2.
PART II. ELEMENTS OF SPATIAL PLANNING: THEORY.

MERGING MAPS: NODE LABELING STRATEGIES

Sandra Lach Arlinghaus
The University of Michigan

Different inventories of roads produce different maps; two different maps can represent one set of actual roads. How is a planner to rectify this situation? To consider the problem abstractly, it is useful to characterize the road network as a graph, composed of nodes and edges joining nodes. Such style of consideration, as a graphical puzzle, dates from 1736 and the Konigsberg Bridge puzzle and from Hamilton's "Around the World" puzzle of 1859 (Harary, pp. 1, 4); as a planner's dilemma it has no doubt always been with us; and, as a cartographer's delight, it manifests itself over and over again in projection selection, a result of the one-point compactification theorem (visualized as stereographic projection) forcing us to recognize that the sphere can never be flattened out into the plane.

In Figure 1, two portions of road maps based loosely on the downtown Ann Arbor, Michigan, street map are shown superimposed—a conceptual view of a practical problem noted by John Nystuen, and implemented at the level of mapping by Andrea Frank and Jyothi Palathinkara (University of Michigan). As Nystuen et al. Noted (unpublished communication), these nodes are, in some sense at least, fairly "close"--one can recognize, visually, what the correspondence is to be from one map to the other. The viewpoint adopted here is that as graph-theoretic trees they are homeomorphic (have the same structure or connection pattern).

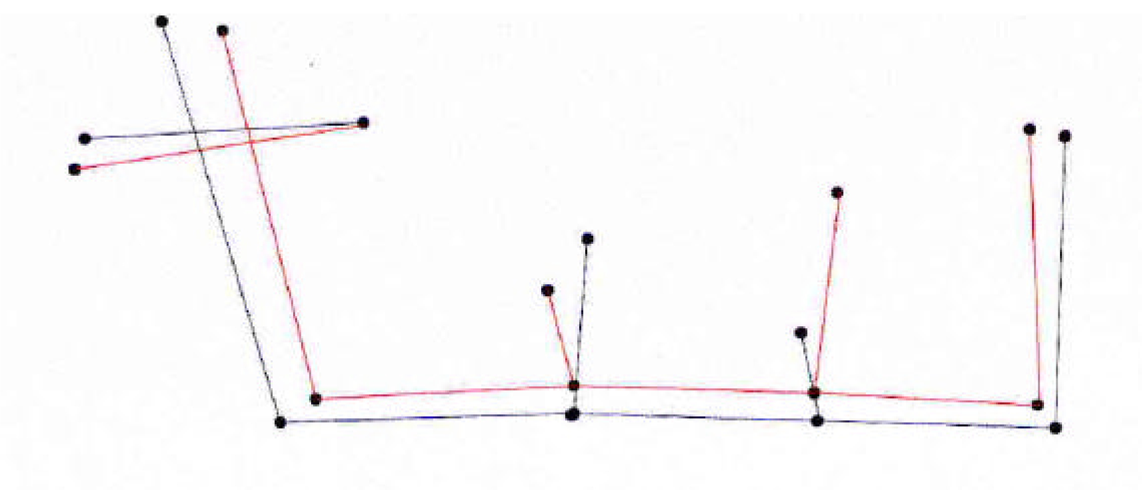

Figure 1. Two trees representing the same actual street pattern.

If the planners who use these maps rely on a node labeling scheme, in which both maps are labeled from the top down, beginning in the upper left hand corner, proceeding horizontally and then in a serpentine pattern from top to bottom, the first node encountered is labeled with the numeral 1; the second node encountered is labeled with the numeral 2, and so forth (Figure 2). However, the labeling imposed on these two maps is not identical, as one would wish (to indicate their identity in topological form). Indeed, the label 3 on the red tree then corresponds to the label 4 on the black tree and vice-versa, as do labels 5 and 6; a situation Nystuen et al. noted and were able to uncover on a number of occasions on actual maps (Figure 2). Naturally, as they note, one would wish, in a situation such as this one, to have the numerals correspond in an appropriate manner.

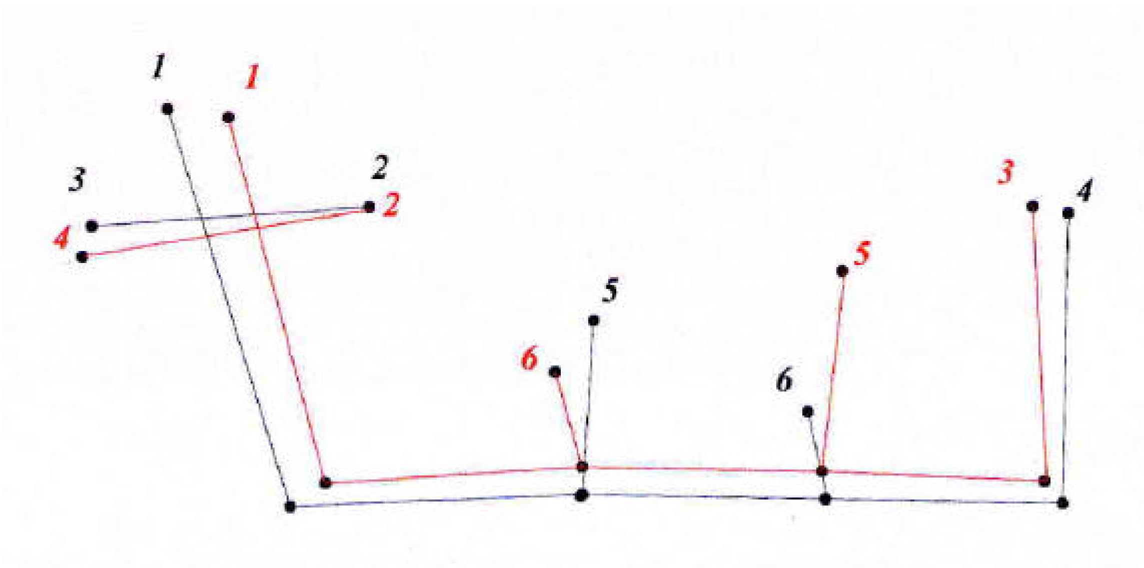

Figure 2. The red route and the black route: numbered nodes do not correspond as desired. The numbering scheme is a common one in which nodes are labeled from the top down in a left/right serpentine pattern (after Nystuen et al.).

Nystuen's observation came from a real-world example; it points, once again, to the appropriateness of considering different labeling schemes that will permit alignment of maps based on topology rather than solely on distance from the top of the screen. One strategy, that takes advantage of the structure of the underlying raster on which these maps were produced, proceeds as follows.

CANTOR PIXEL-LABELING ALGORITHM

1. Designate as a fundamental screen unit the smallest dot size to be used to represent a node; without loss of generality it can be assumed to be a "pixel."

2. In the manner that has come to be conventional with cathode ray tubes, use the top of the screen as the positive x-axis and the left side of the screen as the positive y-axis (with origin at the upper left-hand corner of the screen).

3. Assign to each pixel on the CRT an ordered pair of Cartesian coordinates: (1,2) is a pixel (an indivisible area) located one unit to the right of the left edge of the screen and two units down from the top.

4. Assign to each node in one map the pixel coordinates of the node as follows. If the pixel coordinates are (x,y), label the node with the rational number y/x.

This sort of strategy differs from what may be a computer default labeling scheme. Each possible node location has a unique label that is a single rational number indicating pixel location referred both to raster position and to location in relation to other pixels. The serpentine labeling scheme is still preserved even though its form is different, and perhaps, not evident. Cantor's enumeration technique shows that there is a one-to-one correspondence between the set of natural numbers and the set of ordered pairs of natural numbers. These two sets have the same cardinal number, aleph null (Hahn, pp. 1595-1596). Cantor's serpentine pattern through a square lattice converts the two-dimensional array into a one dimensional stream clearly equivalent to a number line (Figure 3).

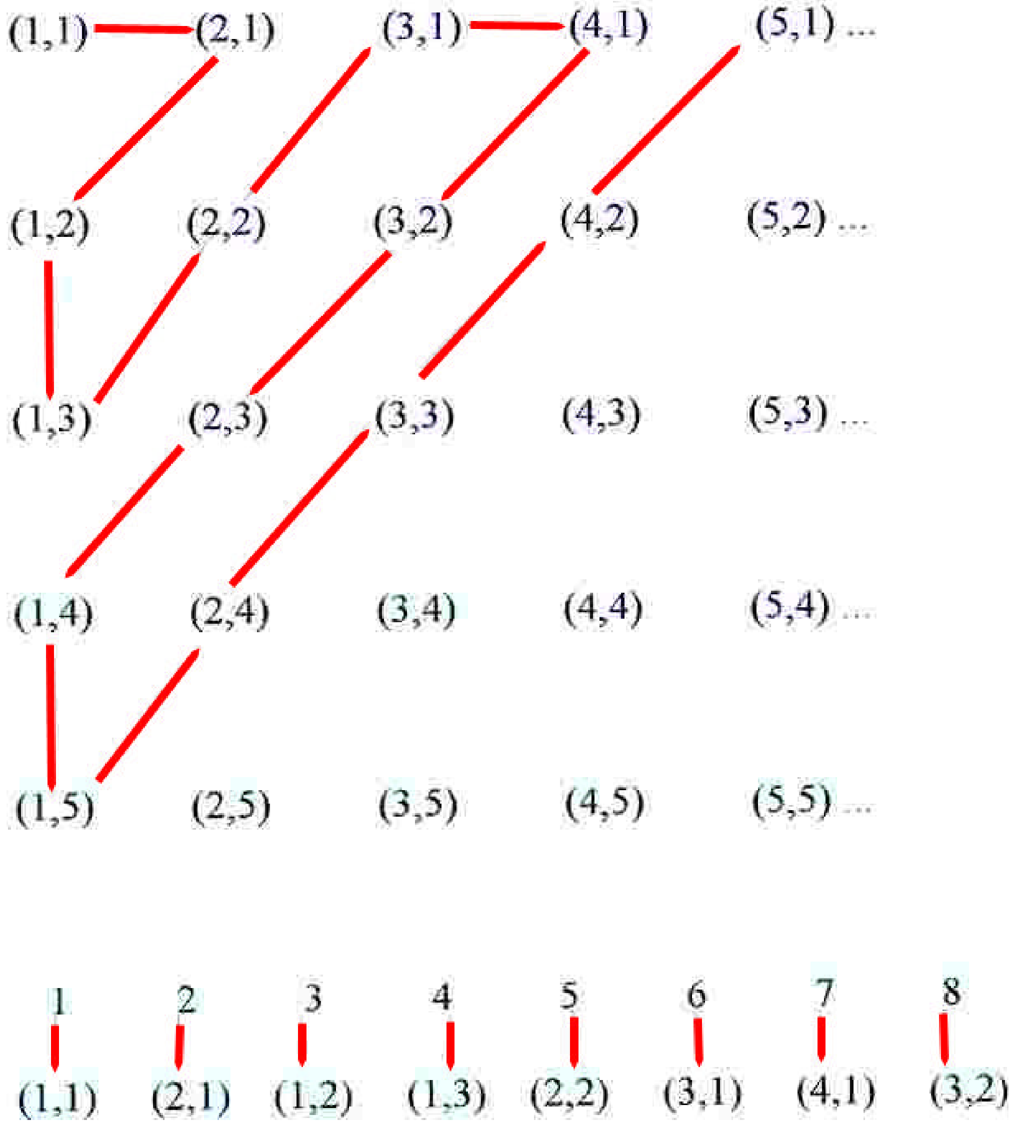

Figure 3. Assignment scheme for illustrating the cardinality of the set of rational numbers (after Hahn, p. 1595).

This strategy was the basis for Georg Cantor's (perhaps counter-intuitive) proof that the cardinal number of the set of rational numbers is aleph null. The set of ordered pairs contains the set of rationals--for, when ordered pairs are converted to rationals, as in the algorithm above, some are equivalent in value although not in visual form--1/2 and 2/4 are equivalent in value but do not look the same. However, the rational numbers contain the natural numbers as a subset. Thus, the cardinal number of the rationals is sandwiched between two sets each of whose cardinal number is aleph null--consequently, the set of rational numbers must also have cardinal number aleph null. Thus, the set of numbers y/x exactly covers the screen.

REPETITION OF THE CANTOR ALGORITHM

Suppose that the Cantor Algorithm has been applied once to a map: to the red tree in Figure 2, for example. Each node of the red tree will have a distinct, unique rational number label based on its position relative to the top and the side of the screen and in relation to other nodes. In a different layer, label the black street map in the same manner. It appears that at most one node on these two maps will have the same label; indeed, using rational number labeling, two nodes have the same label if and only if they are represented by the same pixel location in both maps. What is desired is to identify (match up or glue together) a node from one map with a node from the other map both of which represent the same physical location.

MERGING ALGORITHM

The following algorithm offers a strategy for making such an identification. It can be executed by buffering nodes, in an iterative fashion, and identifying two nodes at the first instance a node from one map falls into a node-buffer in another map.

1. Draw a buffer of one pixel width around each node in the red map. Assign identical labels to any pair of nodes on red and black maps both of which lie within or on the buffer.

2. Draw a buffer of one pixel width around each buffered node created in step 1. Assign identical labels to any pair of nodes on the red and black maps both of which lie within or on this newly drawn ring buffer region.

3. Repeat the process until one pair of buffer zones comes into contact with each other. Then stop the process.

If this Merging Algorithm provides a complete identification of nodes (with no extra nodes lying in the region outside the buffered zone), with identical connection patterns in both, then the two maps are said to be pairwise "well-matched." There is no difficulty in merging the two maps from different sources.

TRANSLATION OF THE MERGING ALGORITHM

Maps may fail to be pairwise well-matched for a variety of reasons. The following list suggests some reasons for such failure, but it may not be exhaustive.

1. The numbers of nodes in the red and black maps are different from each other.
2. The number of nodes in the two maps is the same, but the number of edges is different.
3. The number of nodes and edges is the same in both maps but the topology, the pattern of connection, is different.

4. Some or all of the three factors above may hold, in whole or in part, but the offset of pattern is so large that it falls outside the would-be buffer zone and gets visually confused with other pattern.

Nystuen notes in some of his examples that some parts of each map in the pair are shifted; the structure is recognizable from one to the other, but offset some significant amount. One way to deal with situations of these sorts is to permit translation, locally, of parts of the graph in one map. The Merging Algorithm will run in a satisfactory manner, up to a point. Beyond this point, nodes fall outside buffer zones. In such situations, use the diameter of the buffer zone (number of maximum number of pixels that came about in the use of the Merging Algorithm) as a translation scaling number. Add this number to each coordinate (in the local region) in one map. Now run the Merging Algorithm. Repeat the procedure as needed.

DIRECTIONS FOR FURTHER WORK

This issue of merging maps appears rich from an abstract viewpoint. Some of the other directions one might consider involve use of Hasse's Algorithm applied to graphs, development of error detection and correction criteria, use of sum-graphs or a similar strategy in which the topology of the graph guides the labeling pattern, and any of a host of theorems related to these topics.

Ann Arbor, MI
April, 1996.

REFERENCES

Hahn, Hans. "Infinity" in The World of Mathematics, James R. Newman (Ed.), New York: Simon and Shuster, 1956.

Harary, Frank. Graph Theory. Reading: Addison-Wesley, 1969.

Nystuen, John D. Personal communication, unpublished, 1996.