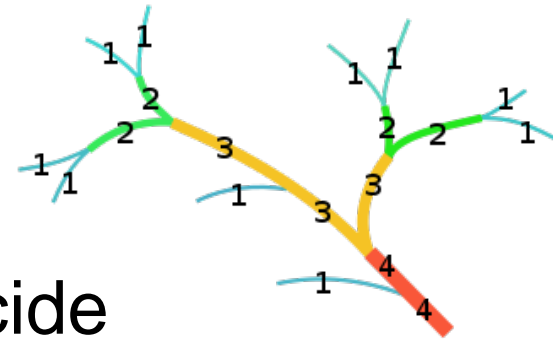
S.Savino

Department of Information Engineering  
University of Padova (ITALY)

# Introduction

## Generalization of rivers:

- Important generalization topic
- Addressed by many works
  - Base assumption: a river is like a tree
  - Evaluate the importance of each branch using flow direction and other metrics (eg. Horton) to decide the ones to remove



# A river is not always like a tree



A tree



A river



A river but not a tree

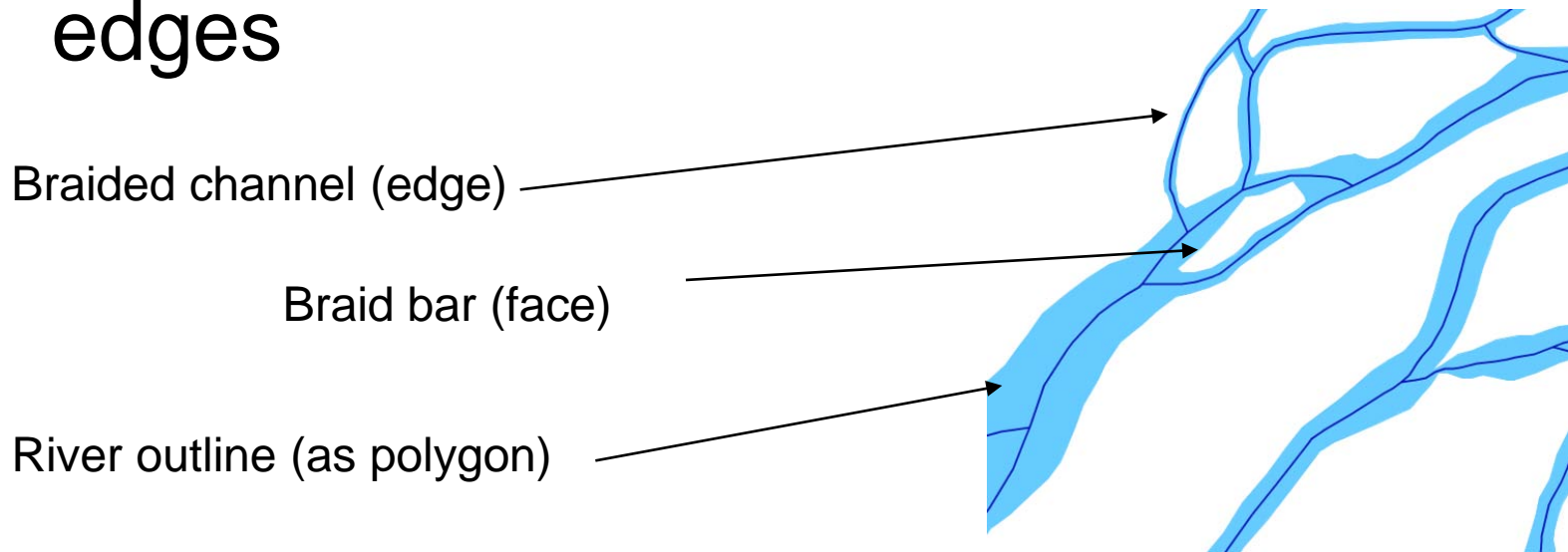
# Braided streams

- Topology is not tree-like
- Many loops and “flat”
- Difficult to estimate flow-directions: many algorithms fail to detect a hierarchy among the branches



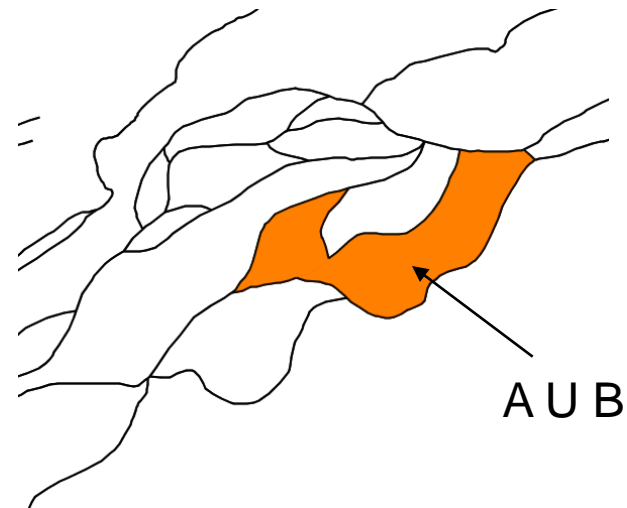
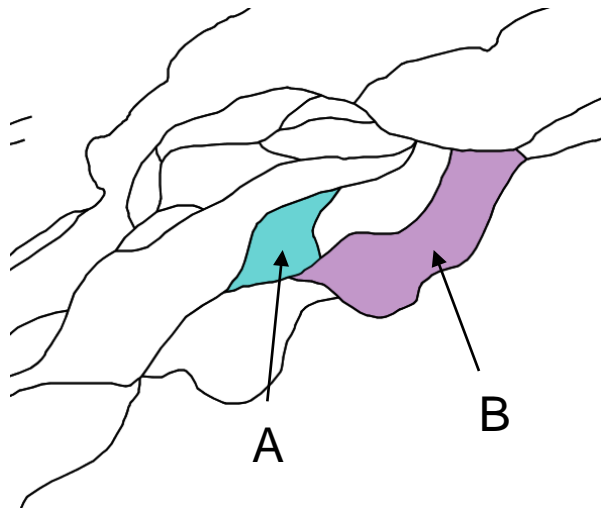
# Our approach

- Braided section as a planar partition
- Use the duality between faces and edges: amalgamate neighbouring faces to delete edges



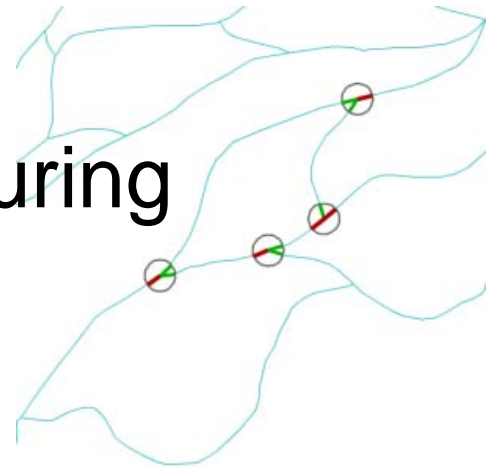
# Area amalgamation

- Well known generalization technique
- Generalizing braided streams not all the faces can be amalgamated

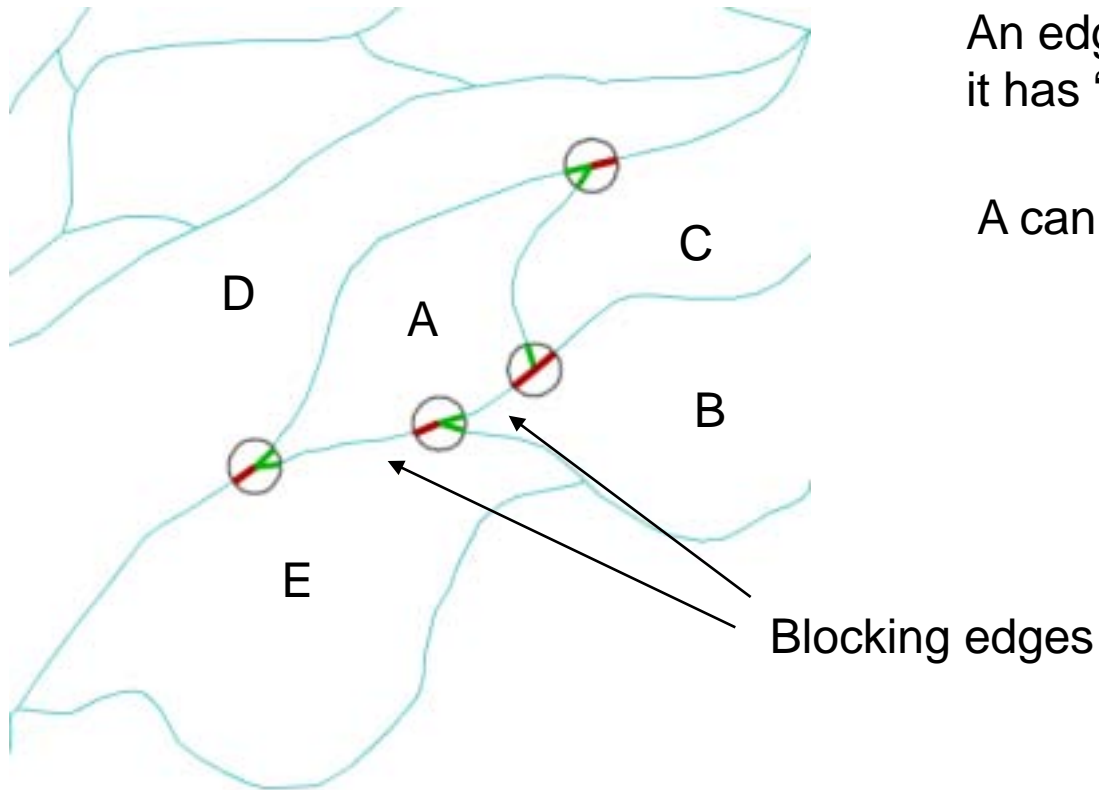


# Blocking edges

- The algorithm avoids hard bends or narrow angles in the resulting network identifying “blocking edge”
- A blocking edge can not be removed during amalgamation
- Blocking edges are found measuring angles between the edges



# Blocking edges



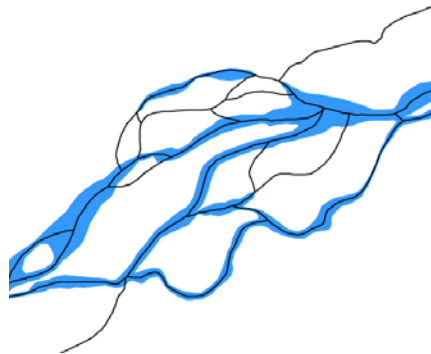
An edge can be removed only if it has “green light” on both ends

A can not be merged to: B and E

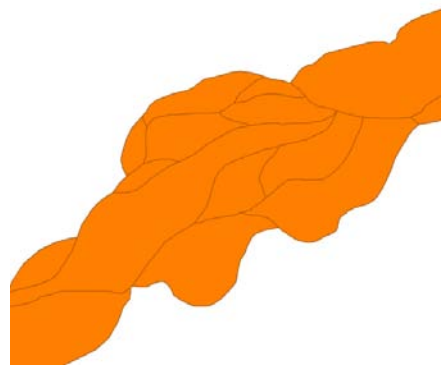


# Wide channels

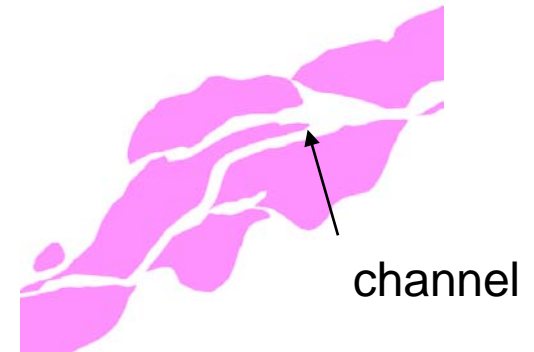
- In braided streams the breadth of the channels may vary considerably
- Removing an edge representing a very broad channel should be discouraged



Network and outline



Network faces



Islands (braid bars)

# Choosing the partner

When evaluating the candidates for amalgamation

- Keep streamlined shape of river: prefer amalgamation that produces compact areas
- Keep in mind the shape of the river outline: try to reduce water loss

# The process

```
compute faces of network
compute the geometry of each island and its area

while all islands generalized = false
  get smallest island i
  if area > threshold
    all islands generalized = true
    break

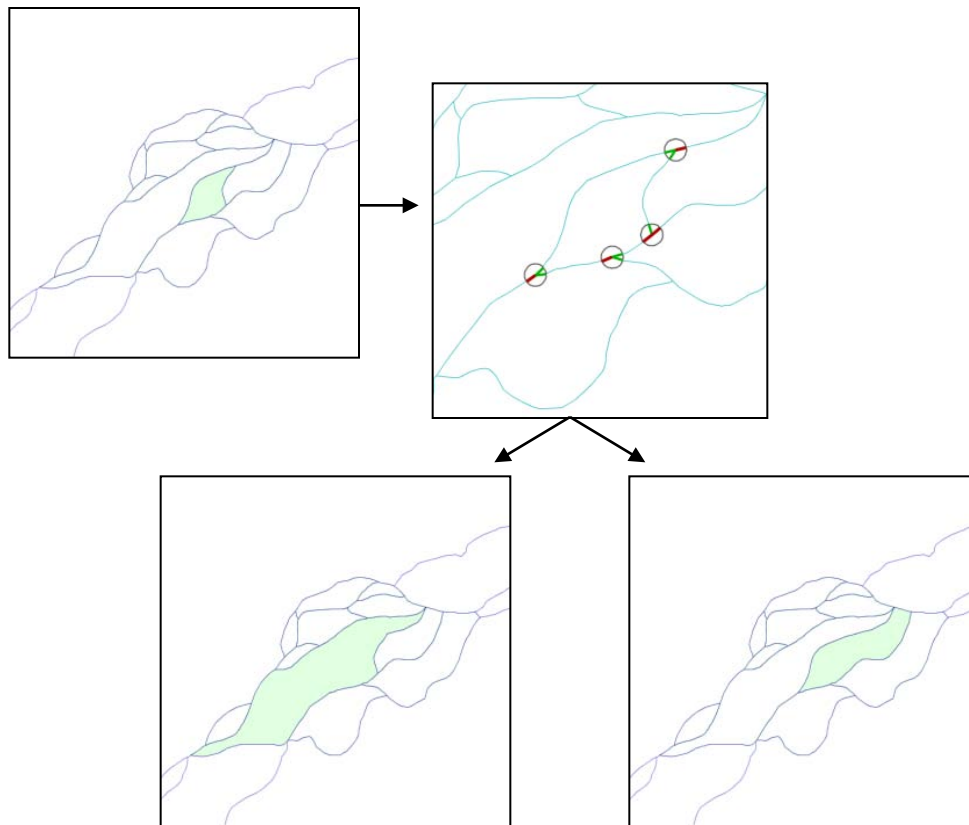
  find neighbouring islands of i

  if no neighbours
    generalize i as lone island
    continue

  foreach neighbour n of i
    find edge e shared between n and i
    if e can not be removed
      continue

    compute geometry union u = n U i
    compute compactness of u

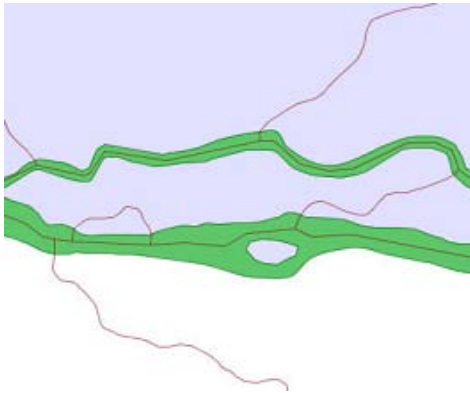
  find geometry u with best compactness
  delete i
  set n = u
```



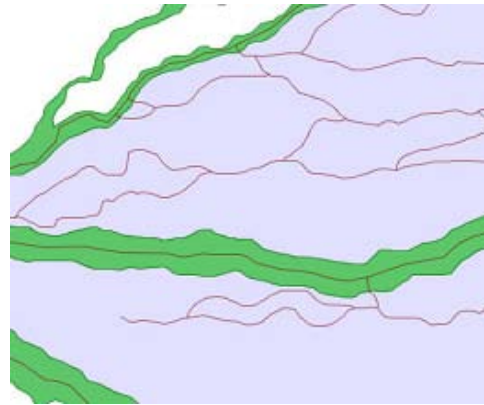
Choice of two candidates for amalgamation

# Special cases

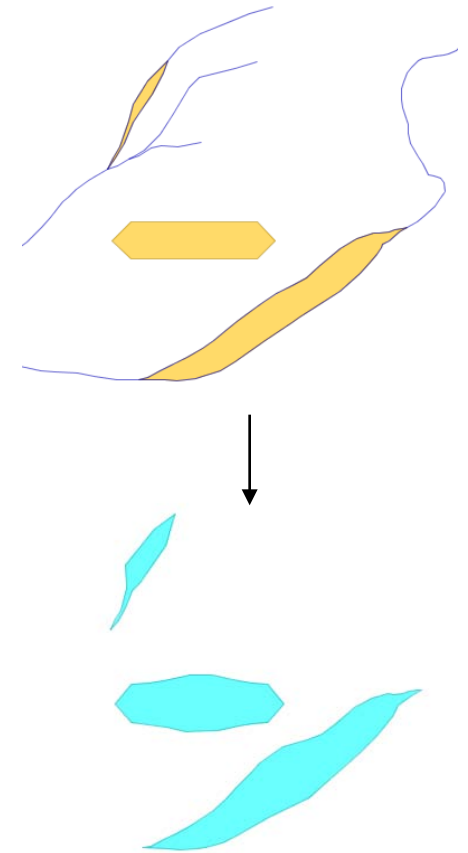
- Holes in the river
- Dangling edges
- Lone islands



Holes are treated as islands

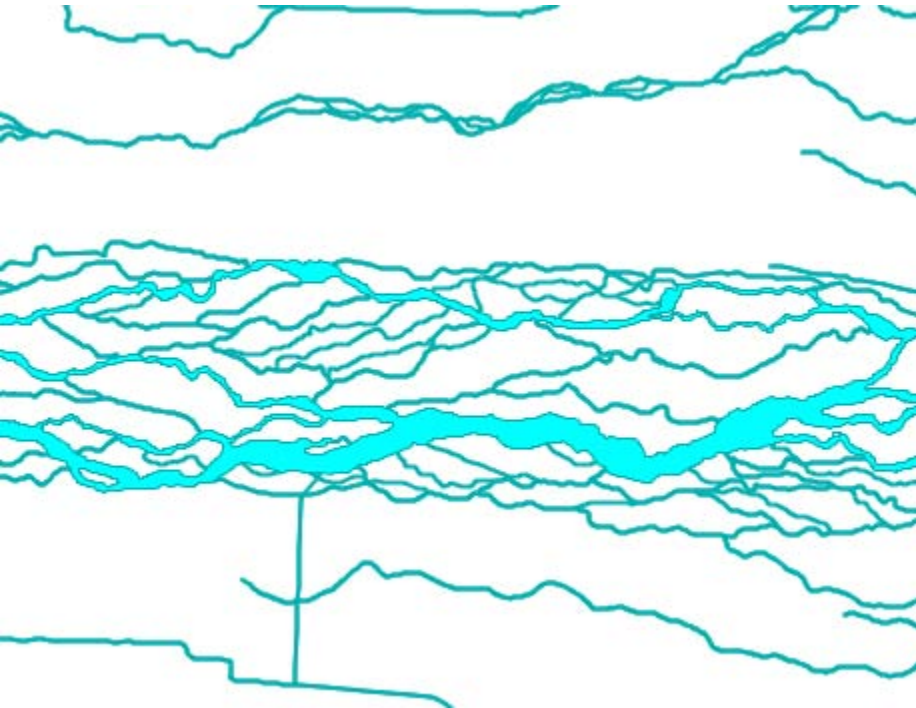


Dangling edges are generalized according to the island around them

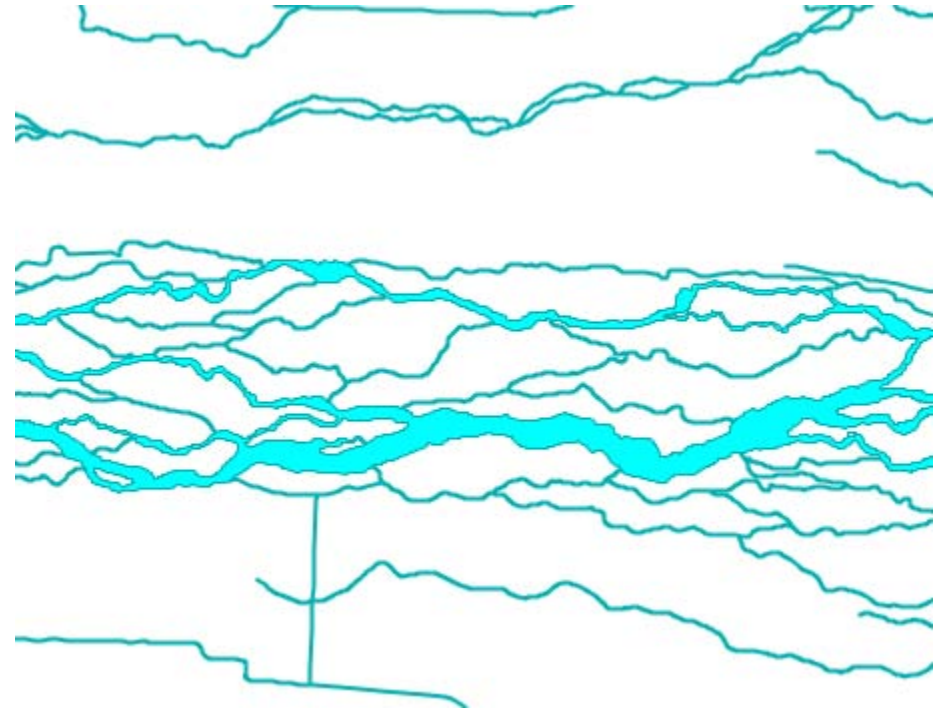


Lone islands are enlarged

# Test and results

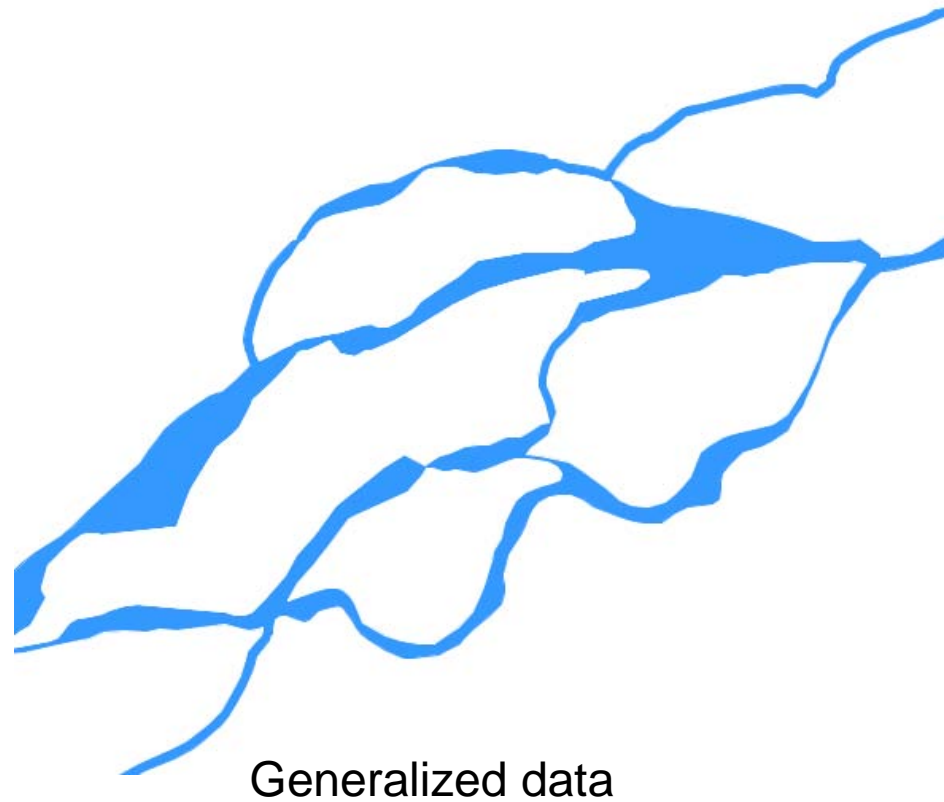
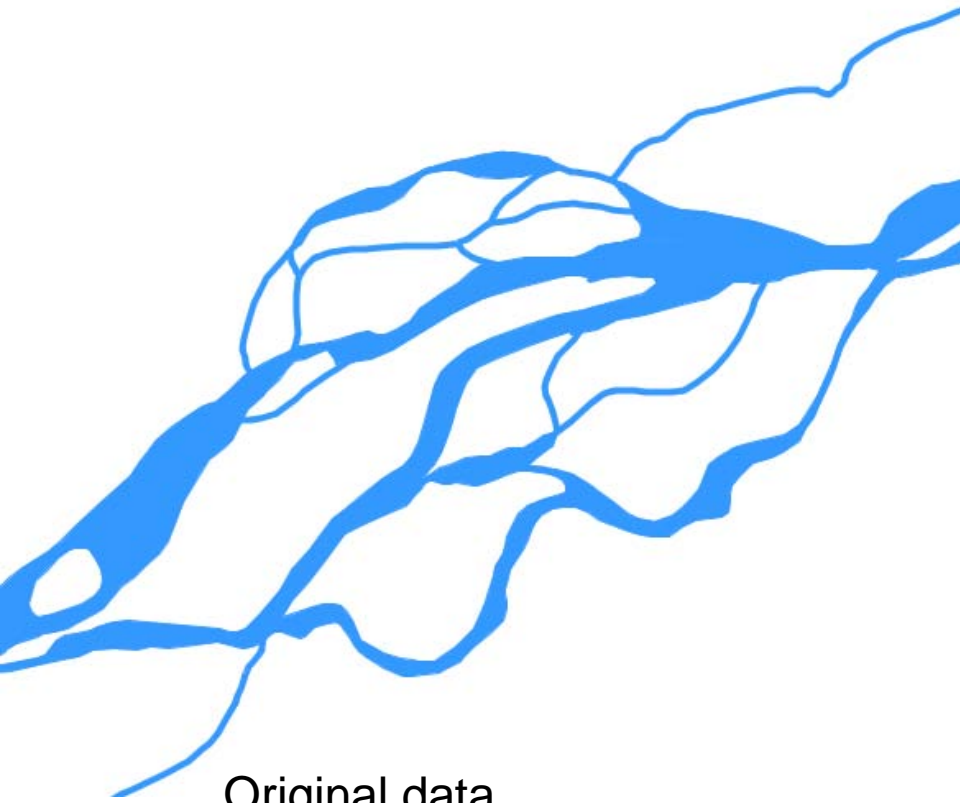


Original data



Generalized data

# Test and results



# Conclusions

- First tests gave good results
- Computing blocking edges on network pose constraints stricter than necessary
- Outline of generalized areas can be improved
- Applicable to other networks (eg. Railroads)