DEMONSTRATION OF IDEAS IN FULLY AUTOMATIC LINE MATCHING OF OVERLAPPING MAP DATA

Raymond J. Hintz
Mike Z. Zhao

Department of Surveying Engineering
120 Boardman Hall
University of Maine
Orono, Maine 04469
(207) 581-2189

ABSTRACT

The analytical matching of similar lines on a common map edge is a difficult process which has been addressed by several authors. The procedure is more complex when the information overlaps. Two examples fall in the latter situation. The first case is when different maps of a common area are being digitized into a GIS environment. A second example is digitized data in the overlap region between adjacent stereomodels in a strip of photos, or in the sidelap region between adjacent strips.

In many cases the stereoplotter operator visually controls the line matching procedure using on-line computer graphics, which contributes to an already tedious procedure. Final map "clean-up" often uses a tablet digitizer which communicates to the computer graphics system. This paper will detail prototype algorithms, which has been developed and tested in a PC environment, which can be resolve various matching problems without operator intervention. The matching procedure follows user-defined tolerance limits in its analysis, and provides error information in situations which cannot be resolved. Many of the same algorithms will be shown also as effective tools in "smoothing" the intersections of dissimilar lines.

INTRODUCTION

While it is a simple, though tedious, procedure for a human operator to match the digitized features in the overlapping region of two photogrammetric models or two adjacent map sheets, there are indeed some particular difficulties in implementing this procedure by a computer. A person has an innate ability to recognize geometric patterns and the trends which should be utilized in the matching process. While a computer is a proven tool in automatically solving large and complex numeric problems, identifying geometric structures and resolving them create a very difficult problem to computer software.

Several authors have worked on various aspects of edge matching problems [Shmutter et al., 1981], [Schenk, 1986], [Beard et al., 1988]. Discussions on the problems of matching topographic feature lines, adjusting elevations of model grids, and establishing boundaries between models were presented by Shmutter, et al. (1981). The difficulty of implementing edge matching with multiple features in a stereomodel overlap region has been discussed, and a generalized theoretical approach has been presented by Schenk (1986). Another approach for edge matching without overlap has been demonstrated by Beard, et al. (1988). The approach utilized in this study was designed for overlapping regions in addition to common edges.

The focus of this investigation is similar to the one studied by Schenk (1986). Many line features cross map or stereomodel boundaries: contour lines, roads, rivers, railways, etc. In this study it is assumed that a feature is assigned a line type and a CAM (Computer-Aided Mapping) attribute. The complex multi-feature matching problem needs to be broken into manageable small pieces so that the whole problem becomes a series of individual feature matching problems. A strongly structured programming style has been employed in solving the problem [Frank, 1987]. The goal of the project was to create a PC-environment line matching program which complements existing Kern analytical stereoplotter and mapping software. Turbo Pascal 4.0 was used as the programming language since it caters to highly modularized programming. The programs developed have been tested using both fictitious data and actual digitized data. In all situations, the developed algorithms can be easily adapted to any generalized edge matching problem.

THE NATURE OF THE MULTI-FEATURE MATCHING PROBLEM

The selected priority of line connections is outlined in Table 1.

Connection (Connector, Connectee) Priority:

1. Curved line to curved line, ends.
2. Curved line to straight line, side.
3. Straight line to straight line, ends.
4. Straight line to straight line, side.
5. Curved line to curved line, side.
6. Straight line to straight line, side.
7. Curved line to symbol (point).
8. Straight line to symbol (point).
9. Straight line to curved line.
10. Curved line to straight line, ends.

Table 1. The specifications the multi-feature matching problem.

At first glance, the requirements presented in Table 1 appear as only manipulations of the geometric relations of lines and points. It is actually the generalization of the aspects of geometry in the multi-feature matching problem. The effective classification and organization of the spatial relations of the data sets are the key to effective management of a large amount of digital information.

The specifications in Table 1 can be generally divided into two distinct categories. Certain situation occurs only in the overlapping region of two data sets. This situation is restricted to the same line types in both data sets as in the first and third rows in Table 1. The second situation is generalized line matching of any map information. This paper addresses the later situation as a stretch/peelback ("clean" line connection) problem, and it should not be mixed with the first line matching problem. The two problems are thus addressed separately.

While a line type is straight, curved, or a symbol, the CAM mode describes the line by color, thickness, and nature (dotted, dashed, etc). Essentially only one line matching algorithm has been developed as straight lines and symbols are treated as special types of curved lines.

GENERAL SCHEME OF THE PROGRAM STRUCTURES

Two programs have been developed to solve the identified problems of line matching in an overlapping region (MODTIE) and the stretch/peelback problem (PRETIE). PRETIE can be implemented before or after implementing the edge matching program MODTIE. Since both programs handle the same geometric entities, many of the same modularized routines can be used in each. Both programs use the same format for an input connection file. An example is listed in Table 2.

CONNECTOR	CONNECTEE	TOLERANCE	CONDITION
L2, C1	L2, C1	5	C
L2, C1	L2, C1	5	E
L2, C1	L2, C1	5	S
L5, C2	L5, C2	3	M
L1, C1	L1, C1	2	M

Table 2. Example of a connection file.

In Table 2 L# is the line type number and C# is the CAM mode number. This constitutes a unique identifier for an individual line type. The programs decide which connection needs to be implemented based on the connection condition. In these conditions, S means end to side, C is self-closed, and E is an end to end connection. These conditions are resolved in PRETIE. M indicates edge matching, and is thus resolved in MODTIE.

The mechanism of program MODTIE will be described in detail in the following sections. Since program PRETIE has many similarities, a discussion of it will be generalized.

While program PRETIE operates on data in a single input file, program MODTIE operates on two input data files and creates one output file. MODTIE needs to manipulate the data only within the overlapping region of the two data sets. In photogrammetric model joins this region is relatively small since at the usual 60% overlap between photos there is only a 10% overlap between the adjacent models. The data outside the overlap region does not need to be analyzed. The overlapping region, called a user-box, can be defined by the software automatically, or by user definition. Figure 1 represents the data separation idea, where Map1 and Map2 are the input data files, and Map3 is the output data file.

After the data in the overlapping region have been separated from the original data files, they are stored in temporary files. The software then utilizes individual line and CAM types for single feature matching in the fashion illustrated in Figure 2.

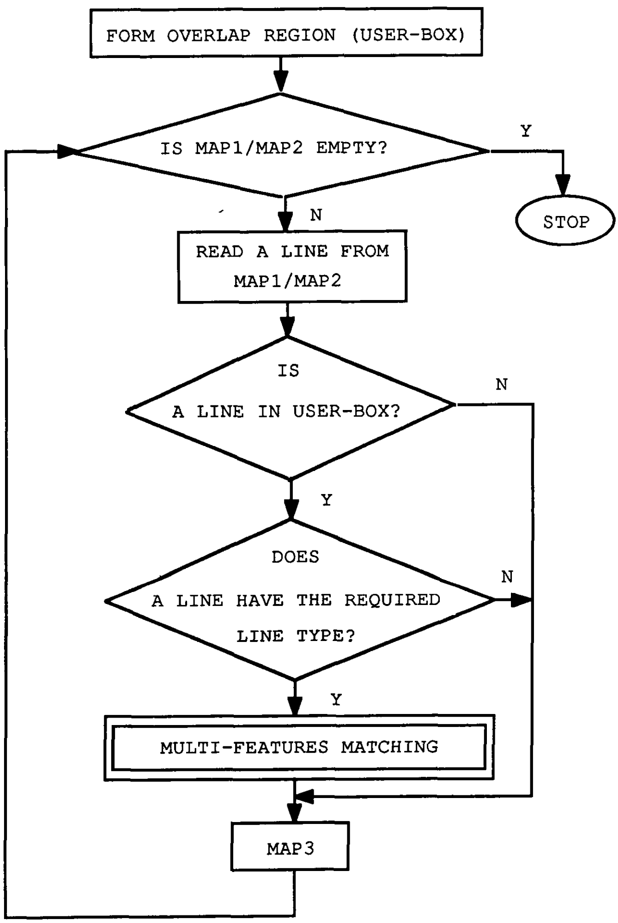

Figure 1. Data separation and analysis.

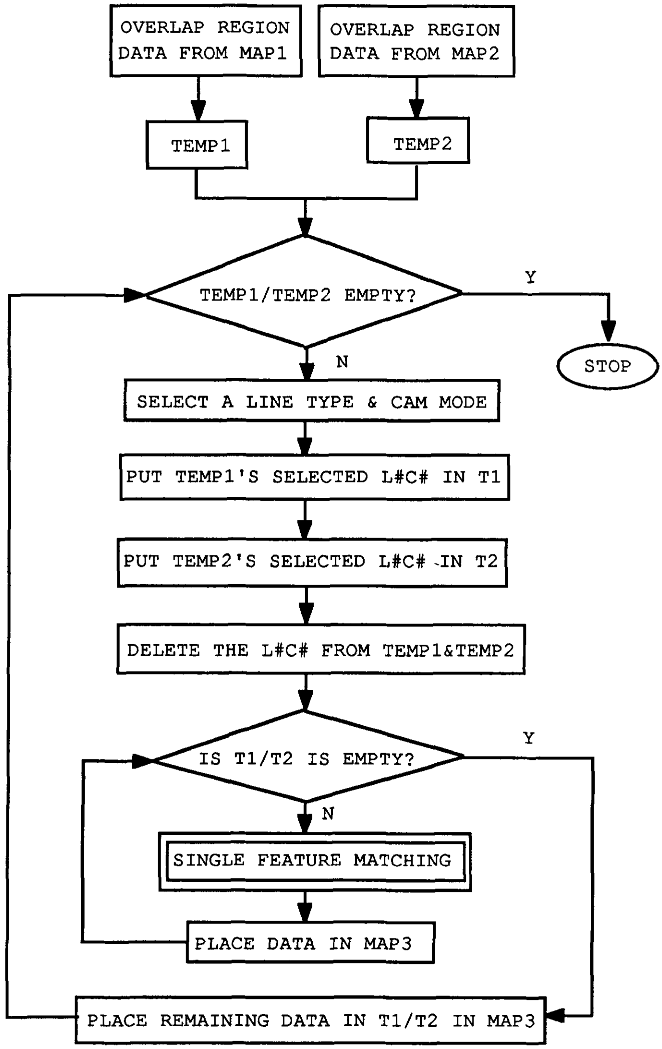

Figure 2. Break down the multi-feature problem into single feature problems.

The general structure of program PRETIE is illustrated in Figure 3.

IMPLEMENTATIONS

Identification of Conjugate Lines

The unique identifier L#C# (line type and CAM mode) allows software to concentrate on the same feature from two groups. Ideally, each line in one file, should have a corresponding line in the other file. If it is not true, there is a "dead end" to the line in the overlap region. Either this is a mistake or an actual occurrence. This situation is alerted to the operator.

To locate a pair of lines that appears suitable for connection, a search for an intersection of the two lines is conducted. Non-intersecting lines are connected if the minimum distance between them does not exceed the user-specified tolerance. A rectangle search region about the lines is created by the software (Figure 4). In addition to the distance checking, information such as elevation associated with contour lines is used to ensure a high percentage of correct identifications.

This method is successful if a user defines reasonable tolerance limits on line matches. These tolerances must be based on the quality of the data.

Based on the identification of conjugate lines, the connection between the lines is then conducted. A line can have multiple connections in the overlapping region (Figure 5). This situation further complicates the connection algorithm. To ensure a high percentage of successful connections, a general algorithm has been developed which efficiently continues to look for additional connections to the conjugate line.

A final problem which had to be resolved in connecting overlapping lines is which portion of a particular line needs to be discarded. All examples in figure 6 represent a curved line which needs to be connected, and it is obvious the connected line should pass through both the left and right ends of the overlap region. The direction of a line is arbitrary and the line connection algorithm has been developed to handle all four situations. The algorithm consists of two steps. The first step of the algorithm ensures that the two lines are merged along the original direction of the first line. The second step of the algorithm solves the ambiguity problem resulting from the four situations in Figure 6. The resolved ambiguity is represented in Figure 7 where Figure 7(a) results from 6(a) and 6(b), and 7(b) from the 6(c) and 6(d). The algorithm inverts the direction of the first line and conducts the line connection again. By comparing the length of the two connected lines, the longer one is selected as the final result.

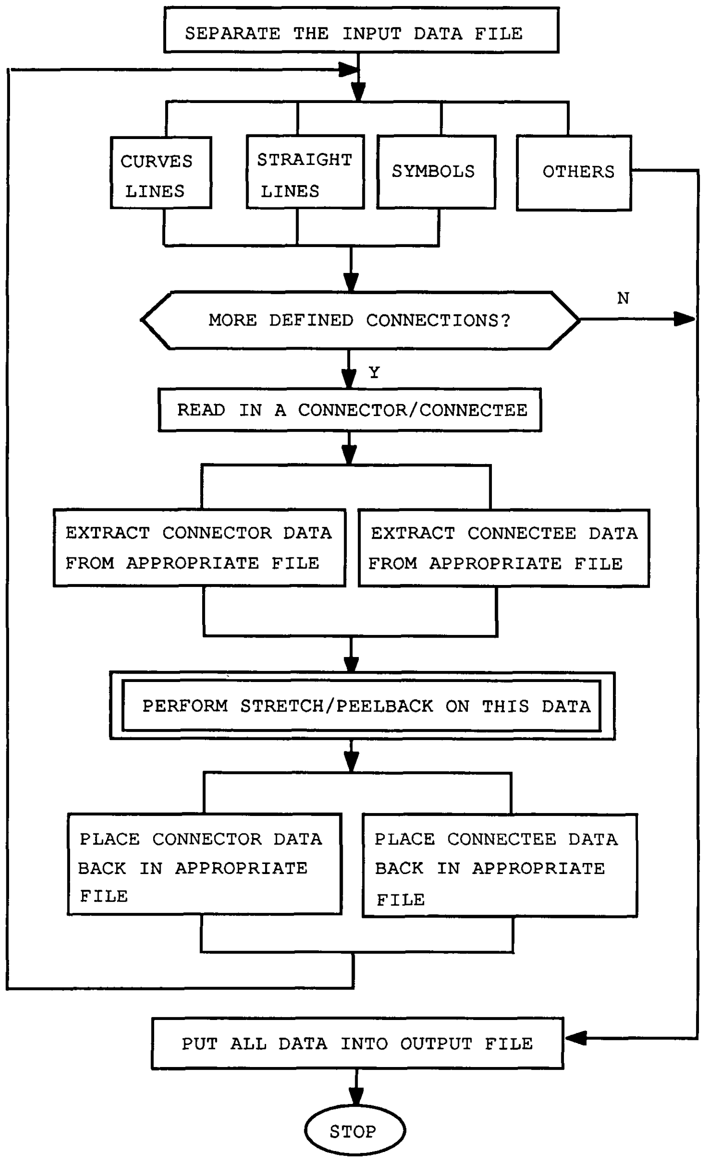

Figure 3. General flowchart of program PRETIE.

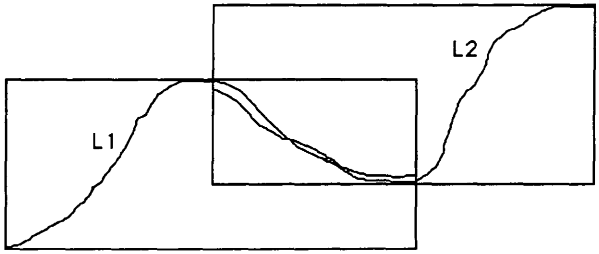

Figure 4. Rectangle search region for line match.

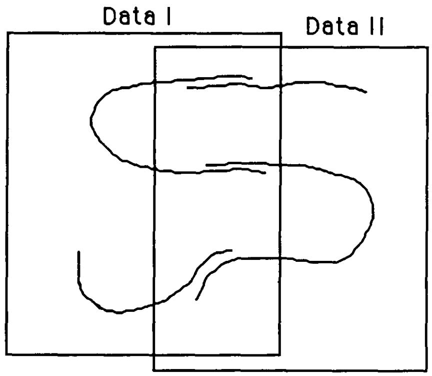

Figure 5. Multiple line matches.

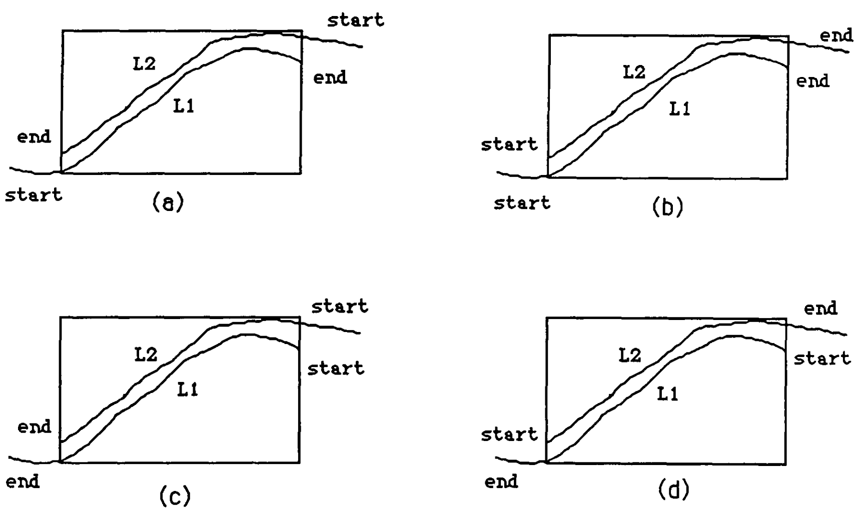

Figure 6. Four situations of the relative relationship between two conjugate lines.

Figure 7. Two possible results after connecting the lines of Figure 6.

Figure 8 demonstrates an example of matching digitized contours, where 8(a) and 8(b) are the results from two adjacent photogrammetric stereomodels, 8(c) is the combination of 8(a) and 8(b) without processing, and 8(d) is the merged results after running program MODTIE.

From Figure 8 it can be seen that there are some disconnections among the contours in the model. This is a special situation of stretch/peelback. The "smoothed" result using program PRETIE is illustrated in Figure 9.

CONCLUSIONS

A structured approach in solving the edge matching problem has been shown to be successful. The key in this approach is to break down a large and complex problem into many small ones that are manageable. With this approach a high percentage of line matching and stretch/peelback connections can be fully automated.

ACKNOWLEDGEMENTS

The authors wish to thank Kern Instruments, Inc., of Brewster, New York for their support in this research.

REFERENCES

Beard, K.M., 1988, A Localized Approach to Edgematching, The America Cartographer, Vol. 15, No. 2, pp. 163-172.

Frank, A.U., 1987, Geometry and Computer Graphics, Class Notes, Dept. of Surveying Engineering, University of Maine, Orono, Maine.

Schenk, T., 1986, A Robust Solution to the line-Matching Problem in Photogrammetry and Cartography, Photogrammetric Engineering and Remote Sensing, Vol. 52, No. 11, pp. 1779-1784.

Shmutter, B., and Y. Doytsher, 1981, Matching Between Photogrammetric Models, The Canadian Surveyor, Vol. 35, No. 2, pp. 109-119.

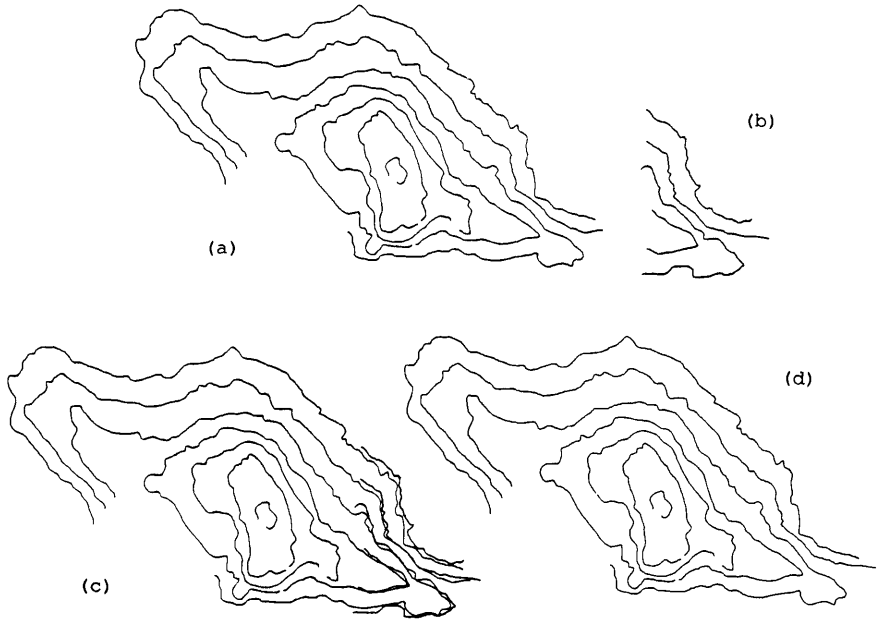

Figure 8. Example of edgematching of overlapping photogrammetric digitized data.

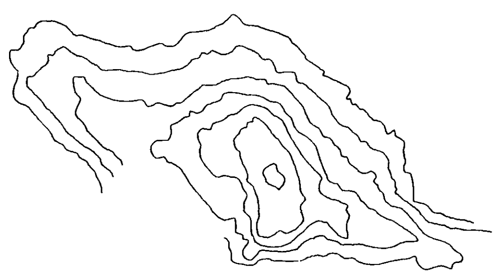

Figure 9. Example of data "smoothing" of digitized photogrammetric data.