1 Using JTS/JCS for Matching 1:20k Streams With 1:50k Streams

1.1 Background

In order to preserve a large legacy data base of various habitat and inventory information, the 1:50 000 Watershed Atlas (WSA) must be "conflated" with the 1:20 000 Corporate Stream Network (CSN). "Conflation" is the process of identifying digital mapping features which are logically identical but are have slightly different spatial representations.

1.1.1 Matching

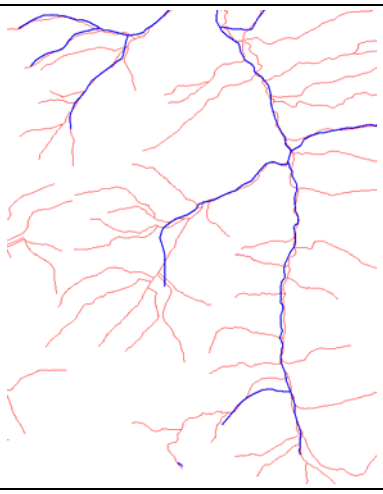
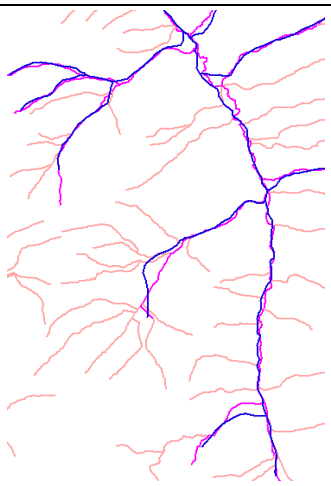
The British Columbia stream conflation problem is a "multi-scale" conflation problem. An older stream network, at 1:50K scale, has been used for many years as the spatial basis for fisheries and habitat inventory databases. New requirements for spatial accuracy require a 1:20K stream network.

The new network describes the same water features as the old network, but because the data were collected using different resolutions of input data, the interpretations of stream locations and network connectivity differ between the two products.

The stream conflation problem is to identify those streams in the new network which are logically the same as the streams represented in the old network. Because the matching is against streams, some additional information can be used to enhance the matching process:

- The two networks tend to describe the same flow behaviors.
- The 1:20K network is generally a superset of the 1:50K network.
- Streams which are the logically the same have similar spatial locations.

	
These dark blue lines represent a small river network taken from the 1:50k BC Watershed Atlas.	These red lines represent the same area but are from the 1:20k Stage 1 CSN.

	
The dark blue (50k) data is layered on top of the red (20k) data.	The matching process correlates the 20k river segments to the corresponding 50k rivers. The matched 20k segments are shown in purple, the unmatched segments are in red.

1.1.2 JTS/JuMP/JCS

The Java Topology Suite (JTS) is a Java API that implements a core set of spatial data operations using robust geometric algorithms that follow the OpenGIS Simple Features Specification definitions (http://www.opengis.org/techno/specs/99-049.pdf). The JTS toolkit allows Java programs to easily create Java spatial objects such as points, lines, and polygons as well as determine relationships between these objects (i.e. touches, contains, and overlaps), and perform operations on them (i.e. intersection, union, symmetric difference). More information on the JTS toolkit is available at http://www.vividsolutions.com/jts/jtshome.htm.

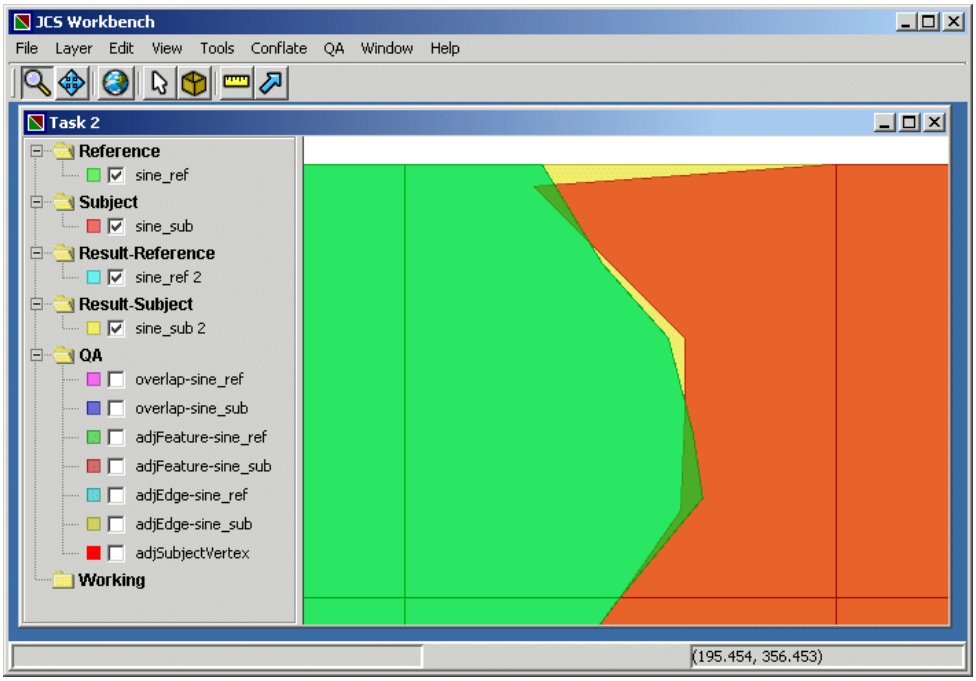
The Java Conflation Suite (JCS) is an extension of JTS that provides richer spatial objects and algorithms specifically for dealing with conflation-type problems. It provides access to complex algorithms for validating, cleaning, and matching datasets.

The Java unified Mapping System (JuMP) was created at the same time as the JCS to provide a graphical user interface (GUI) and toolkit that allows custom process flows to be easily implemented. JuMP provides an extendable viewing, querying, and editing interface similar to other GIS software such as ESRI's ArcView product.

Designed with conflation in mind, the JuMP platform allows Java classes and UIs to be plugged into the main interface environment. These classes normally provide a fully integrated interface that will guide the user through the matching process by controlling the process flow and providing access to the power conflation routines provided by the JCS toolkit.

For programmers, JuMP provides an easily extended UI platform, access to JCS and custom conflation function, as well as low-level access to the JTS. For users actually doing conflation work, the resulting application is provides real-time guidance, validation, and advanced features not available in other GIS software that facilitates the

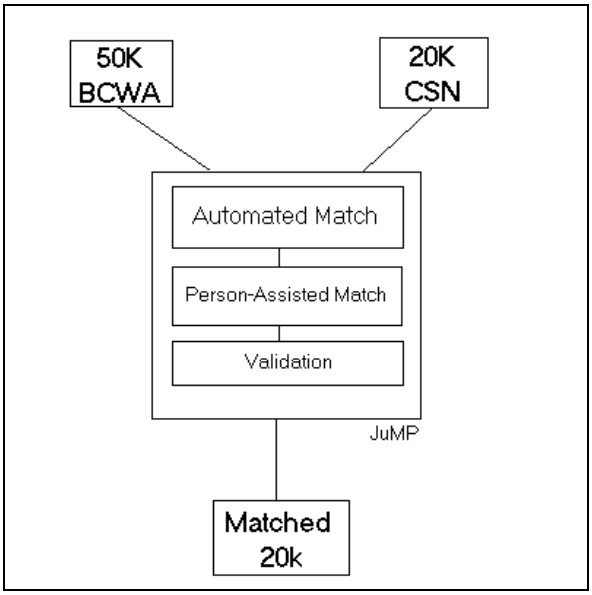
person assisted portion of the matching process. The resulting application leverages the underlying technology to minimize the actual work required by the human operators.



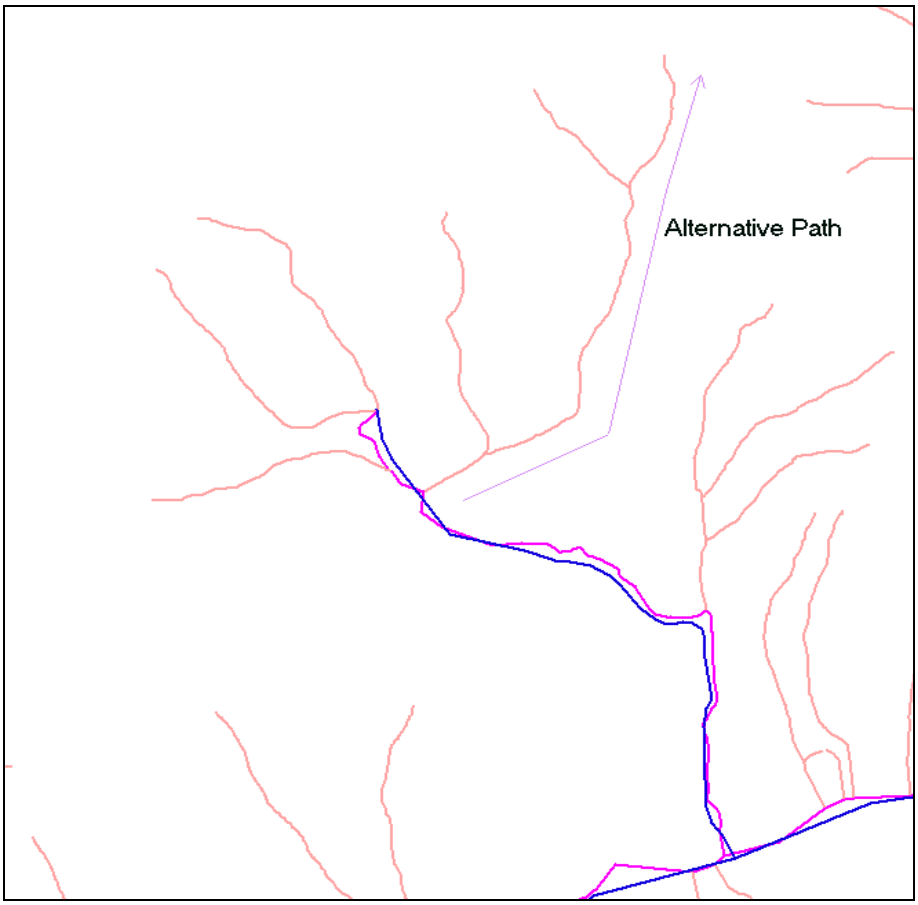
Here JuMP is being used to detect errors in a polygonal coverage. A JCS conflation algorithm will be used to modify the polygons so they share a common edge.

1.1.3 Approach Investigated

The basis of the matching process is a JuMP application that provides access to very specialized conflation algorithms, a process flow UI to ensure all problems are proper address, specialized UI tools to make intricate corrections trivially applied, and validation to ensure no errors are introduced.



The driving principal behind the matching process is to ensure that the best *geometric* match is made, and metrics about the quality of the match is also retained. The best geometric match means that the 20k path chosen to represent a 50k river is the one that is the *least* spatial distant.

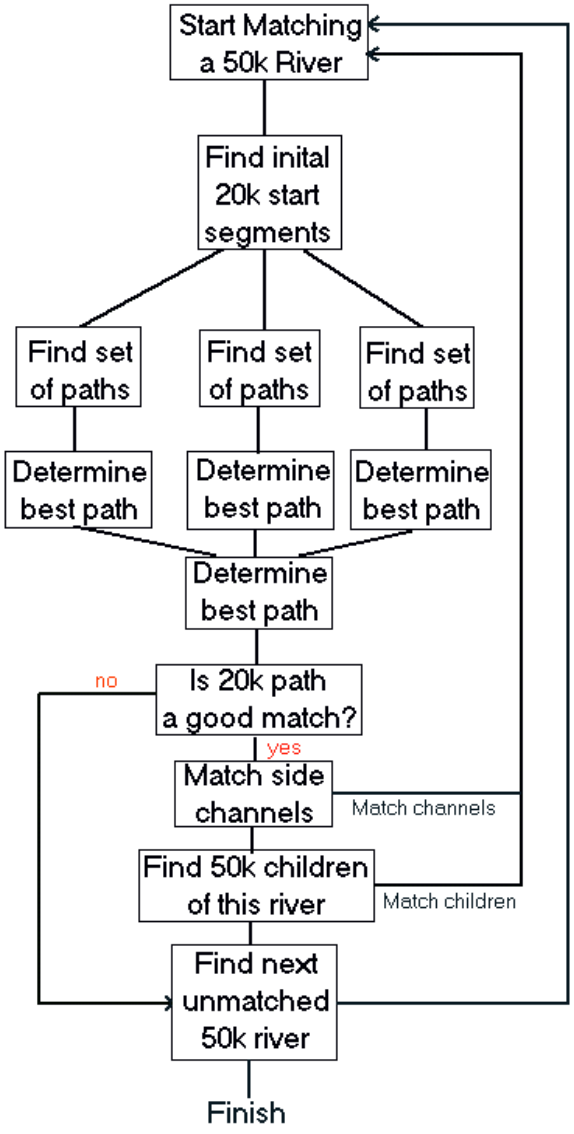


Best geometric match (purple 20k) between the
50k (dark blue) and 20k (red) datasets.
An alternative path – *longest stream* – is also shown.

1.1.4 Automated Matching

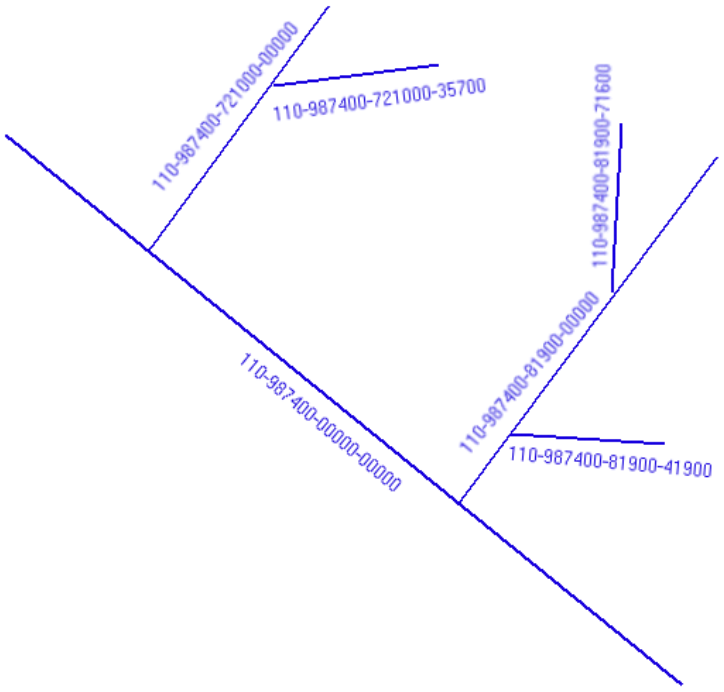
The automated matching process takes the original 50k and 20k data, and finds all the good matches it can find. The basic flow of this matching process is shown below. The rest of this section provides details for most of these steps.

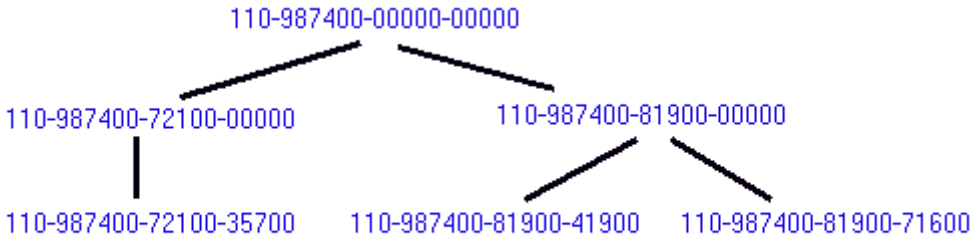
The basis for matching is to follow the hierarchy already inherent in the 50k river network – both the parent-child relationship, but also parent-side-channel relationships. Following these hierarchies makes ensuring connectivity and flow relationships easy to validate and preserve. The actual river matching process simply finds sets of possible paths in the 20k data then chooses the best path. This path is tested to ensure that it is an accurate depiction of the 50k river and does not break any flow relationships.



1.1.5 Hierarchy Following

Following the 50k hierarchy means that a tributary (child) of a river is not matched until its parent has been matched. The following diagrams present a hypothetical watershed and show the ordering of matching. The JuMP application will provide the user tools to quickly fix a parent stream that could not be automatically matched, and tools to subsequently match its children.

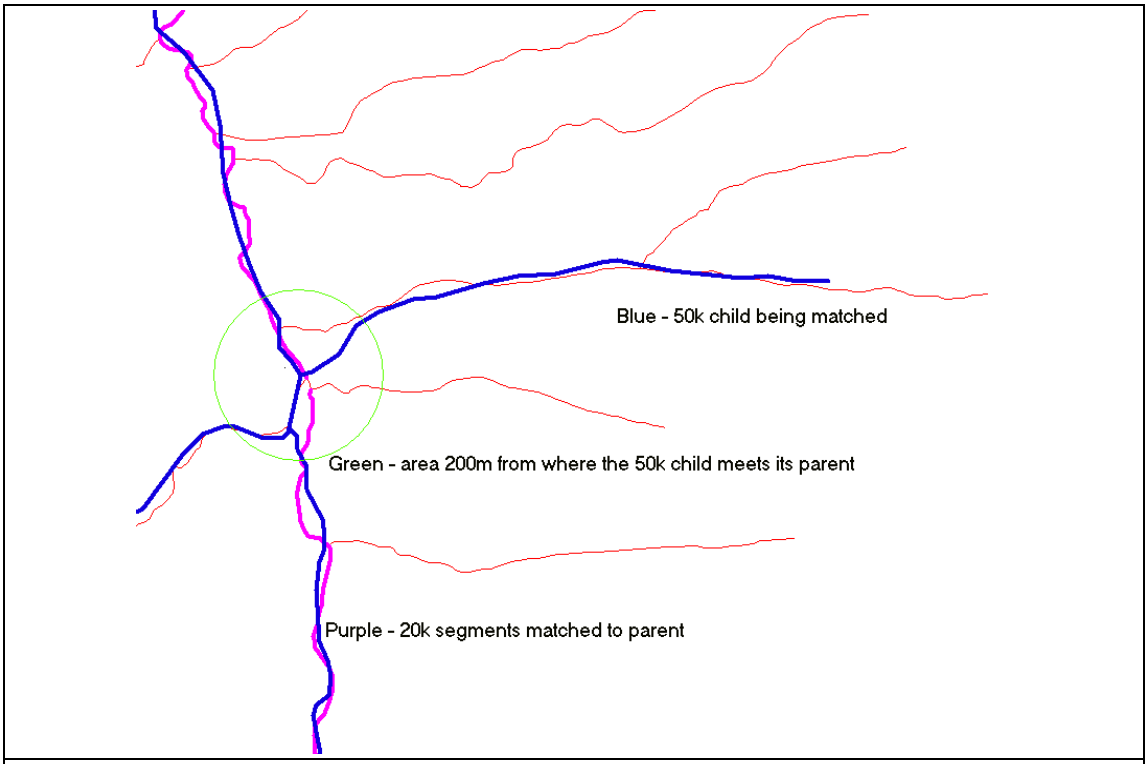





```
Attempt to match 110-987400-00000-000000(Stem A)
    If A matches, attempt 110-987400-72100-00000 (Stem B)
        If B matches, attempt 110-987400-72100-35700 (Stem C)
    If A matches, attempt 110-987400-81900-00000 (Stem D)
        If D matches, attempt  110-987400-81900-41900 (Stem E)
        If D matches, attempt 110-987400-81900-41900 (Stem F)
```

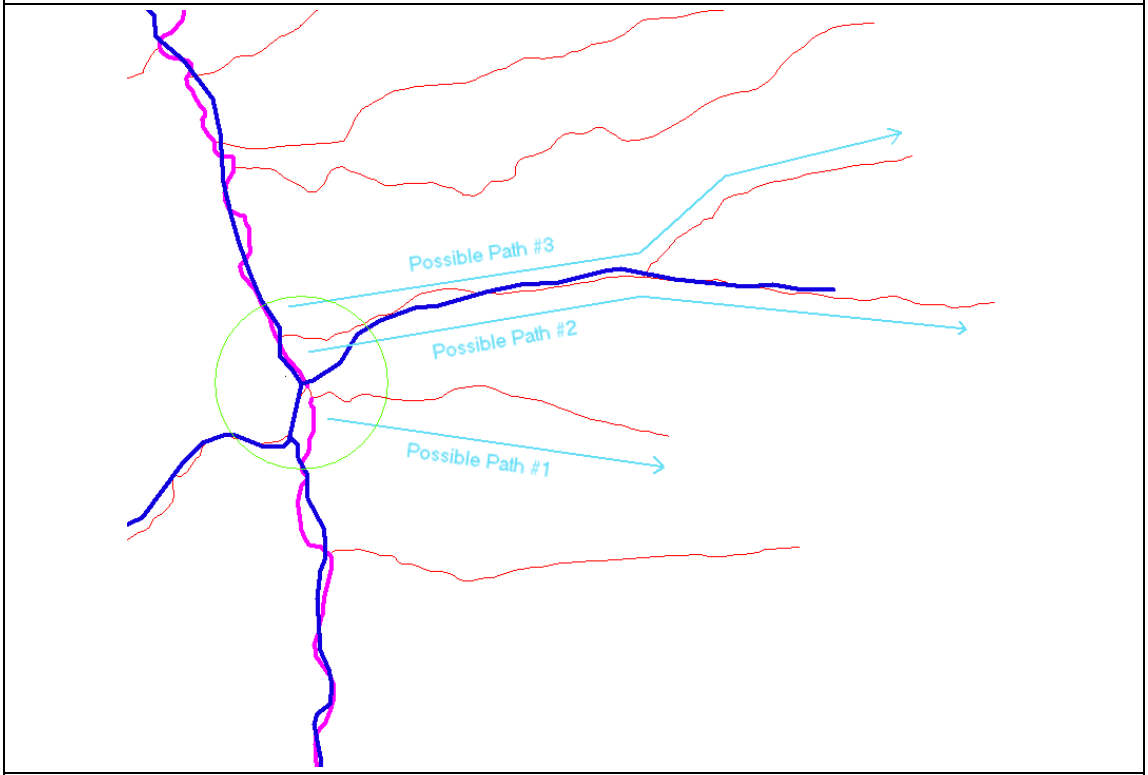
1.1.6 Path Formation, Walking, and Pruning

The basis of the matching process is to create a set of possible 20k matches, then choose the best path. The creation of these paths is quite simple – find 20k tributaries near the mouth of the 50k river, and start following the 20k rivers upstream. Incorrect tributaries will be quickly pruned away during the walking process.



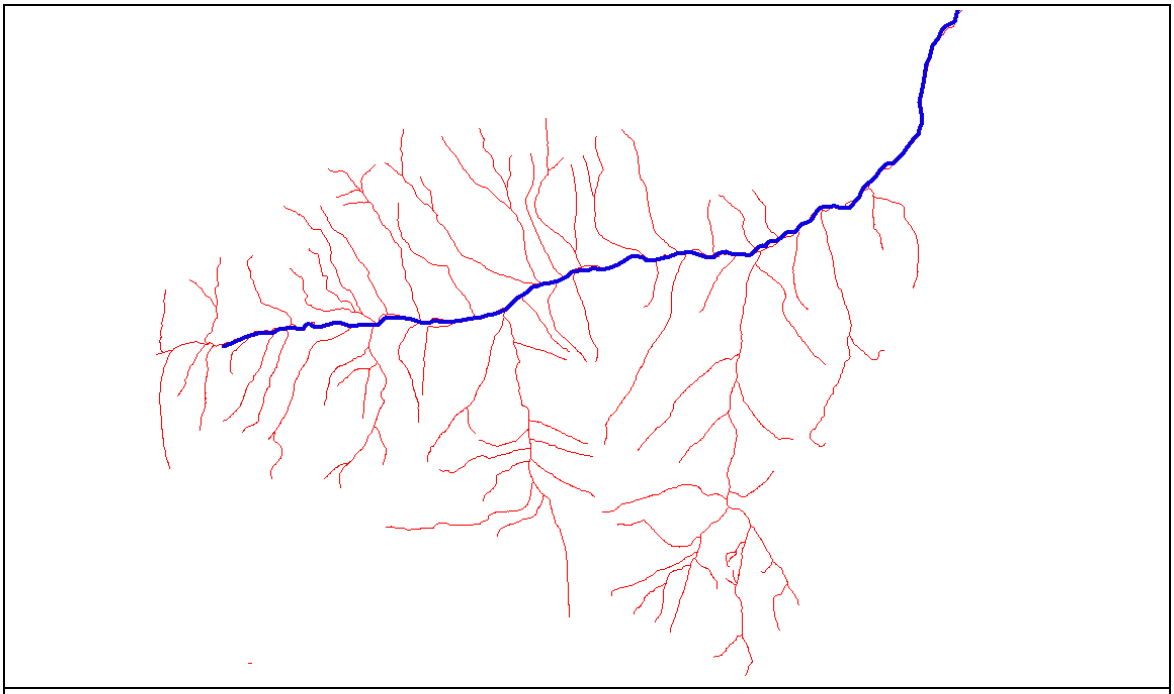
The parent river has been matched (20k match shown in purple) so its children are being matched now.

For the 50k tributary, the location of its mouth is determined, and all the 20k tributaries that are within 200m are found.

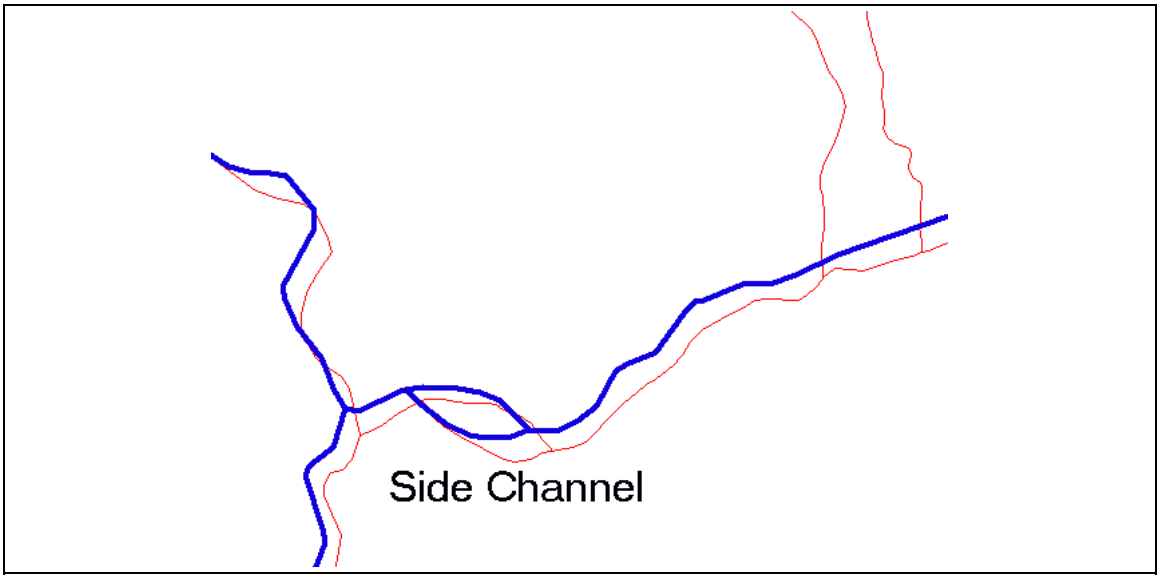


Starting from each of the start points found, paths are found by walking up each of the streams. The best of all the paths is found and then verified against rules for minimally acceptable matches.

Unfortunately, finding *all* possible paths can create too many paths to be evaluated – especially on large rivers. In order to be more efficient, a pruning process is used to quickly remove paths as soon as they diverge too much from the river they are trying to match.



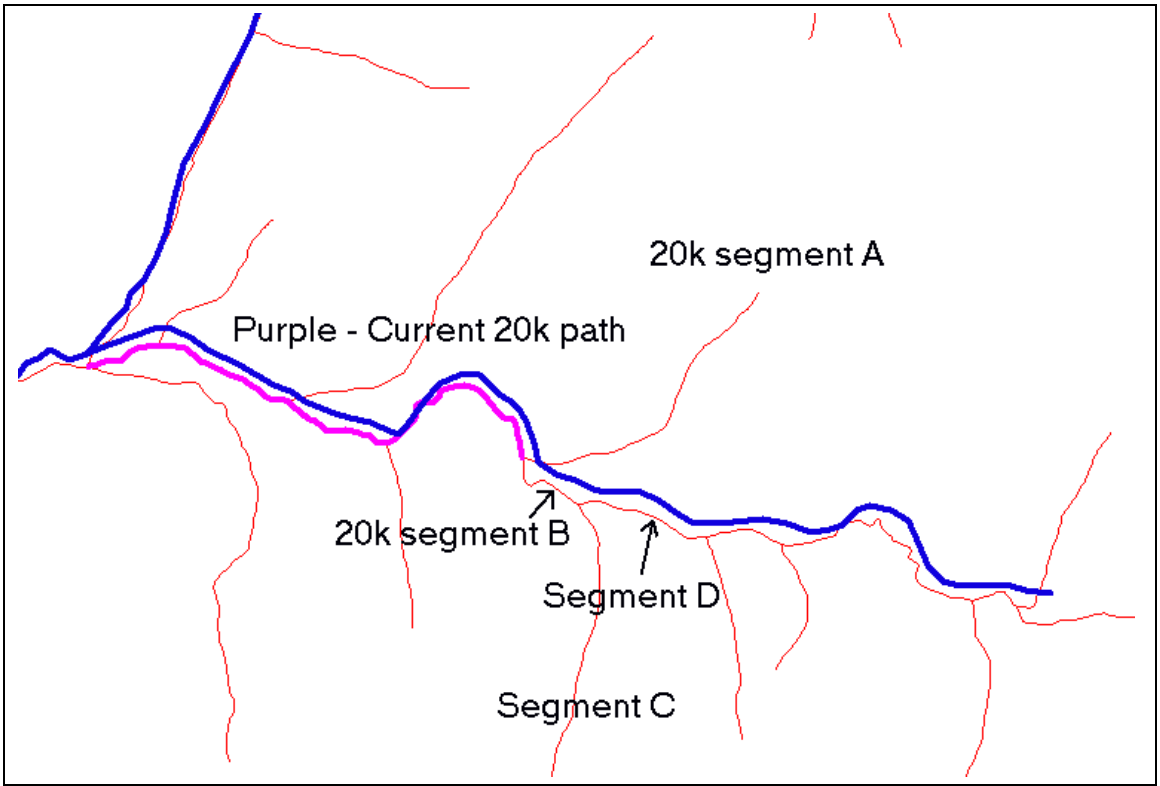
In this example, a naïve path finding approach would find over a hundred paths. On larger rivers, a full set would contain over 100,000 possibilities. In order to be more efficient, paths are pruned – removed from consideration – as soon as they diverge too much from the 50k river.



The type coding of rivers needs to obeyed too. In this example, the lower loop is a side channel. The main stem of a river is first matched – it will not attempt to find paths that include a "secondary flow" segment. If the match is successful, all the side channels will be matched (in longest-channel first order

to help with heavily braided rivers). During this match, only segments that are coded "secondary flow" will be considered.

The actual path formation is very simple – start at the initial segments and find the upstream segments. The path is built by adding this segment to the already existing path. If there was a junction in the river, simply make two copies of the path – on for each branch.



This shows the path building method. The purple represents an already existing path that is being built. The 50k data has not been adequately matched, so the path must be extended.

At this junction, we have two possible choices to follow – adding segment A or adding segment B (or creating two new paths).

If segment B meets an accuracy criterion, a new path will be created from the original by adding segment B. This extended path will continue to be investigated and segment C and/or segment D will be added in the next step.

Typically, several possible paths are produced. The best of all the possible paths is chosen. If it passes the match-is-good-enough criterions, the path is accepted.

Active stack – stack of all the paths currently being investigated
Match list – list of paths that might be matches to the 50k river

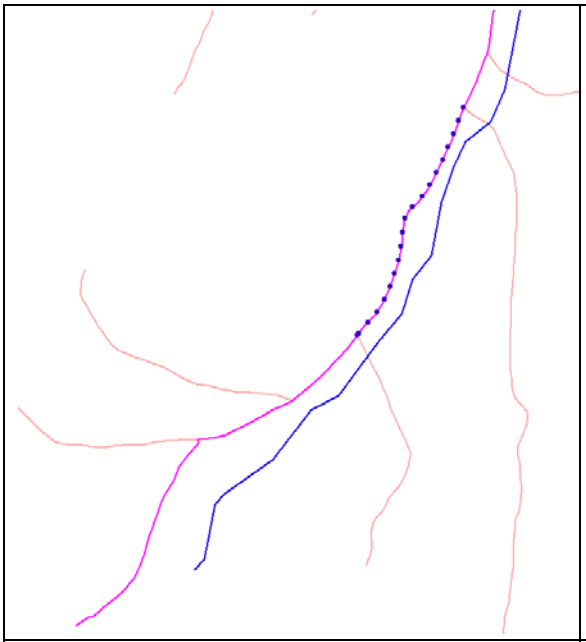
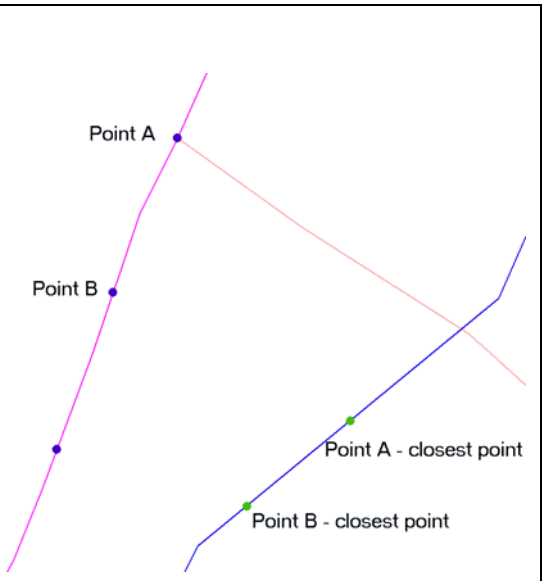
```
while (paths in active stack)
   pop a path from the stack
   find all the segments upstream from this segment
   if there are no more upstream segments,
       add this path to the match list
   for each of these upstream segments
      if this segment passes the criteria:
           * coded properly (i.e. Primary flow)
           * does not diverge too far from 50k river
```

```
        then add it to the current path and push the
            new path on the stack
```

1.1.7 Path Evaluation

The basis of providing information driving the matching process is determining how well a 20k segment matches a 50k river. A very simple sampling technique is used to make judgements about *nearness* and *shape*. The 20k segment is sampled every 25m along it length. For each of these sample points, the nearest corresponding point on the 50k river is found. Using this information, the nearest, furthest, and average distance between the two lines is determined and used to determine the line's *nearness*.

Shape similarity is determined at the same time as nearness by watching the movement of the corresponding nearest-distance matches on the 50k data. If the shapes are very similar, the corresponding 50k points should move the same distance (along the segment) as the sample points on the 20k segment. If the shapes are dissimilar, the corresponding distance on the 50k river will be either too small or too large. This principle is used to estimate similarity of the shapes.

<table>
<tr>
<td>  </td>
<td> 
</td>
</tr>
<tr>
<td>A 20k segment is sampled every 25m along its length. Each of these points is evaluated against the 50k river. The results of each of these point matches are summarized into a segment match.</td>
<td>This picture shows the details of the point matches made. The closest point, on the 50k river, to points A and B are found.
The distance, along the 20k data, between the two points is 25m. The distance between the corresponding points, on the 50k data, is also recorded.
In this case, the corresponding distance match is only 19m.</td>
</tr>
</table>

Once the set of paths has been made, and each segment in the path has been compared to the 50k data, the best path needs to be found. This is done by pair-wise comparing paths in a *commutative*[1] manner. This process involves looking at:

- How well the shape of the path matches the 50k river
- How close the path is to the 50k river
- How much of the 50k river is "close to" the 20k path

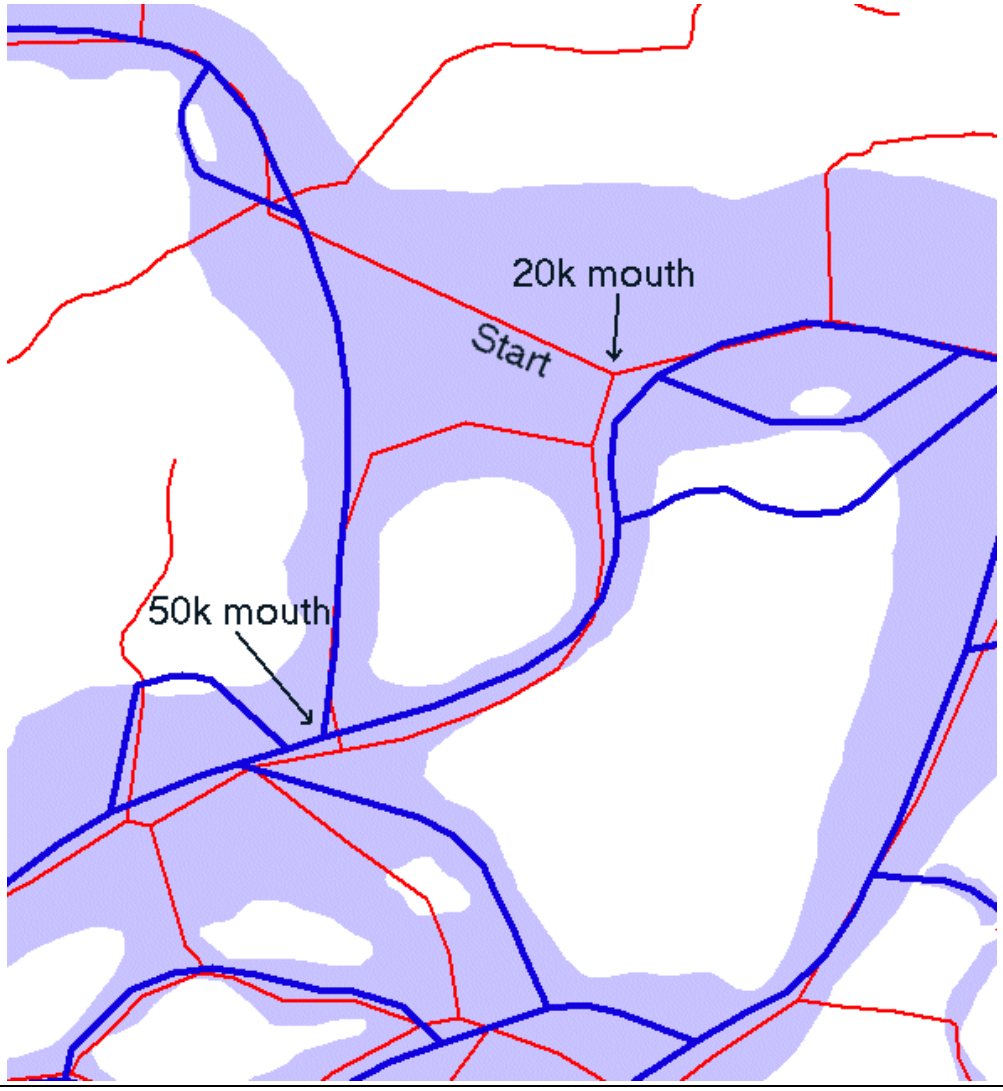
The best path may still not be a close enough match to consider a "true" match. The final match is evaluated to ensure that it really does follow the shape and location of the 50k river to a degree that is satisfactory.

1.1.8 Person Assisted Matching

The automated matching process provides an excellent initial matching of the 50k and 20k data. Unfortunately, not all matches can be automatically detected, and some matches may need to be verified as correct. In general, this type of manual process can be extremely time consuming. By providing the user with workflow control and powerful tools, the time required to do this can be dramatically reduced.

The underlying bases of the majority of these tools already exist in the algorithms used for the automated process. Three examples of how a user could quickly solve simple problems are given below:

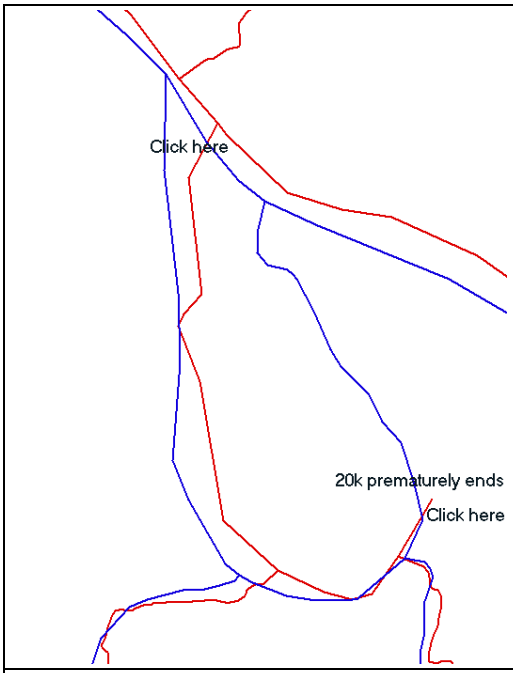
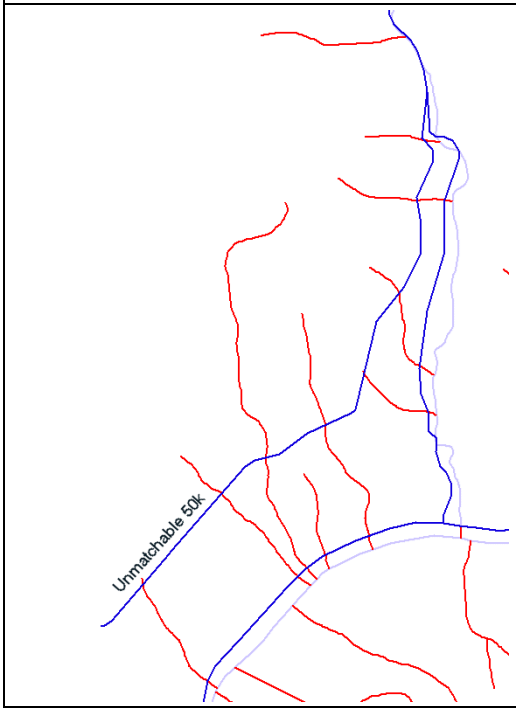
[1] Mathematically, commutative operations mean the order application is immaterial. In this context, the "better than" function means that if A "better than" B and B "better than" C then A "better than" C.



Due to differences in interpretation between the 20k and 50k data, the river mouths are about 1250m apart. The automated process rejected the match due to the discrepancy.

During the person-assisted process, this situation would be presented to the user. The user would simply start the automated matching process with the correct initial segment (labeled “Start”). Once the match is complete, the river’s children can also be automatically matched.

This problem is solved in a few simple mouse clicks.

	In this example, the 20k data prematurely ends (relative to the 50k interpretation). The user decides that this side channel should be matched. They automatically build a new route by click at the start and end of the desired path (the two "Click here" points in the picture). The match is made, evaluated, and logged. The children are then matched.
	This 50k river does not have a corresponding 20k equivalent. The automated process could not find a match and the user can very quickly verify that there really is none.

The workflow control and validation routines ensure that all problems are satisfactorily addressed, and the specialized tools allow each of the problems to be solved incredibly quickly. The application would record and log all the changes so that they could be re-applied if there are any changes to the underlying CSN 20k dataset.

The end result would be an application that a GIS professional could be trained to competency in as little as one day. This application would control the entire matching process, providing logs, presenting problems, facilitating correction, and determine the quality of each match.

1.2 Assumptions

The following assumptions are of key importance to the algorithm as currently developed:

- Both (20k and 50k) networks are topologically clean before matching begins.
- Both networks are can be built into upstream/downstream networks using digitization direction as a consistent guide.
- The attributed 50k network (BCWSA) includes a hierarchical watershed code used to determine parent-child relationships.
- Both networks have the BCWSA feature coding applied to their segments. These codes are used to distinguish between main flow and secondary flow matches. Differences in interpretation between the two datasets might cause problems to be investigated by the person-assisted application.
- Waterbody (i.e. marshes and lakes) will be matched by another process.

1.3 Issues and Risks

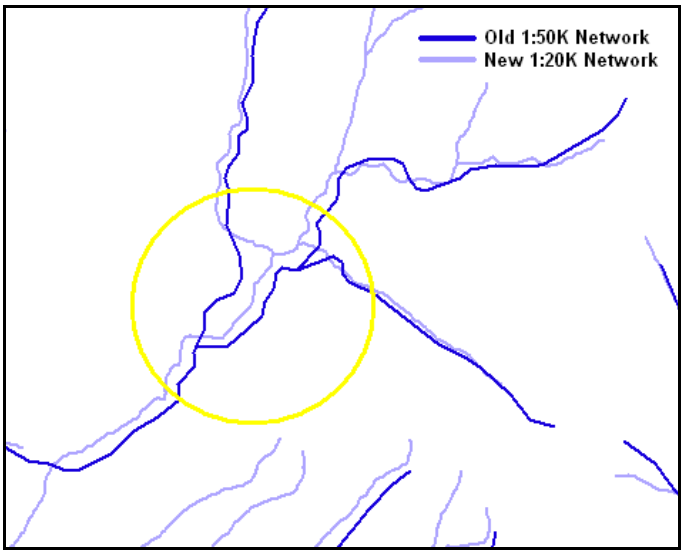
1.3.1 Issues

While the CSN and WSA data describe the same physical features, they were captured at different periods of time and (more importantly) from aerial photographs of substantially different resolutions.

This "multi-scale" aspect of the data leads to numerous disagreements between the two data sets. The calculation of non-physical portions of the network (e.g. skeletons) has also left room for interpretation, and as a result the networks frequently differ in their construction lines.

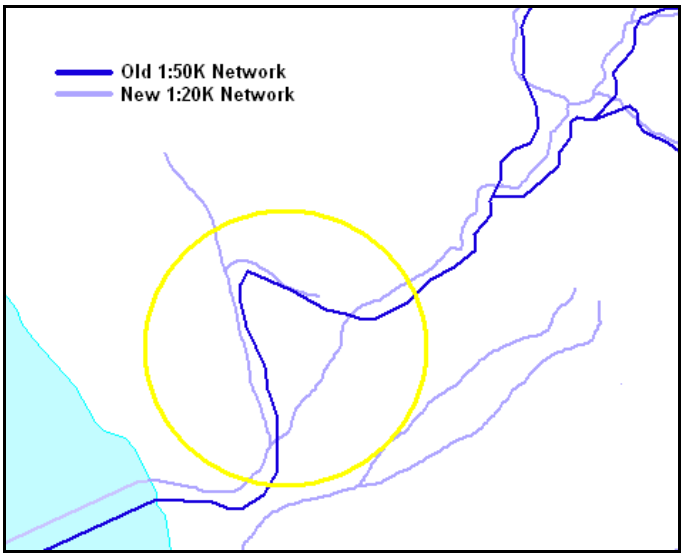
1.3.1.1 Confluence Interpretations Differ Radically

Where the interpretation of confluences is substantially different, network-walking algorithms can have difficulty determining possible candidate segments for matching.



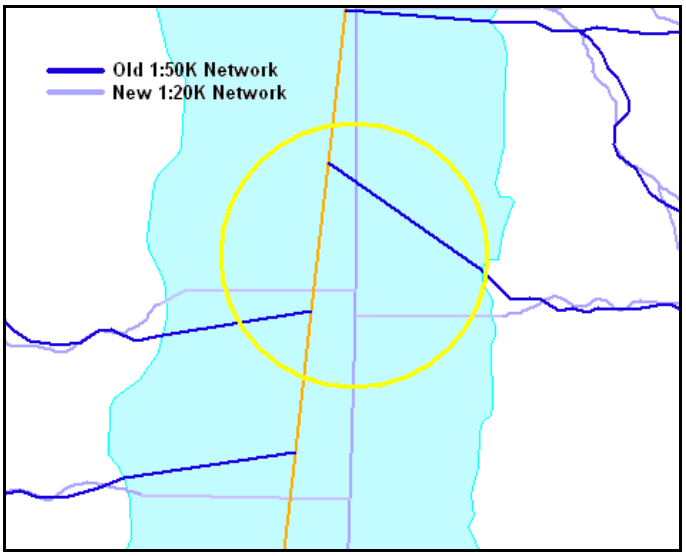
1.3.1.2 Topological Disagreement At Network Ends

At the upstream ends of the BCWSA network, where the photographic interpretation was most subjective, there are frequent differences of interpretation with the CSN data regarding direction of flow and/or stream channel location.



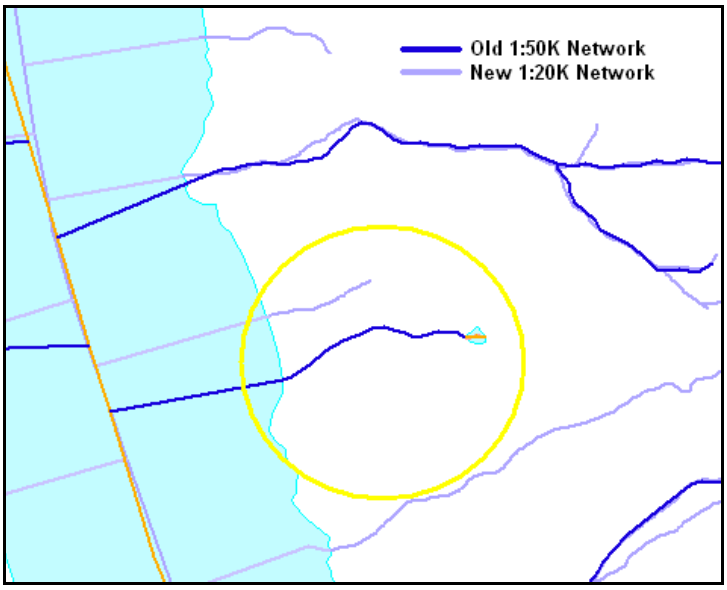
1.3.1.3 Topological Disagreement In Skeletons

When bringing together streams into a skeletonized network inside a waterbody, interpretation of where "connections" occur between streams becomes highly subjective. As a result, there are numerous cases of disagreement between the BCWSA and CSN skeletonization.



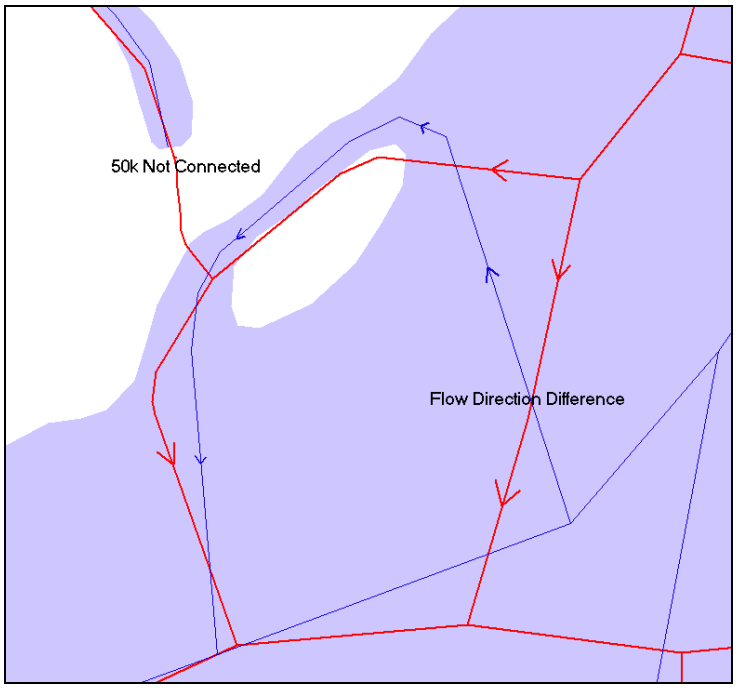
1.3.1.4 Unmatchable WSA Streams

Occasionally BCWSA streams at the upstream ends of the network veer out into areas where CSN has no streams at all. This is probably a case of photo interpretation error, where the 1:50K interpreter saw a "stream" in the photo which did not exist.



1.3.1.5 Flow Differences

The interpretation of flow in skeletonal structures is sometime ambiguous. This example shows a side channel in which the 20k and 50k data have geometrically similar solutions, but flow patterns are greatly different. The automatic matching process will not match these channels, and there is no *correct* solution.



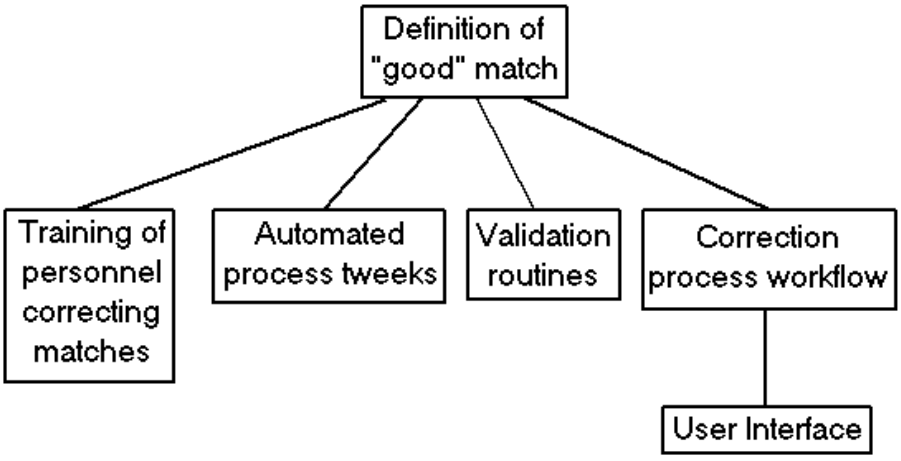
1.3.2 Risks

In general, the automatic process will correctly match the vast majority of the 50k rivers that do have a close 20k match. The rivers that are not matched are, in almost all cases, significantly different. The matching risks are almost exclusively involved in solving problems of this type.

The most important step is defining what constitutes a *good-enough* match and how to express this to the end-users. This is fundamental to the matching process – it determines what is matched, what is left unmatched, and how they are matched – especially in the "difficult" areas.

Once this definition is give, the automated process can be changed so unsatisfactory matches are not erroneously made. The validation routines that look for matching errors made by the automated process and the user corrections are also, in essence, driven by this definition.

The definition of a "good match" also feeds the training of the people will be using the JuMP application to correct and verify the matches that the produced by the automated process. It is essential that these people be given specific and detailed rules for matching so they can make consistent and repeatable changes. The risk is that this definition might not be specific enough to allow people to make consistent correction, or not intuitive enough for the end users to fully realize the issues of using their legacy data with the CSN product.



Another risk is that the automated process, has not been completely tested. Although it has been run on several watershed groups with excellent results, it has not been run on all possible types of watersheds. Its behavior in areas of highly complex canal systems has not been evaluated. No systematic quality testing or metrics has been performed on the automated matching process.

A third risk is that the UI (and workflow control) has not been developed. Ideally this requires designing a workflow that maximizes productivity and consistency while minimizing errors. A JuMP UI needs to be developed to guide the users through the process while providing powerful tools to make corrections. This UI and toolkits, based

on the existing automated process Java source code, could be difficult to develop and depends on a previous definition of what a *good* match represents. Since this interface has not been developed, there is no timing information available that would reveal how long the actual matching correction and validation process will take. The consistency between different users making corrections is also unknown.

A fourth risk is that the waterbody (lakes and wetlands) matching process has not been developed. Its accuracy and ease of integration with the stream matching process is likely to be highly synergetic, is unknown.

1.4 Recommendations

Based on the current state of development, evaluation, and testing of the automated and person-assisted matching application, the following recommendation are made:

- Creation of the definition and rules associated with a defining a *good* match. This must be verified as powerful enough to handle the majority of cases, and intuitive enough that the correctors can make consistent decisions quickly and easily. The end user community must also be consulted to ensure their legacy data can be easily transferred to the new 20k base.
- Continue testing, development, and evaluation of the automated process to ensure that it is appropriate for all the different they of watersheds. Extra processes need to be added so that complex networks can be matched with the minimum of user corrections.
- The workflow must be delineated and implemented so that the JuMP application UI and tools can be developed so they maximums the user's efficiency, accuracy, and consistency.
- Perform timing tests so that the cost of validation and correction can be estimated.
- Investigate the waterbody matching system and its integration with the stream matching system.
- Run the automated and person-assisted system on a few watershed and provide the results to end users so they can determine if the end product will actually allow them to transfer their legacy data.

NB: PAUL AND DAVE NEED TO PROVIDE ESTIMATES OF COST AND TIME FRAME FOR DEVELOPMENT OF A PROTYPE AND PRODUCTION ENVIROMENT