Apictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition

H. FREEMAN, SENIOR MEMBER, IEEE, AND L. GARDER, STUDENT MEMBER

Summary—**This paper describes the development of a procedure that enables a digital computer to solve "apictorial" jigsaw puzzles, *i.e.*, puzzles in which all pieces are uniformly gray and the only available information is the shape of the pieces. The problem was selected because it provided an excellent vehicle to develop computer techniques for manipulation of arbitrary geometric patterns, for pattern identification, and for game solving. The kinds of puzzles and their properties are discussed in detail. Methods are described for characterizing and classifying piece contours, for selecting and ordering pieces that are "most likely" to mate with a given piece, for determining likelihood of fit, for overcoming ambiguities, and for evaluation of the progressive puzzle assembly. An illustration of an actual computer solution of a puzzle is given.**

Introduction

The familiar two-dimensional jigsaw puzzle is a problem in pattern recognition that has long served as a pastime for both young and old. It is a one-person game in which one is given a picture cut into many irregularly shaped pieces; the objective of the game is to fit the pieces together to reconstruct the original picture. With some practice, a person can develop techniques which reduce the time required to assemble a complete puzzle. Some of these techniques are expressible and some are intuitive; in some way all involve the application of concepts of pattern recognition to both the shapes of the pieces and the pictorial information on them.

As is the case with many puzzles,[1,2] unexpected difficulties are encountered when one attempts to program a digital computer to solve jigsaw puzzles. The difficulties relate to three different aspects of the problem: 1) the description of the pieces (encoding and classifying), 2) the manipulation of the pieces (rotating and matching), and 3) the evaluation of "correct fit." The difficulties arise primarily from the language limitations of the computer; however, the restriction to coarse quantization (imposed by considerations of practicality) introduces additional difficulties. The greatest source of trouble is the tendency for the dimensionality of the problem to "run away." A computer scheme (algorithm) is of value only if the time and storage requirements increase only modestly with increases in the complexity of the puzzle. A "brute-force" approach, which may work for small puzzles, very rapidly becomes futile as the number of pieces is increased.

If the pieces of a jigsaw puzzle are turned over so that no picture information is available, *i.e.*, so that all pieces are uniformly gray, assembly of the puzzle becomes considerably more difficult. Techniques for solution must now rely entirely on *shape*. Puzzles of this type will be referred to as *apictorial* puzzles. A procedure for solving such puzzles by means of a digital computer is the subject of this paper.

Since the apictorial jigsaw puzzle represents a well-defined and easily understood problem, and since it aptly illustrates the difficulties besetting all pattern recognition problems, it is ideally suited as a guinea pig for testing new pattern recognition techniques. The analogy between solving jigsaw puzzles and solving problems in map matching or photointerpretation is readily apparent.

Characteristics of Jigsaw Puzzles

The objective of any jigsaw puzzle is the arrangement of a set of given pieces into a single, well-fitting structure, with no gaps left between adjacent pieces. Under the cover of this rather general statement lies a large variety of different kinds of puzzles.

The distinction between pictorial and apictorial puzzles has already been drawn. The following characteristics may be used to identify additional kinds of puzzles.

Orientation

Jigsaw puzzle pieces are usually given without information about their orientation in the assembled puzzle. In a so-called *oriented* puzzle, however, each piece has a clearly marked direction with respect to the other pieces. Assembly of the puzzle thus requires no rotation of pieces. A puzzle based on a geographic map (say, the boundaries of the individual states of the U.S.A.) where north is indicated on each piece is an example of an oriented puzzle.

Connectedness

An assembled puzzle may cover a simply-connected area, or it may have "holes" in it and thus be multiply-connected [Fig. 1(a)]. Alternately, a given set of pieces

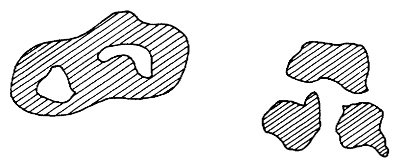
Manuscript received September 30, 1963; revised December 27, 1963. This work was part of a research study in pattern analysis supported by the Air Force Office of Scientific Research, Information Sciences Directorate, under grants AF AFOSR 62-152 and AF-AFSOR 24-63.

The authors are with the Dept. of Electrical Engineering, New York University, N. Y.

[1] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. and Rev.*, vol. 3, pp. 211–219; July, 1959.

[2] M. Minsky, "Steps toward artificial intelligence," Proc. IRE, vol. 49, pp. 8–30; January, 1961.

may assemble into two or more disjoint areas, in which case the set of pieces is simply a mixture of two or more puzzles [Fig. 1(b)]. The multiply-connected case also covers the situation in which one or more pieces of a puzzle are missing. Similarly, the disjoint-area case covers the situation where one or more extraneous pieces are present.



(a) (b)

Fig. 1—Multiply-connected and disjoint-area puzzles.

Exterior Boundary

For some jigsaw puzzles the exterior boundary is known to be rectangular; the length and width may or may not be known. In others the boundary may be irregular and either known or unknown. A puzzle formed by selecting only a portion of a given puzzle will nearly always have an irregular and unknown exterior boundary.

Uniqueness

Most commercially available puzzles are unique; *i.e.*, the pieces can be assembled properly in only one way.[3] One can, however, easily visualize the design of a puzzle which is not unique, in that more than one configuration of interior or exterior boundaries can be realized. A puzzle may be nonunique even though its exterior boundary is fixed. One notes also that there is no direct relation between puzzle uniqueness and piece uniqueness; *i.e.*, a puzzle may have some pieces occur in duplicate and still be unique, and vice versa.

Radiality

Radiality refers to the kinds of interior and exterior junctions in the assembled puzzle. A *triradial* junction is the junction of three boundary lines. A *quadradial* junction joins four boundary lines and a *quintradial* junction joins five (Fig. 2). The popular puzzle configuration of Fig. 3(a) has triradial junctions on its exterior boundary and quadradial junctions in its interior. The puzzle of Fig. 3(b) has only triradial junctions; that of Fig. 3(c) has only quadradial junctions. (A map of the state boundaries of the U.S.A. excluding Hawaii and Alaska shows only one quadradial junction; all other junctions are triradial.) Junctions of radiality greater than four are rare, though it is, of course, not difficult to construct puzzles containing them. The puzzle of Fig. 3(d) has nine triradial junctions as well as one of radiality nine.

A small (nine-piece) but nevertheless complicated jigsaw puzzle is illustrated in Fig. 4. The puzzle is assumed to be unoriented. It is simply connected and has an irregular exterior boundary. It is also unique, although this is not obvious from casual inspection. The puzzle has eight triradial junctions on its exterior boundary; in its interior it has three triradial junctions, one quadradial junction, and one quintradial junction. One observes that this puzzle, as well as any other, is *completely characterized by knowledge of all piece boundaries plus the mating relationships between the boundaries.* The piece boundaries are given; the mating relationships are to be determined.

The mating relationships of a particular jigsaw puzzle can be represented in simple form by means of a *mating diagram.* This diagram is topologically the *dual* of the puzzle. Each puzzle piece is replaced by a *node*, and mating boundaries between pieces become *branches* connecting the corresponding nodes. One notes that no branches may cross each other. An extra node[4] is provided for the "exterior"; branches to this node correspond to the exterior boundary contributions from each piece. The mating diagram for the puzzle of Fig. 4 is shown in Fig. 5. (The significance of the Roman numerals in Fig. 5 will be pointed out later.) One notes that in the mating diagram the radiality of each of the puzzle's junctions is represented by the branches of a mesh window; *e.g.*, the nodes 6, 2, 3, 4 and 5 in Fig. 5 are part of a five-branch mesh window, indicating that the corresponding pieces in Fig. 4 join at a quintradial junction.

At every interior n-radial junction ($n \geq 3$) of a jigsaw puzzle, the boundaries of at least $n-2$ of the pieces must possess a slope discontinuity. Thus at least one piece has a slope discontinuity in its boundary at a triradial junction, and at least two have slope discontinuities at a quadradial junction. Of course, slope discontinuities may also occur at points other than junctions. However, the knowledge that there must exist a minimum of one slope discontinuity at every interior junction is of great importance for the solution of apictorial jigsaw puzzles. (See Fig. 6.)

It follows from the foregoing that, in a given puzzle, a large number (if not all) of the pieces must possess one or more slope discontinuities. (Exceptions to this occur in special cases. See, for example, the two puzzles of Fig. 7.) Since the extent of the mating between two pieces is defined by two junctions, it is instructive to distinguish between different *types of matings* on the basis of the presence or absence of slope discontinuities at the junctions. The six possible types of matings are shown in Fig. 8. In Type I mating, both pieces have slope discontinuities at both junctions. In Type II both pieces have slope discontinuities at one junction, but only one has a discontinuity at the other junction. In Type III only one piece has a slope discontinuity at the two junctions. Types IV, V and VI are similar except

[3] This assumes that the possibility of disjoint areas is excluded.

[4] One such node must be provided for each distinct area which is not covered by a puzzle piece.

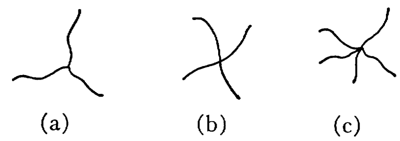


Fig. 2—Triradial, quadradial and quintradial junctions.

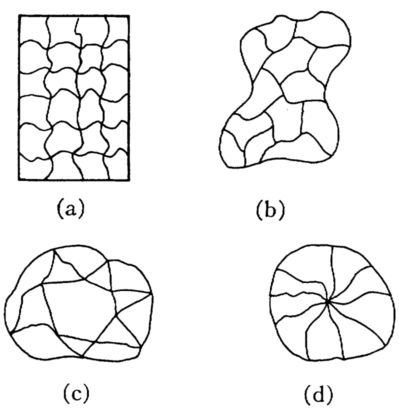


Fig. 3—Illustration of junction radiality in puzzles.

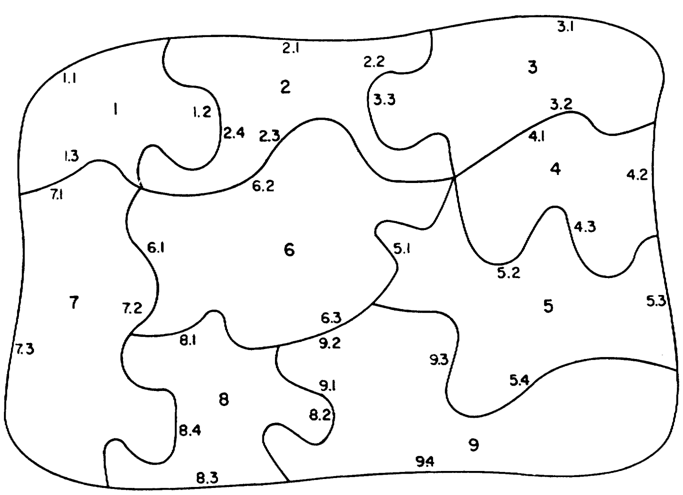


Fig. 4—Illustration of a jigsaw puzzle.

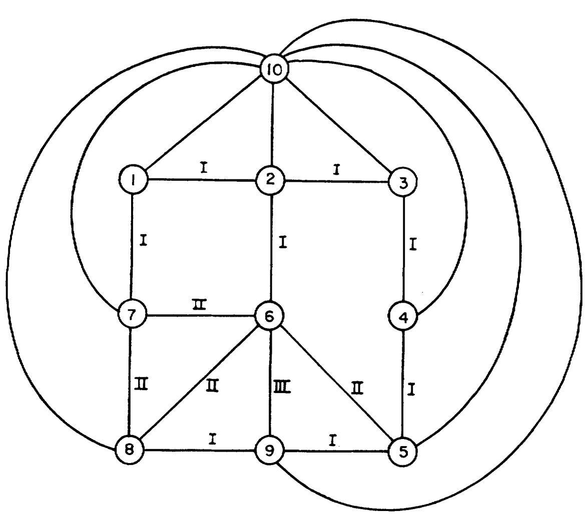


Fig. 5—Mating diagram for puzzle of Fig. 4. (The Roman numerals indicate the type of mating.)

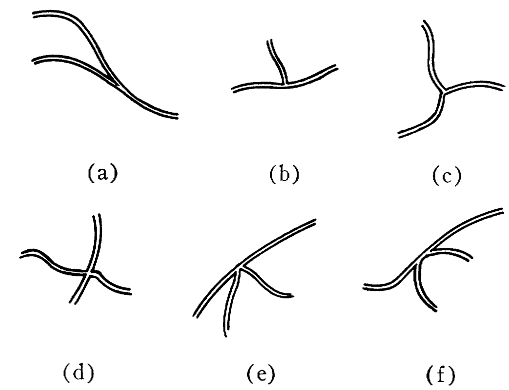


Fig. 6—Slope discontinuities at triradial and quadradial junctions.

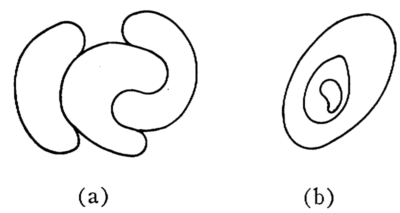


Fig. 7—Puzzles lacking slope discontinuities.

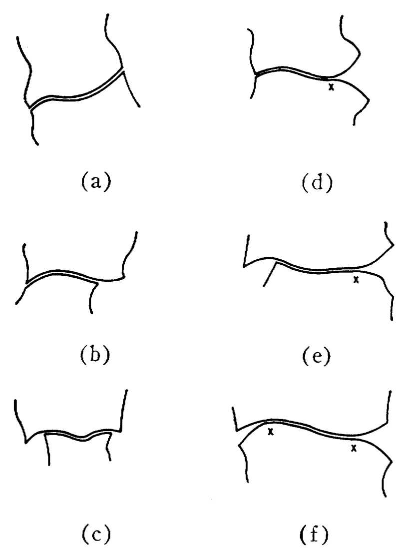


Fig. 8—Types of piece matings. (a) Type I, (b) Type II, (c) Type III, (d) Type IV, (e) Type V, (f) Type VI.

that the pieces lack discontinuities at one or both of the junctions. As an illustration, consider the puzzle of Fig. 4. There are eight matings of Type I, four of Type II, and one of Type III. The type of each mating is indicated in the mating diagram of Fig. 5.

It is clearly possible to construct a puzzle that contains *only* Type I matings [*e.g.*, Fig. 3(a)]. It is also possible to construct puzzles with only Type II matings. The puzzle of Fig. 9(a) consists of five pieces and contains only Type II matings. (The pieces are shown as rectangles for ease of visualization.) By taking this configuration as a building block and using the arrangement shown in Fig. 9(b), an arbitrarily large, all Type-II puzzle can be assembled. The corresponding mating dia-

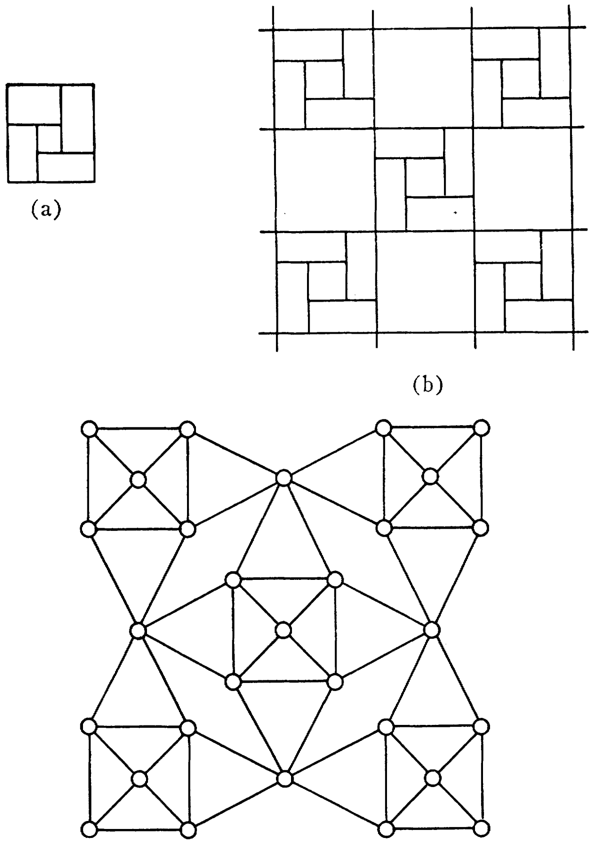


Fig. 9—Illustration of an all Type II configuration.

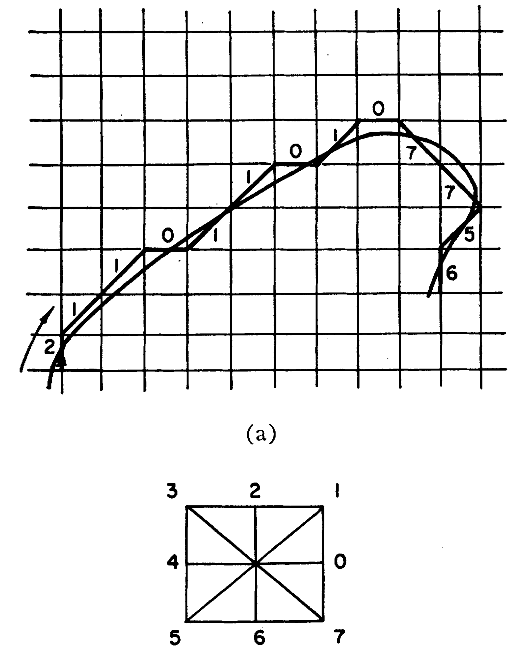


Fig. 10—Chain encoding scheme.

gram is given in Fig. 9(c). Except for certain degenerate cases (*e.g.*, Fig. 7), Types I, II and VI are the only ones from which a single-type puzzle can be constructed. It is relatively simple to show that all other types (III through V) can occur only in mixed-type puzzles.

Encoding and Segmentation

A necessary first step for the computer solution of a jigsaw puzzle is the encoding of the individual pieces. An effective scheme for this is provided by the so-called chain-encoding technique.[5–7] In this scheme, the piece is placed upon a square grid and the intersections between the piece's boundary and the grid lines are marked. The piece is then removed and the grid node closest to each intersection is chosen as a *curve point*.[6] Successive points are connected by means of straight lines to form a *chain*. The chain is thus a particular straight-line segment approximation to the given boundary. [See Fig. 10(a).] Since there are eight possible directions in going from one curve point to an adjacent one, the succession of straight-line segments can be represented by a sequence of octal digits in accordance with the scheme indicated in Fig. 10(b). For uniformity, one traces all contours in a clockwise sense. Thus the chain-code representation of the curve of Fig. 10(a) is 2110110107756 (in the direction of the arrow). Once the boundary information of a puzzle piece is encoded in the form of a sequence of octal digits, it is readily stored or processed in a digital computer.

Since mating between two pieces usually occurs over only a small fraction of their respective boundaries, it is desirable to separate each complete piece chain into a set of *chainlets*, each of which is likely to mate with one and only one chainlet from another piece. The best guide for forming the chainlets is, therefore, provided by the slope discontinuities mentioned previously. Hence, one examines a piece for slope discontinuities and breaks up the chain at each of these points. The resulting chainlets are labeled to conform to the piece from which they are derived. (This scheme of labeling is illustrated in Fig. 13.)

In puzzles in which many pieces possess very few or no slope discontinuities, an additional boundary characteristic is required to serve as a basis for the formation of chainlets. One characteristic suitable for this purpose is *curvature inflection*. However, since a point of inflection implies nothing about the presence or absence of a puzzle junction, its use for forming chainlets is less desirable than the use of slope discontinuities; it is hence used only when segmentation at slope discontinuities is inadequate.

Points at which a chain is severed into chainlets are called *critical points*. In general, it is best if the critical points are identified *before* the encoding is undertaken. However, if this is not possible, the computer can be

[5] H. Freeman, "On the encoding of arbitrary geometric configurations," IRE Trans. on Electronic Computers, vol. EC-10, pp. 260–268; June, 1961.
[6] H. Freeman, "Techniques for the digital computer analysis of chain-encoded arbitrary plane curves," *Proc. Natl. Electronics Conf.*, vol. 17, pp. 421–432; 1961.
[7] H. Freeman, "On the digital computer classification of geometric line patterns," *Proc. Natl. Electronics Conf.*, vol. 18, pp. 312–324; 1962.

programmed to determine the critical points.[7] In the latter case, there will be some uncertainty in the locations of the critical points due to the quantization noise; in addition, some valid critical points may be overlooked and some false ones introduced. No serious difficulties arise from this provided such misidentification occurs in only a relatively few cases.

When two pieces mate on a Type IV, V or VI basis, a slope discontinuity appears in the composite boundary at a point at which neither of the individual pieces had a critical point [*e.g.*, the points marked "x" in Figs. 8(d), 8(e) and 8(f)]. The points at which such slope discontinuities are formed are called *secondary critical points*. They can be used for the formation of additional chainlets in the same way as ordinary critical points. The availability of secondary critical points greatly aids the solution of puzzles involving large proportions of Type IV, V or VI matings.

Feature Determination and Classification

Segmentation into chainlets has the effect of directing the solution of a jigsaw puzzle from one of fitting "pieces" to one of mating chainlets (or portions of chainlets) subject to certain interdependence constraints. The number of chainlets will normally be many times the number of pieces, say four to ten in typical cases. Since the chainlets may also be at arbitrary orientation, it would be most inefficient to make an exhaustive search for mating chainlets. Instead, the chainlets are first classified according to some of their measurable, orientation-invariant *features*.[8] If the selected features are such that similarity of features[9] is a necessary condition for Type I mating, the search for mating chainlets is severely narrowed.

Typical features are the length of the chainlet, the length of the straight line between the end points (*distance vector*)[6] of the chainlet, the number of times the chainlet crosses its distance vector, the ratio of the areas enclosed by the chainlet to either side of the distance vector, and the perpendicular distance between chainlet and distance vector. Additional features can be defined in a similar way.

Algorithms for the computer determination of features are readily obtainable.[6] One notes that a feature derives its value in part from its ability to provide discrimination between nonmatching chainlets and in part from the relative speed with which it can be computed.

The set of m features of a chainlet can be expressed as a vector in m-dimensional feature space. Since similarity of features is a necessary condition for Type I mating, the *feature vectors* of two chainlets that mate on a Type I basis must lie in close proximity. Conversely, if no feature vector lies in close proximity to the feature vector of a given chainlet, that chainlet either forms part of the exterior boundary or mates on a basis other than Type I.

Let the feature vector of chainlet i be denoted by F_i, and its components by f_{ik} ($k=1, 2, \cdots, m$) where m is the total number of features. Two chainlets i and j are said to be *similar* if the relation

$$|f_{ik} - f_{jk}| \leq T_k \tag{1}$$

is satisfied for *all* k, $k=1, 2, \cdots, m$. Here T_k is a positive *threshold* assigned to the kth feature. Two chainlets similar to a third chainlet are not necessarily similar to each other.

The feature separation S_{ij} between two vectors F_i, F_j is given by the relation

$$S_{ij} = \sum_{k=1}^{m} [f_{ik} - f_{jk}]^2 w_k \tag{2}$$

where w_k is a nonzero, positive weighting coefficient assigned to the kth feature. The feature separation between two vectors is zero if and only if corresponding components of the vectors are equal. Note that a "small" feature separation does not imply that the two vectors are similar.

The m thresholds are assigned on the basis of the *precision* with which various features can be determined. The more precise a measurement, the smaller the permissible deviation in the feature, and hence the smaller the threshold. The weighting coefficients are assigned on the basis of the probability distributions associated with the features. For example, if most chainlets in a puzzle have the same length, so that the variance in length is small, the weighting coefficient for the length feature is given a corresponding small value. As a result, large deviations in accurately determined features are not tolerated and deviations in significant features (*i.e.*, those possessing large variances) are emphasized. Generally, the precision of measurement and the significance of a feature are totally unrelated.

The procedure for finding a mate to a given chainlet on a Type I basis is as follows: 1) Threshold tests are applied to the feature vectors of all chainlets. Those chainlets found to be similar to the given chainlet are listed; they will be referred to as *mating candidates*. 2) The feature separations between the given chainlet and each of the candidates are calculated. Similar chainlets are listed in the order of increasing feature separation from the given chainlet. 3) The candidate chainlets listed in 2) are consecutively compared with the given chainlet until a mating chainlet is found. The comparisons begin with the chainlet whose vector has the smallest feature separation from the selected vector. If no mating chainlet is found among the candidates, it is assumed that the chainlet either mates on a basis other

[8] G. S. Sebestyen, "Decision-Making Processes in Pattern Recognition," The Macmillan Company, New York, N. Y.; 1962.
[9] This can also include features that must be *complementary* (*e.g.*, left-right) for two mating chainlets.

than Type I or that it forms part of the exterior boundary.

For example, suppose a Type I mate is sought for chainlet 6.4 in Fig. 13. Possibly, the feature vectors of chainlets 3.2, 7.2, 5.1, 11.2 and 8.3 are similar to the feature vector of 6.4. Of these similar vectors, assume that the vector of chainlet 7.2 has the *smallest* feature separation from the vector of 6.4, and that the vector of chainlet 3.2 is next in order of increasing feature separation. Comparison of chainlets 7.2 and 6.4 (most likely mating candidate and given chainlet) may reveal that these two chainlets are nearly alike. Comparison of chainlets 3.2 and 6.4 may reveal that the next most likely candidate bears little resemblance to 6.4. It would, therefore, be decided to discontinue further comparisons and to assume (subject to possible reevaluation at a later time) that chainlet 7.2 is the mate to 6.4.

In Type I mating, paired chainlets mate over their full length. However, in Type II or IV mating, paired chainlets mate only over a portion of their respective ranges, starting from one common critical point (see Fig. 8). For the latter two types, proximity of the feature vectors is, therefore, *not* a necessary condition for the mating of two chainlets.

To obtain a necessary condition for Type II or IV mating similar to that for Type I, the so-called *subfeatures vectors* are introduced.[10] These vectors are analogous to the feature vector except that instead of referring to an *entire* chainlet, they refer only to specific fixed-length initial and terminal portions of a chainlet. (Subfeature vectors thus always occur in pairs.) The components of a subfeature vector (*i.e.*, the *subfeatures*) are similar in nature to the features of the whole chain, except, of course, for length, which is fixed instead of variable. The length of the initial or terminal portions (measured along the chainlet) is taken as some small fraction of the *average chainlet length* for the particular puzzle. Typically, the fraction may range from 1/10 to 1/3, depending on the standard deviation of the chainlet length. If desired, additional subfeature vectors can be formed by using more than one fixed length. If a chainlet has a shorter length than the selected fixed length, the particular subfeature vectors of this chainlet do not exist. The concepts of similarity and distance defined for feature vectors apply equally to the subfeature vectors.

Puzzle Assembly

The solution of apictorial jigsaw puzzles consists of a succession of boundary matings and assemblies of matching pieces. Initially, mating is attempted on a Type I basis. When this fails, subfeature vectors are formed and Type II matings are sought. This procedure is adequate for puzzles in which matings of Type I and II predominate. Matings of Type III through VI may require the use of secondary critical points.

A contour comparison procedure consisting of three steps has been devised. It is used to determine the mate for a given chainlet after the candidate chainlets have been ordered according to their feature separation from the given chainlet.

In the first step, the direction of the candidate chainlet is reversed. One notes that if all contours are consistently encoded in the clockwise direction, mating chainlets will describe a contour in opposite directions. Hence in order to permit direct comparison between the given chainlet and a mating candidate, the latter must be *inverted*.[6] Inversion is accomplished by reversing the sequence of chain elements and by performing modulo-8 addition of 4 to each element.

In the second step, the candidate chainlet is rotated to bring it into alignment with the given chainlet. The angle of rotation is chosen to be the angle between the lines that connect the initium to the terminus of each chainlet.

After inversion and rotation, the two chainlets will be properly oriented for an element-by-element comparison. In this third step of the comparison procedure, the squared geometric distances between successive points along the two chainlets are calculated. The sum of these squared distances, Δ, is a measure of the *resemblance* between the two chainlets; the greater this sum, the less the resemblance. The use of squared distance rather than linear distance yields a more critical resemblance measure since large differences between chainlets are emphasized.

The assembly of a puzzle can take place in a variety of ways. According to one way, one assembles mating pieces in pairs and repeats the assembly process on the combined pieces until the entire puzzle has been assembled. In this procedure the puzzle is assembled in a sequence of disjoint clusters, the size of each cluster being roughly doubled in each pass. Severe difficulties can, however, be encountered in the assembly of such disjoint clusters. For if an invalid match is made in any cluster, the fact that it is not valid will usually not become apparent until a much later stage of assembly, *i.e.*, when certain clusters fail to mate. For example, in the puzzle in Fig. 4 assume that piece 2 is initially mated with piece 7, piece 1 with piece 6, piece 8 with piece 9 and piece 3 with piece 4. The two invalid matings (2-7 and 1-6) may have occurred either because of noise or because of the general resemblance of all four chainlets. In the next pass, assume that the combined piece 2-7 is mated with 3-4, and 1-6 with 8-9. In the last pass, assume the clusters 1-6-8-9 is mated with piece 5. At this stage, it is discovered that the two clusters 1-6-8-9-5 and 2-7-3-4 cannot be properly joined. One notes that this incorrect solution can now be corrected only at the expense of a major reconstruction of the two clusters.

[10] As is shown later, the subfeature vectors are usually not required if only a few matings of Types II through VI occur in a predominantly Type I puzzle.

A better scheme for the assembly of a puzzle appears to be one that is based on outward spiraling growth centered around one piece. Such a scheme possesses three advantages: 1) Invalid matings are usually detected shortly after their occurrence. 2) The spiral can be readily retraced piece by piece and reconstructed where necessary. 3) The probability of undetected invalid matings decreases as the size of the cluster increases. For this last reason, a great deal of the information pertaining to the assembly of the cluster can be discarded as the assembly progresses. This is, however, not true of the first-described assembly scheme.

The outward-spiral assembly scheme lays stress on the concept of *junction closure*. A *junction* is any critical point located on the exterior boundary of a cluster of pieces. The junctions on the cluster are examined sequentially in a counterclockwise sense and an attempt is made to *close* them, *i.e.*, to surround them completely with mutually mating pieces. Junctions are represented on a mating diagram by mesh windows except for one junction point which is represented by the area outside the diagram. In Fig. 5, the twelve mesh windows and the outside area form thirteen junctions, all of which are readily identifiable in Fig. 4. The number of branches around each mesh window is equal to the number of puzzle pieces around the junction. In the example of Fig. 4, if piece 6 is chosen as a starting point, the spiral assembly procedure closes junctions in accordance with the sequence indicated by the spiral shown in Fig. 11.

Junctions are uniformly closed in the counterclockwise sense. Each time a new piece is added to the junction, a *test for closure* is performed. This test is used to determine the size of the remaining gap at the junction. If the gap is small or nonexistent, the junction is considered to be closed. However, if the gap is large, additional pieces are sought to close it.

When matings of other than Type I are present or when one or more critical points are located on the interior of a mating boundary, it is possible to close several junctions in one assembly operation. This is referred to as *multiple junction closure*. One distinguishes between counterclockwise multiple junction closure, in which the closed junctions are grouped in the counterclockwise direction away from the main junction (piece 5 in Fig. 12) and clockwise multiple junction closure, in which the direction of closures is reversed (piece 12 in Fig. 12). Multiple closure facilitates the solution of puzzles with other than Type I matings. Since multiple closures can occur at any stage of the assembly, the closure test must be repeated over the immediate vicinity of each junction.

As each junction is closed, a new *open* junction is selected. The rule for choosing new junctions is as follows: On the last-mated piece one chooses the *first open junction* located clockwise from the junction just closed. The sequence of closed junctions will then form a spiral traced in the counterclockwise sense and centered around the initially chosen piece.

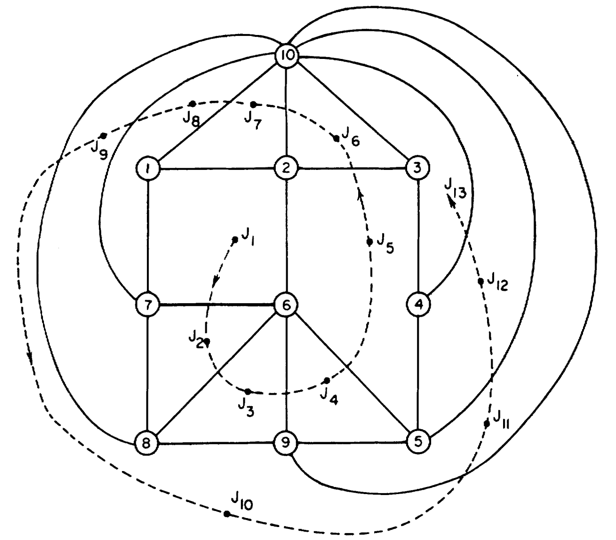


Fig. 11—Mating diagram for puzzle of Fig. 4 showing sequence of junction closures.

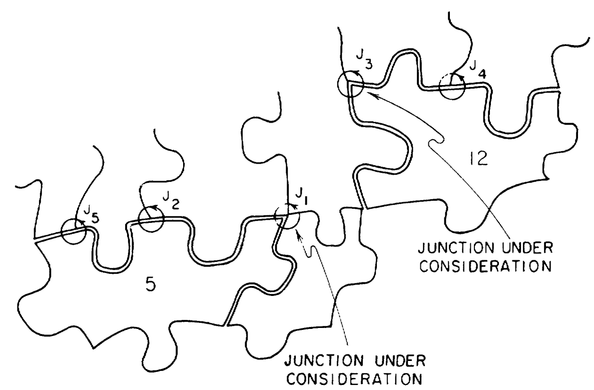


Fig. 12—Multiple closure. Piece 5 closes junctions J_1, J_2 and J_5; piece 12 closes junctions J_3 and J_4.

Ambiguities

A primary source of difficulty in the solution of jigsaw puzzles is the possible presence of ambiguities. An ambiguity is defined as a mating that appears valid but is subsequently shown to be in error. It is not to be confused with nonuniqueness; the latter refers to the existence of two or more equally valid solutions of a puzzle.

Consider the assembly of 12 pieces shown in Fig. 13. As far as one can tell from the figure, pieces 1 through 12 are properly assembled. Suppose, however, that chainlet 27.2 (of piece 27) is identical to 7.1 and that it, instead of 7.1 is mated with 2.2. If now $7.4 \not\approx 27.1$, the closure of the junction 2,1,8,27 will fail since 8.2 will not mate with 27.1.[11] One then examines all the matings around this junction for alternate possibilities. One such possibility is the use of 7.1 instead of 27.2, a possibility that then leads to the correct assembly.

One can imagine the existence of a piece such as 37 for which $7.1 = 37.3$, $7.3 = 37.1$, $7.4 = 37.2$, but $7.2 \not\approx 37.4$, as

[11] The symbol $\approx$ is used to indicate "mates with." The symbol $\not\approx$ indicates "does not mate with." The symbol $=$ is used to indicate that two chainlets are indistinguishable, while $\neq$ indicates that two chainlets are not alike.

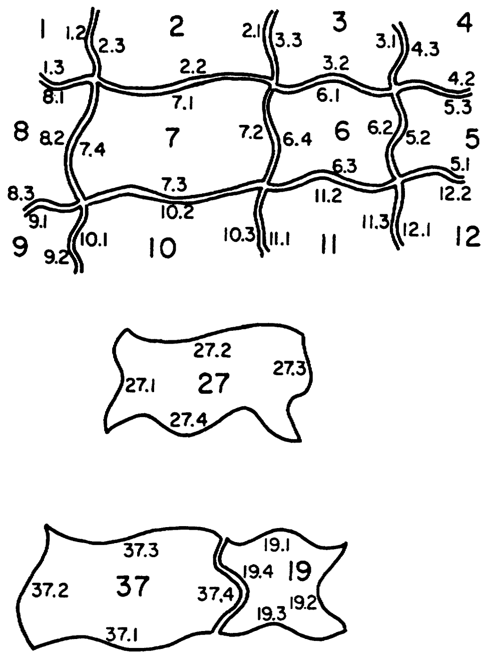


Fig. 13—Illustration of ambiguities.

shown in Fig. 13. Upon mating 2.2 with 37.3, the junction 2,1,8,37 can be closed without any difficulty. Similarly, the next counterclockwise junction 8,9,10,37 will close. The first sign of error appears at junction 10, 11, 6, 37, where 6.4 fails to mate with 37.4.

In addition to the ambiguities involving single pieces, one may also encounter multiple-piece ambiguities. Referring again to Fig. 13, suppose that there exists a piece such as 19 for which $19.4 \approx 37.4$, $19.3 \approx 11.2$, and $19.1 \approx 3.2$. It is then possible to close junction 37,10,11,19 as well as junction 37,19,3,2. One might, therefore, assume that pieces 1, 2, 3, 19, 37, 8, 9, 10, and 11 are all properly assembled. However, if $19.2 \not\approx 5.2$, this assumption is seen to be in error as soon as one attempts to close junction 19,11,12,5. In this case, both 37 and 19 must be replaced by 7 and 6, respectively, before the correct assembly shown in Fig. 13 is achieved. Note that if $19.2 = 6.2$, one has a nonunique puzzle rather than an ambiguity.

Monitoring of Assembly

Assembly errors arise partly as the result of quantization noise which may hide some minor contour differences and partly because of ambiguities. Assembly errors are detected by the failure to close certain junctions. The cost in assembly time of such errors increases with the number of steps required to correct them. It is, therefore, of paramount importance to detect these errors as soon as possible after their occurrence.

The probability of occurrence of an invalid mating around a junction may be decreased by an investigation of alternate possibilities for junction closures (if they exist). Whenever there are several seemingly acceptable choices, a *junction figure of merit* is determined for each such assembly combination. The combination with the highest figure of merit is selected; however, the data on the alternate assembly combinations is retained (at least for a while). When an assembly error is detected, the alternate combinations are examined (in order of decreasing figure of merit) in search for the correct solution.

It is also possible to incorporate certain quality monitoring features into the assembly scheme. Thus one may keep a running score of the figures of merit of the junctions as they are being closed. If, as the assembly proceeds, the value of the figures of merit appears to deteriorate below a preassigned level, one reexamines some of the previously assembled junctions for possible errors. By monitoring the assembly in this way, the presence of some of the more subtle errors may be detected at a relatively early stage.

A simple junction figure of merit can be derived from the sum of the squared distances between mating chainlets around a junction. This sum may be divided by a similar sum evaluated for the next best solution for the closed junction (if one exists). A low value for the quotient signifies that the best solution is significantly better than the next best one.

Illustrative Example

To illustrate the various aspects of the theory of solving apictorial jigsaw puzzles discussed here, a program for aseembling relatively complicated puzzles with Type I, II and III matings was written and tested. In its present form, the program can handle Type IV, V and VI matings in special cases, but not in general.

To simplify the program, certain restrictions have been imposed on the jigsaw puzzles:

1) Type I matings must predominate.
2) There may be no missing or extraneous pieces in the jigsaw puzzle.
3) There must be relatively high discrimination between nonmating chainlets.
4) The outer boundary of the puzzle is assumed to be unknown and irregular.
5) A number of simple ambiguities may be included.

The rough flow chart for this program is shown in Fig. 14. The following is a brief description of the flow chart.

1) *Feature Determination:* Five features are determined for each chainlet. These consist of chain length, length of the distance vector from initium to terminus, angle with respect to x axis of that vector (used in the rotation algorithm), area between the chainlet and the distance vector, and maximum and minimum distances from the distance vector to the chainlet (along the perpendicular to the distance vector).

2) *Storage of Chains and Feature Vectors:* The chains and feature vectors are read in and stored in a section of memory.

3) *Determination and Ordering of Similar Vectors:* Threshold tests are applied to all feature vectors to determine which (if any) of them are similar to the fea-

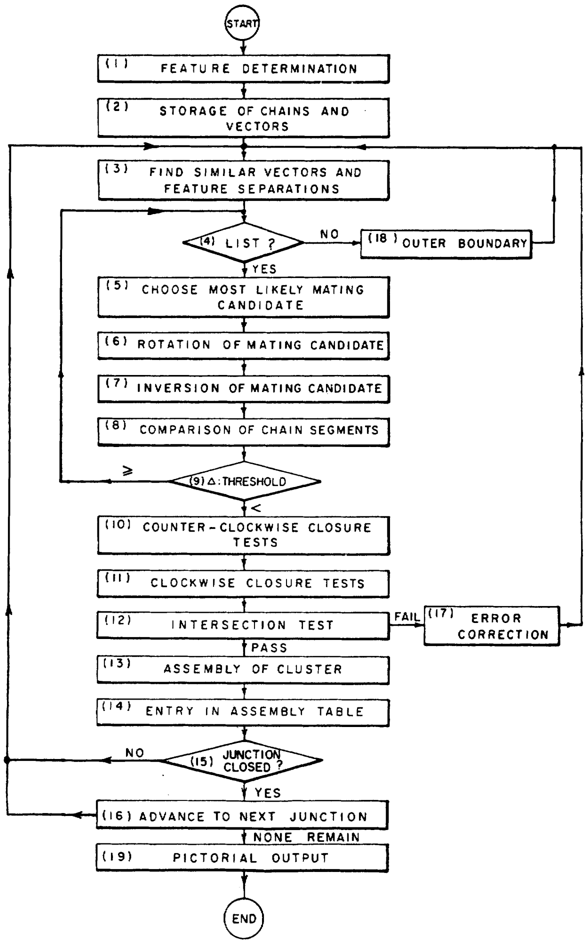


Fig. 14—Program flow chart.

ture vector of the selected chainlet. The feature separations from the similar vectors to the selected vector are determined; the similar vectors are ordered on the basis of these feature separations.

4) *Examination of List:* The list of likely mating candidates is examined. If all candidates have either been used already in the assembly or have failed the mating tests, it is assumed that the selected chainlet is part of the exterior boundary of the puzzle; the program then jumps to 18). Otherwise, the program proceeds to 5).

5) *Choice of Candidate Chainlet:* The chainlet with the "most eligible" vector is selected for further mating tests.

6) *Inversion of Candidate Chainlet:* The direction of the candidate chainlet is inverted.

7) *Rotation of Candidate Chainlet:* The inverted chainlet is rotated into alignment with the selected chainlet.

8) *Comparison Between Selected Chainlet and Candidate:* The selected chainlet is compared element by element with the inverted and rotated candidate chainlet. The sum of the squared distances between successive points along the two chainlets is determined.

9) *Threshold Test:* The sum computed in 8) is compared against a threshold T. If the sum exceeds the threshold, the program jumps to 4), whereupon the next vector on the list is examined. Otherwise the program proceeds to 10).

10) *Counterclockwise Closure Tests:* A number of tests are performed to determine whether the mating chain closes the current junction in the counterclockwise sense. These tests are also repeated over the immediate vicinity of the junction.

11) *Clockwise Closure Tests:* Tests similar in nature to the ones in 10) are performed for closure in the clockwise sense.

12) *Intersection Tests:* Tests for a possible intersection between the candidate chain and the cluster are performed. If these tests indicate that an intersection exists, the error-correcting routine 17) is entered. Otherwise, the program proceeds with 13).

13) *Assembly of Mating Chain with Cluster:* The mating chain is attached to the cluster. The common boundary is removed so that only the exterior boundary of the new cluster remains. This exterior boundary is then used in future intersection tests.

14) *Assembly Table Entry:* An entry is made in a so-called *assembly table.* This entry consists of the mating chain serial number, the angle of rotation and the x-y coordinates of its initium with respect to a common origin.

15) *Junction Closure:* If the junction is not closed, the program jumps to 3) to search for further mating pieces. Otherwise, the next junction is chosen in 16).

16) *Advance to Next Junction:* A new junction is chosen. If all pieces have been assembled, however, the pictorial output routine 19) is entered.

17) *Error-Correcting Routine:* This routine consists of the reexamination of the matings around the present junction, and the assembly of alternate solutions. If one of these solutions satisfies all mating tests, the junction is considered closed. Otherwise, the program halts and prints an error message.

18) *Outer Boundary Routine:* In this routine, the sample chainlet is assumed to lie on the exterior boundary of the puzzle. The program attempts to close the next junction point located in the counterclockwise sense. This assumption may be later revised if necessary.

19) *Pictorial Output:* This routine[12] uses data from the assembly table to produce a pictorial representation of the machine-assembled puzzle (see Fig. 15). Plotting is achieved by means of a line printer. Puzzles of large outside dimensions are broken up by the program into rectangular sections each of which fits on one page of the printer. The pages are then pasted together after plotting.

A program was written for the IBM 1620 computer in the symbolic SPS language. It consists of 2200 instructions. An IBM 1401 computer was used for rapid

[12] This routine was developed by N. Mehta at New York University, N. Y.

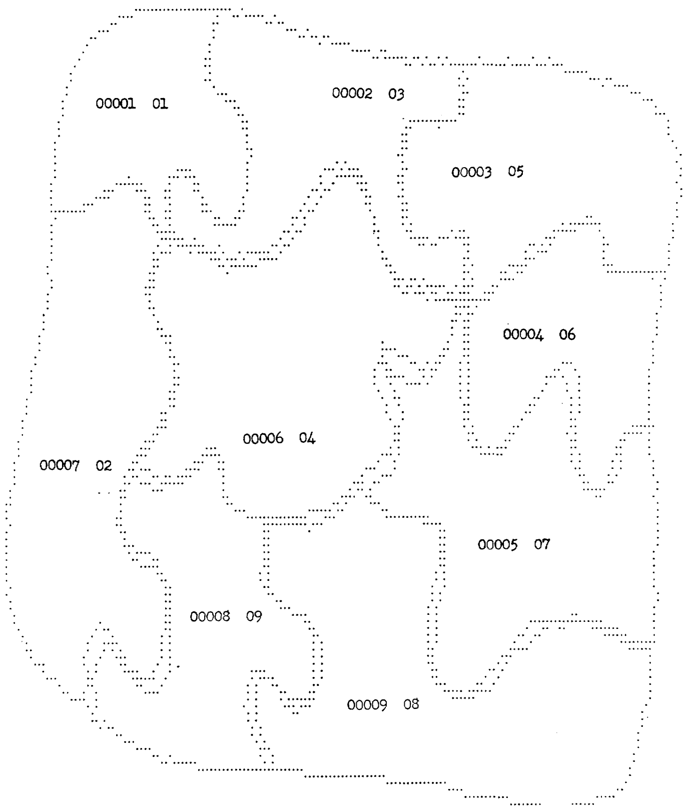


Fig. 15—Pictorial output of assembled puzzle.

TABLE I

Assembly Procedure for Illustrative Example

1) Read Thresholds and Coefficients	35) Clockwise Closure Tests
2) Border Routine	36) C-Clockwise Closure Tests
3) Candidate Number 007.01	37) Intersection Test 1
4) Clockwise Closure Tests	38) Intersection Test 2
5) C-Clockwise Closure Tests	39) Chain Number 004 Matched
6) Intersection Test 1	40) Border Routine
7) Chain Number 007 Matched	41) Candidate Number 005.02
8) Border Routine	42) Clockwise Closure Tests
9) Candidate Number 002.02	43) Type 2/3 Clockwise Closure
10) Candidate Number 002.04	44) C-Clockwise Closure Tests
11) Clockwise Closure Tests	45) Intersection Test 1
12) C-Clockwise Closure Tests	46) Intersection Test 2
13) Intersection Test 1	47) Chain Number 005 Matched
14) Intersection Test 2	48) Border Routine
15) Chain Number 002 Matched	49) Border Routine
16) Border Routine	50) Candidate Number 009.03
17) Border Routine	51) Clockwise Closure Tests
18) Candidate Number 006.02	52) Type 2/3 Clockwise Closure
19) Clockwise Closure Tests	53) C-Clockwise Closure Tests
20) Type 2/3 Clockwise Closure	54) Intersection Test 1
21) C-Clockwise Closure Tests	55) Intersection Test 2
22) Intersection Test 1	56) Chain Number 009 Matched
23) Intersection Test 2	57) Border Routine
24) Chain Number 006 Matched	58) Border Routine
25) Border Routine	59) Candidate Number 008.02
26) Candidate Number 008.02	60) Clockwise Closure Tests
27) Candidate Number 003.03	61) Type 1 Clockwise Closure
28) Clockwise Closure Tests	62) Clockwise Closure Tests
29) C-Clockwise Closure Tests	63) Type 1 Clockwise Closure
30) Intersection Test 1	64) Clockwise Closure Tests
31) Chain Number 003 Matched	65) C-Clockwise Closure Tests
32) Border Routine	66) Intersection Test 1
33) Border Routine	67) Chain Number 008 Matched
34) Candidate Number 004.01	68) End of Assembly

print out, but provision was made for standard printing on off-line card printers.

A sample puzzle was devised for testing purposes (Fig. 4). This puzzle contains two simple ambiguities: Piece 9 mates with pieces 1 and 8, and piece 1 mates with pieces 2 and 9.

The sample puzzle was assembled by the computer in twenty-five minutes. The assembly time was distributed as follows: eight minutes for the initial phases (boxes 1 and 2 in the flow chart), seven minutes for the main assembly (boxes 3 through 18) and ten minutes for the pictorial output routine (box 19).

As the puzzle was assembled, a printout was obtained of each of the main steps of the assembly procedure (see Table I). Table I shows the various mating candidates, the tests performed on them and the results of these tests. The pictorial output for this puzzle (Fig. 15) displays the pieces in their proper orientation; in addition, each piece is labeled with a 5-digit serial number and a 2-digit sequence number. The sequence number indicates the order in which the pieces were assembled.

The first chain segment chosen is segment 1.1. As no mating candidate was found, the border routine was entered (line 2 in Table I). The next chain segment in the counterclockwise direction, segment 1.3, was picked. The first mating candidate examined was segment 7.1. It passed all required tests (lines 4, 5, and 6). Pieces 1 and 7 were assembled. No mate was found to segment 7.2 (line 8). The first mating candidate to segment 1.2 to be examined was segment 2.2. This segment did not satisfy the contour comparison test (line 9). Piece 2 was then rotated through almost 180 degrees and segment 2.4 was compared with segment 1.2 (line 10). This time all necessary tests were satisfied. In line 15, piece 2 was joined to the cluster. The remaining sequence of operations may be deduced in similar fashion from Table I and Fig. 15. (Note that the spiral assembly sequence differs from the one shown in Fig. 11 since the assembly began with piece 1 rather than with piece 6.) Since there were only a few matings of Types II and III, it was possible to assemble the puzzle without the use of subfeature vectors.

Conclusion

The work described in this paper was motivated by the desire to explore the capabilities of the digital computer for solving graphical-data problems. In particular, the study was concerned with the class of problems in which graphical data must be extensively manipulated and eventually "recognized." The jigsaw puzzle was particularly well-suited for this purpose. The solution of jigsaw puzzles requires the application of techniques in graphical data manipulation, in pattern recognition, and in "heuristic" programming. The study provided insight into the structure and characterizing features of graphical contour-line data. The results should be applicable to a large class of problems in the field of graphical data processing.