# A Robust Solution to the Line-Matching Problem in Photogrammetry and Cartography

#### Tony Schenk

Department of Geodetic Science and Surveying, The Ohio State University, Columbus, OH 43210-1247

ABSTRACT: In the overlapping area of adjacent photogrammetric models features are normally digitized in both models and are, therefore, subject to averaging. Though this process does not act as a significant difficulty when performed manually by the operator, it poses interesting problems when data are stored digitally and a computer solution is preferred. In this case, the problem consists in determining the overlapping area, identifying and averaging conjugate (corresponding) lines, and replacing the original by the averaged lines. The solution proposed in this paper is based on discrete geometry and digital image processing methods, hence resembling the draftsman's problem-solving ability much more closely than methods which are based on Euclidean geometry. After extensive testing on a McIntosh personal computer, the software prototype has been implemented in a Kern MAPS200/300 digitizing and editing system.

#### INTRODUCTION

**S** INCE THE SPREAD of computers through universities and corporations in the late 1950s, photogrammetrists have been increasingly attracted by numerical methods. Today it is not very difficult to adjust, rigorously and simultaneously, hundreds or even thousands of photogrammetric models. The computation of digital height models is another example demonstrating the relative ease with which we treat problems involving a large number of observations, by manipulating them according to statistical methods such as, for example, the leastsquares method.

More recently, the computer has been used to perform tasks previously left to an operator or draftsman; in photogrammetry, for manuscript plotting and editing, and also in cartography. In many respects, these applications differ a great deal from the problems referred to above. For one thing, the statistical treatment and analysis of data is fairly well known and understood; once appropriate algorithms have been developed, the task is reduced to a mere data processing problem. The second class of problems does not behave in the same way, and, surprisingly enough, tasks solved with the greatest of ease by a draftsman often defy computer methods. This problem area is more complex than it might seem at first glance, and the temptation is strong, if not irresistible, to apply the same methodology which has proved so successful for solving adjustment problems. This paper describes a departure from our traditional methods which proves to be superior in the case of line-matching two adjacent photogrammetric models.

In the compilation of a map involving several

photogrammetric models, line-matching almost inevitably becomes a problem, because data in the overlapping area of two adjacent models differ due to unavoidable errors related to the data acquisition process. A topographic feature, such as a contour line, is normally digitized as a string of points and recorded either by equal time or distance intervals, or by a combination of the two. It is, therefore, very unlikely that all the points recorded in the two models are identical.

Contour interpolation based on heights measured in some manner, along profiles or by means of a grid, lead to the same problem, assuming that interpolation is performed independently in a model or parts of a model. The contours will not match exactly along the dividing line even if the neighboring sections overlap. Line-matching can be used to overcome this defect.

If we assume that several models have been independently digitized and data are stored in a common ground-control coordinate system, then the task of line-matching may be divided into determining the overlapping area, identifying and averaging conjugate (corresponding) lines, and, finally, replacing the original by the averaged lines. This paper describes all phases in this process, but emphasis is placed on identifying and averaging conjugate lines, because this presents a major challenge when approached by conventional methods. Confronted with the problem of identifying and averaging conjugate lines between neighboring models, one is easily tempted to solve the problem by applying Euclidean geometry; because the lines are given as a sequence of points, it seems most obvious to compare distances between points on different lines,

PHOTOGRAMMETRIC ENGINEERING AND REMOTE SENSING, Vol. 52, No. 11, November 1986, pp. 1779–1784.

look for the shortest distance in each case, and average these distances. An examination of Figure 1a. quickly reveals the pitfalls in this approach; the shortest distance between two points on different lines is no guarantee that the two conjugate lines have been found. Even the more sophisticated approach of computing the distance of points projected onto the line segments may lead us badly astray, as may be seen in Figure 1b.

Why do we try so hard to solve this problem by Euclidean geometry when a draftsman solves it with great ease? A human being "sees" a line in its entirety, consisting of all the intermediate points, materialized as small patches. The mathematical abstraction with the two end points is meaningless to him. If, for example, the best matching line has to be found for a given line, a draftsman's decision is always based on the full information of the lines or, we may say, on the solid representation as distinct from the abstract representation. Because this approach is far more efficient, it has been adopted for solving the line-matching problem. Consequently, the lines are represented by raster elements rather than by their end points, and discrete geometry is preferred to Euclidean geometry. A classic approach to the solution of the line-matching problem, based on analytical geometry, has been proposed by Shmutter and Doytsher (1982).

## DETERMINATION OF THE OVERLAPPING AREA

If photogrammetric models or map sheets are digitized independently, then the first step in solving the line-matching problem is to determine the overlapping area, in order to consider only relevant data in the subsequent steps. This is a fairly straightforward task and does not impose any serious problems. Initially, perimeters of both models are found by consecutively comparing minimum and maximum coordinate values. Next, the two perimeters are intersected and the resulting polygon is the perimeter of the overlapping area as depicted in Figure 2. The exact shape of the perimeter, however, is not of importance; it suffices to replace it by a surrounding rectangle. The approximation of the overlapping area by a rectangle allows for simple clipping algorithms to be applied. In other words, the overlapping area is regarded as a window against which all features of both models are clipped, using standard methods of interactive computer graphics as described, for example, in Foley and Van Dam (1982). Note that true clipping is not necessary; in fact, there are advantages to be gained by including in the linematching process the entire line segment crossing the boundary of the overlapping area. Hence, crossing line segments need not be cut off by clipping.

### IDENTIFYING CONJUGATE LINES

There are many features crossing model boundaries: contour lines, roads, rivers, and railways, to mention just a few. The features are graphically rep-



FIG. 1. (a) Distances between points on line 1 and line 2 are longer by comparison with those between lines 1 and 3, even though line 2 is closer to line 1. (b) Projection of line 2 onto line 1 results in greater distances by comparison with line 3, although line 2 is closer to line 1.



Fig. 2. Overlapping area between two adjacent models.

resented by lines of various types and forms: solid lines, dashed lines, heavy lines, parallel lines, etc. Regardless of the particular representation, these lines are generally stored as a sequence of points accompanied by a pen-down pen-up code. For the remainder of this paper, only this generic line representation is considered, even though different line types (e.g., parallel lines) and forms (e.g., a line pattern of dash-point-dash) may considerably add to the complexity of the problem.

As stated, the part of the line digitized in one model is independent of its continuation in the next

model. The task of line-matching is to link the parts together as if the line were digitized continuously, without interruption at the model boundary. When we consider not only the digitizing errors but also the systematic errors, which tend to increase toward the edge of a model, there is good reason to digitize lines, especially those representing natural features, so that they overlap. Thus, the operator is not forced to continue precisely where he left off in the previous model. It is even better to disregard the digitized portion completely, in order to avoid that the operator "interprets" measurements to conform to the digitized line from the previous model.

Where does this leave us in our attempt to identify and average conjugate lines? At the outsset, when the preliminary procedure has been completed, we have no idea what lines are conjugate; in other words, which of them belong to each other and, in fact, represent the same feature. Focusing on one line of the first model, all lines in the second model that are of the same feature class are eligible as conjugate lines and must somehow be compared with that in the first model.

In order to limit the search process, a top-down approach is used by which, in a first coarse step, the lines of the second model are reduced to a few candidate lines, again applying a method of interactive computer graphics. As shown in Figure 3, a rectangle enclosing the line, commonly referred to as extent or bounding box, is computed for all the lines. Candidate lines are lines whose extents overlap with that of the line in the first model (line 1). Comparison of these extents can be programmed extremely efficiently using clipping techniques.

The next step in our top-down approach requires a more datailed analysis, in order to determine which of the candidate lines is closest to the line of the first model and is, in fact, the conjugate line. Because an operator solves this problem so easily, a computer method that mimics the operator's problem-solving abilities seems particularly promising. As stated, the operator perceives the line in its entirety, not as an abstract representation in the form of the two end points. We therefore leave the realm of analytical geometry and computer graphics; instead, we turn to discrete geometry and rely on image-processing methods. Figure 4 demonstrates the contrast between the analytical and discrete representation of lines. The latter allows a quick decision on which line is closer to line 1 (Figure 4b). The analytical representation of the same lines almost resembles a picture puzzle when one is faced with making this decision (Figure 4a).

In discrete geometry, a line is represented as raster elements, sometimes also called pixels, though this term should be used in connection with images rather than with line drawings. The transition from an analytical to a discrete representation is commonly referred to as rastering and is, in fact, a form of discretization. In our problem, the lines under consideration are rastered using Bresenham's algorithm as described in Foley and van Dam (1982). Figure 5 shows the result of this intermediate step. The raster elements of a line simply form a string of x and y integer values. Note that rastering leads to a loss of accuracy. That is, the original coordinates of the end points of the line's vector representation cannot be accurately recovered, on account of the rounding effect of rastering. Clearly, the error increases proportionately to the size of the raster elements.

Even when the raster elements for all the lines are available, we still need to find the conjugate line. Our only guide is the determination of identical raster elements: the candidate line with the most identical raster elements to the line of the first model is the conjugate line. As may be seen in Figure 5, line 2 has no identical element, line 3 has two, and line 4 has one. In accordance with the rule as stated, line 3 should be the conjugate line. This is obviously wrong and necessitates a refinement of the rule: the







FIG. 4. (a) Analytical representation of lines by two points and a pen command. (b) Discrete representation of lines.



FIG. 5. Rastering of lines 1 to 4.

number of identical raster elements must be reasonably large so that the decision can be based on statistical methods. But how do we increase the number of identical raster elements?

Two different approaches have been developed and tested, depending on the class to which a line may belong. Lines can be grouped into two classes: a first with relatively long vectors, and a second with short vectors, where a short vector has a length comparable to or even smaller than the average distance that the two conjugate lines are apart from each other. Stated differently, assuming that the size of a raster element is a fraction of the shortest vector, say one-fifth or thereabouts, then there may well be no identical raster elements except where the lines intersect. This problem can easily be overcome when the thickness of the line is incresed, a process also called painting or brushing the line. Every raster element is considered as the center of a 3 by 3 mask that is continually dragged along the line, element by element, resulting in more raster elements representing the line. Clearly, the same effect cannot be obtained by starting with a largersize raster element. After thickening the lines we obviously find more identical raster elements, and the process can be repeated iteratively, using increasing mask sizes, until a suitable stopping criterion can be satisfied (see Figure 6).

Where the vectors of the original lines are considerably longer than the distance between the lines, a top-down approach has proved more efficient in terms of computer time, even though the abovementioned process can also be applied. In this case, we begin with raster elements that are as large as one-third of the shortest vector. Because, in this class of lines, the shortest vector is still quite long, the first iteration will yield many identical raster elements for every line (see Figure 7). In order to make the conjugate line converge, the raster size must



FIG. 6. A 3 by 3 mask dragged along a line.



FIG. 7. Large-size raster elements yield more identical elements (dark quadrangles) between two conjugate lines.

subsequently be decreased until a stopping criterion is reached. In every iteration, the number of identical raster elements is multiplied by a weighting that is reciprocal to the raster size.

We conclude that a determination of conjugate lines involves rastering of lines with various raster sizes, and a comparison of the raster elements.

Rastering is a common real-time process used in many computer graphics systems. Hence, very efficient algorithms are available. The process of comparing raster elements is a search process for which various fast algorithms are available. Not surprisingly, the procedure described in this paragraph can be programmed extremely efficiently in terms of throughput time. However, line-matching need not be implemented in an interactive environment where speed is of utmost importance; a batch mode is quite adequate, and operator interaction is required only at the end to control and to solve interactively all the cases the program rejected or failed to solve properly.

#### AVERAGING CONJUGATE LINES

The procedure described above results in the positive and unambigous determination of the conjugate lines, i.e., in identifying the two lines that represent the same feature in the overlapping area. We are now left with the task of finding an average line which will replace the two conjugate lines. Again, we propose to apply a method that mimics the problem-solving abilities of a draftsman when confronted with this problem. The draftsman's solution probably comes close to finding the skeleton of the area defined by the two conjugate lines. In image processing, determining a skeleton is referred to as thinning and is a well-known task; *cf.* Pavlidis (1982). Before a thinning algorithm can be applied, the conjugate lines have to be completed to form a closed polygon, as shown in Figure 8.

The problem is to find two pairs of suitable points that, when connected by a straight line, close the polygon. This might seem fairly straightforward at first glance, but a closer look at Figure 8 shows that it is not. The procedure is quite elaborate and deserves mention. The program developed will eliminate spikes as they may occur at the beginning or at the end of digitizing a line. During the process of determining the conjugate lines, the program computes and stores the location of best match, vividly demonstrated in Figure 8. Hence, the last point of line 1 and the first of line 2, which both belong to the part of best match, are known and have a valid claim to be considered as starting points for the closing vector (see Figure 8). The end points of the two vectors are chosen in such a way that a certain portion of line 1 and 2 is included in the area to be thinned, depending on the distance between digitized points.

Thus far, the perimeter of the area has been established. In order to bring the thinning algorithm into effect, the area ought to be filled with raster elements. Once again, filling touches on some interesting problems. It starts with a raster element, the seed that is computed near the end point of one of the closing vectors.

The thinning algorithm imitates a parallel process and begins by examining all the raster elements along the boundary of the area. The raster elements not belonging to the skeleton are then removed. Thinning is an iterative process; the area is peeled off in every iteration until its width is reduced to a single raster element. The remaining raster elements form the skeleton, shown in Figure 9.

## VECTORIZATION OF THE SKELETON

The lines are originally digitized as a sequence of points; they are stored in vector format. But the skeleton, that is, the average of the two conjugate lines, is in a raster format. Thus, the remaining task in our line-matching problem is to transform the skeleton from a raster to vector representation. The objective is to approximate the string of raster elements by a sequence of straight lines or vectors, but preserve the character of the original lines. This is a rather vague statement indeed that can be further defined qualitatively, i.e., the average length of the vectors to be found should be comparable to those of the original lines. The same also holds true for the shortest to the longest vector. Further, changes of direction between consecutive vectors should be similar. These rules are just a few out of all the aspects that a skilled draftsman takes into account with great ease in solving this problem manually.

A special algorithm has been developed to match the rules mentioned above. The solution reflects a typical bottom-up approach, starting from the microscopic world of individual raster elements and ending in the macroscopic world of straight lines defined by two points. For every raster element, the position of the two adjacent elements is computed with respect to the N-neighbor relationship, N =(0,1...8) (cf. Figure 10). The difference between two N-neighbors should be four for a straight line, but only in the eight possible directions as shown in Figure 10. All other straight lines have a different pattern, as may, for example, be seen by examining Figure 7 more closely. This effect must be attributed to the process of discretization: in analytical geometry the number of different directions is unlimited, but in discrete geometry it is severely limited because there are only eight possible directions between two adjacent raster elements. The more raster elements there are between the two points in a line, the greater will be the number of distinct directions.

In order to determine a straight line, we have to take into account N-neighbor differences other than four. A closer look at the result of rasterization reveals that Bresenham's algorithm never produces any other N-differences except three, four, or five. We therefore conclude that, whenever an N-difference differs from three, four, or five, the three raster elements concerned do not belong to a straight line. Rigorously applied, this rule leads to relatively short vectors. In the less than ideal world in which we live, we find it expedient to slacken this rule and accept N-differences of two or even one. This, of course, leads us into generalization, because the more deviations from the stringent rule we are willing to accept, the coarser will be the approximation and the greater the degree of generalization. Equally important is the question of how many N-differences in succession other than four are tolerable. For example, an N-difference of three should be followed by one of five, in order to bring the raster elements into a straight line again. If, for example, a difference of three is followed by another of three or one



FIG. 8. Closed area between two conjugate lines.



FIG. 9. Skeleton of area shown in Figure 8.



Fig. 10. Eight-neighbor code.

of four, this must be considered as a definite bend in the line and thus as the end point of the straight line. Ultimately, the algorithm merely checks the *N*differences of successive raster elements. If a difference of three is not compensated by one of five (or vice verse), the algorithm concludes that it indicates the end of a straight line.

#### CONCLUSIONS

The problems in photogrammetry and cartography suitable for solving by computers may be arbitrarly divided into two categories. The first includes problems that can be formulated more or less easily by mathematical models, on which the development of suitable algorithms can be based to take full advantage of the computer's data-processing capability, which is far superior to that of a human being. For the second category, the opposite holds true, because it concerns problems to which conventional methods that may have proved successful in the first category can be applied only with difficulty or not at all. Line-matching is a typical example of the second category; it belongs to a type of problem normally solved by a draftsman, whose problemsolving ability is often far superior to computer methods.

Because conventional methods often fail, new methods have to be found. Methods that resemble the draftsman's problem-solving abilities are likely to be a good choice, because he solves this type of problem with great ease. Suitable methods can be borrowed from discrete geometry and digital image processing or from the higher realm of artificial intelligence. Unfortunately, these methods are still unexplored territory to many photogrammetrists and cartographers. But obviously, we are merely at the beginning of a new, exciting, and challenging era in which the computer will increasingly be used for non-numerical tasks.

The line-matching problem is more amenable to methods of discrete geometry than to those of Euclidean geometry, because the representation of lines as a string of raster elements strongly resembles the draftsman's perception of a line, namely, as an entirety. Therefore, fairly straightforward algorithms are sufficient for an elegant and rigorous solution of the line-matching problem. The author has developed a prototype solution in Pascal on the Mc-Intosh personal computer, which also gives satisfactory results as far as throughput and computer resources are concerned. The prototype has been implemented into the Kern MAPS200/300 digitizing and editing system.

The solution presented in this paper is suitable for comparing lines consisting of many vectors with each other, leaving the computer to answer questions such as which line offers the best fit to another line. The training of operators offers an interesting application: a line, for example a contour line digitized and stored by an instructor, can be compared with the trainee's line; the numerical result obtained is a useful criterion of how closely the two lines match.

## ACKNOWLEDGMENTS

The work described in this paper has been supported by Kern Instruments, Inc., Brewster, N.Y., under a consultancy contract. The author expresses his thanks to Hans Wehrli and Louis Cogan who have contributed to the solution of many details and to Arthur Lindsay for checking the manuscript.

#### REFERENCES

- Foley, J.D. and A. Van Dam, 1982. Fundamentals of Interactive Computer Graphics. Addison-Wesley Publishing Company.
- Pavlidis, T., 1982. Algorithms for Computer Graphics and Image Processing. Springer Verlag.
- Shmutter, B. and Y. Doytsher, 1981. Matching Between Photogrammetric Models. *The Canadian Surveyor*, Vol. 35, No. 2, pp. 109–119.

(Received 29 January 1986; accepted 3 April 1986; revised 3 July 1986)