Isthmus Critical Points for Solving Jigsaw Puzzles in Computer Vision

Roger W. Webster, Paul S. LaFollette, and Robert L. Stafford

Abstract—**A computer vision system that can assemble canonical jigsaw puzzles is described. The most novel aspect of this system is that the methodology presented here derives a new set of critical points that define a feature that can be used in matching partial boundaries (or contours) of planar regions. This global feature, called an isthmus, can be efficiently and reliably computed from the Euclidean skeleton or medial axis transformation (MAT) of an object. A heuristic matching technique using isthmus critical points is applied to the partial boundary matching problem of jigsaw puzzle fitting. The isthmus feature may also be a useful feature in a broader class of image processing problems such as: the narrowing of arteries in medical applications, geographic images, collision avoidance problems in robot path planning, and any application in which the point of narrowest necking of a planar region needs to be located.**

I. Introduction

Partial boundary matching (PBM) is the process of matching shapes based upon matching pieces of their respective boundaries. PBM techniques are useful in shape fitting or assembly applications and in object recognition applications where the entire boundary may not be available. Such applications include: industrial vision systems where parts touch or overlap, environments where shadows may affect portions of the boundary, and in scene analysis with occluded objects.

It is well known that the points that segment the boundary are referred to as critical points. Critical points that have been used include: sharp corners (discontinuity in curvature), points of inflection, and curvature maxima and minima. These points can be computed using local border processing (chain encoding) originally defined by Freeman (see [1]). In this paper we present a method for deriving a new set of critical points that describe a feature that can be used in matching partial boundaries of planar regions.

This global feature, we call an isthmus, can be reliably computed from the Euclidean skeleton or medial axis transformation (MAT) of the object. A heuristic matching technique using isthmus critical points is applied to the partial boundary matching problem of jigsaw puzzle fitting.

The Jigsaw Puzzle Problem

The task at hand is to assemble canonical jigsaw puzzles in a robust fashion. The jigsaw puzzle problem is originally summarized by Radack and Badler in [2] as:

> "Given a set of simply connected planar regions
> (silhouetted puzzle pieces), rotate and translate each
> piece so that the pieces fit together into one region,
> with no significant area gaps or overlapping pieces."

The jigsaw puzzle problem was chosen because it contains a number of problems endemic to many machine vision applications: shape description, partial boundary matching, pattern recognition, feature extraction, and heuristic matching. A survey of the different approaches to the jigsaw puzzle problem and previous work in partial boundary matching can be found in [3].

Manuscript received May 11, 1990; revised March 23, 1991. This work was supported in part by the National Science Foundation (NSF) under Grant No. USE-9050371 and in part by the Faculty Grants Committee of Millersville University.

R. W. Webster is with the Robot Vision and Artificial Intelligence Laboratory Department of Computer Science Millersville University Millersville, PA 17551.

P. LaFollette and R. Stafford are with the Department of Computer Science, Machine Vision Laboratory, Temple University, Philadelphia, PA 19122

IEEE Log Number 9100398.

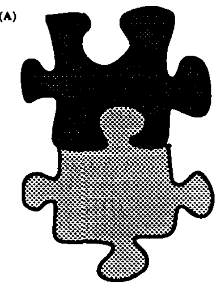


Fig. 1. The match segment of piece (A) and piece (B) is defined as the set of contiguous border points that are in common when (A) and (B) are properly assembled.

A puzzle piece is defined as a simply connected planar region. A jigsaw puzzle will be defined as a set of puzzle pieces that, when properly assembled, fit together into one region. All puzzle pieces are processed as binary images. Only shape information will be utilized in the solution. Pictorial information or scene analysis will not be a factor. Many apictorial jigsaw puzzles require a nontrivial degree of intelligence to assemble. The jigsaw puzzles used in this paper will have a unique solution, i.e., each puzzle piece has a unique location and orientation within the puzzle.

The essence of the jigsaw puzzle problem can be found where puzzle pieces mate. Two pieces that mate share a common border called the match segment (Fig. 1). The main task of a computer vision program to assemble a jigsaw puzzle is to locate and mate these match segments.

Some jigsaw puzzles may have mating pieces in which the match segment contains no distinguishable or easily extractable points (critical points). The teardrop puzzle piece (Fig. 2) exemplifies this characteristic. The process of finding the match of puzzle pieces with match segments of this type becomes an arduous computational task. There is little alternative but to attempt to match each point with every other point along every other puzzle piece.

The previous methodologies used to perform jigsaw puzzle matching, Freeman and Garder [4], Hirota and Ohto [5], Nagura, *et al.* [6], Radack and Badler [2], are all based on the notion of extracting critical points from local border information. One of the problems with using these critical points (sharp corners, points of inflection, curvature maxima, and curvature minima) is that the number of match segments can become many times the number of puzzle pieces.

In Freeman's classic paper on jigsaw puzzle matching [4] he describes the principal complication incurred with jigsaw puzzle matching and partial shape matching, in general, which has been that the problem has a tendency to run away. If the number of critical points extracted is sufficiently large, the matching algorithm can take a prohibitive amount of computation time.

We present a new approach to this problem based on a global feature called an isthmus. The isthmus feature yields a pair of critical points that are bounded together to produce a candidate match segment. These isthmus critical points can be extracted from a global shape descriptor, the MAT or Euclidean skeleton. The MAT is a technique originally developed by Blum [7] that can reduce a planar object to a skeleton or stick figure.

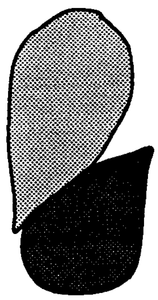


Fig. 2. The teardrop puzzle piece match depicts a match segment that contains no obviously discriminating critical points.

(a) (b)

Fig. 3. Puzzle pieces that introduce many false candidate match segments. (a) Sharp corners as critical points. (b) Inflection points as critical points.

This methodology of using a higher-order entity (isthmus critical points) rather than just single critical points (e.g., sharp corners), helps control the number of matches to evaluate. For example, a puzzle piece with N single critical points would yield potentially $N * (N - 1)$ candidate match segments. Thus, a piece with only 10 critical points may have, as an upper bound, 90 candidate match segments. The matching process can quickly get unruly. It should be noted that $N * (N - 1)$ is an upper bound and in the case of no backtracking not all of these segments would need to be checked.

The puzzle pieces depicted in Fig. 3 would generate many inflection points and sharp corners that can yield prohibitive computation times for the matching process. This is because the number of candidate match segments would be many times the number of puzzle pieces.

II. Shape Matching

Techniques for shape matching are classified into two categories: internal techniques and external techniques. Internal techniques deal with the area or global shape of an object. External algorithms traverse the local boundary of the object while attempting to quantitatively analyze and categorize the local shape.

Many internal techniques are concerned with the process of determining whether or not two regions enclosed by curves are the same shape. Most of the internal techniques used include either the derivation of a feature vector or a representation such as Fourier descriptors (FD's).

Internal representations are generally not applicable to partial boundary matching because they are derived as a function of the entire object and do not contain detailed curvature information. FD's, which use the Fourier transform of the border points $B(x, y)$ of the object, have been successfully used in object recognition problems (see [8]). The problem with FD's is they are not easily adaptable to piecewise curve analysis.

Feature vectors contain various measures that are used in object recognition systems. Such classic feature measures include the following.

1) Moments.
2) Perimeter.
3) Area.
4) Circularity (ratio of area to perimeter).
5) Elongatedness (longest chords).
6) Euler number.
7) Integral optical density (grey scaled area).
8) Length and width.
9) Aspect ratio (ratio of length to width).
10) Horizontal and vertical projections (signatures).

Feature vectors, while effective in object recognition, are not sufficient for partial shape processing. This is because many of the classic feature measurements do not provide detailed curvature information. They can be effective, however, in comparing an object's features to a stored model of the object.

The method of using invariant moments is an example of an internal technique. The area normalized central moments computed in relation to the principal axis are invariant to scale, rotation, and translation. Hu, in [9], has shown that from the second and third moments, a set of seven invariant moments can uniquely define the global shape of an object. Hu's seven moments are invariant to rotation, translation, and scale. Invariant moments have been applied to various object recognition tasks. The problem with moments is that first order moments bear substantial information about simple objects, but lose their significance in direct correlation with the complexity of the object [10]. Moments can be computationally expensive and are not easily applicable to partial boundary matching.

The MAT is an internal, space domain technique and can be appropriate for curve analysis because the skeleton derived in the MAT is a representation of the global shape or curvature of the object. The MAT has been used in many applications of shape recognition [11], [12]. Chords across shapes have also been proposed as measures of shape (see [13]) but have met with prohibitive computational complexity.

External techniques are based on the notion of local border processing. Much of the work relating to the processing of borders uses the concept of chain encoding developed by Freeman. Nagura *et al.* have also described an external technique based on the concept of a curvature function that is defined as the derivative of the angular function of the border points of an object [6].

III. Isthmus Points and Interlocking Puzzles

An inspection of conventional jigsaw puzzles shows that a large class of the match segments of pieces that mate also interlock. Isthmus critical points can be used to detect some interlocking shapes (puzzle pieces).

Fig. 4 illustrates the notion of isthmus critical points. It can be defined as follows:

> Given a planar region, an isthmus line is a chord partitioning the region whose length is locally minimum. An isthmus point is the midpoint of the isthmus line. The endpoints of the isthmus line will be called isthmus critical points.

An isthmus is a global feature of a planar region. Isthmus critical points can be derived from this feature.

Isthmus critical points have several beneficial features: 1) An isthmus defines a pair of critical points that are bounded together and yield a candidate match segment, 2) They can be efficiently extracted from a global shape descriptor, 3) They are rotation and translation invariant, 4) They are invariant to equal scaling in X and Y, 5) They can detect some interlocking shapes (puzzle pieces).

Two puzzle pieces are said to be interlocking iff they can not be assembled or disassembled (pulled apart) without taking the

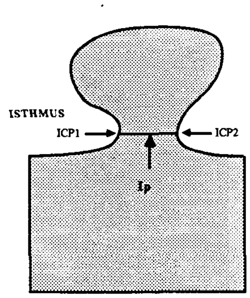
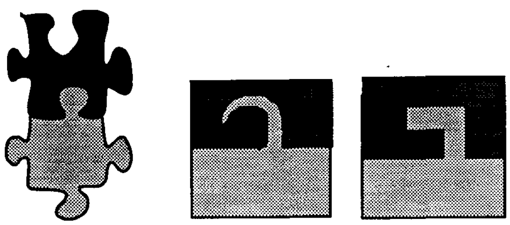


Fig. 4. Sample isthmus critical points. The isthmus of this shape is the line segment across the piece. The isthmus point is the midpoint of the isthmus denoted by Ip. The isthmus critical points, denoted by ICP1 and ICP2, are the border points that are minimally distant from Ip.



(a) (b) (c)

Fig. 5. Sample interlocking pieces. (a) Conventional interlocking pieces with isthmus. (b) Snake example without isthmus. (c) Interlocking pieces without isthmus.

pieces out of the XY plane (see Fig. 5). Not all interlocking pieces contain isthmus points. The "snake" example in Fig. 5(b) shows an interlocking puzzle piece without an isthmus. However, a sampling of conventional jigsaw puzzles shows that many of the interlocking pieces do have isthmus points. The algorithm for jigsaw puzzle matching described here is concerned with match segments that are interlocking and contain at least one unique isthmus.

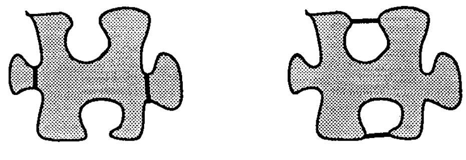
Fig. 6 illustrates the notion of positive isthmii and negative isthmii. The matching algorithm attempts to mate a positive isthmus of one piece with the matching negative isthmus of another piece.

One of the more elegant features of isthmus critical points is that they can be extracted from a global shape descriptor, the MAT. Of course, since the MAT is equivalent to the border in terms of information content it follows that anything that can be found in the MAT can also be found from the border points. The MAT, however, contains global information, namely the opposite points of the boundary at each skeleton point. This makes the extraction of isthmus critical points (points of narrowest necking) from the skeleton computationally more efficient than from local border processing.

IV. A Multiplicity of Skeletons

The term "skeleton" has been widely referenced in the literature. This term has a number of aliases that include: medial axis transform, medial axis, medial axis skeleton, symmetric axis transform, symmetric axis, stick figure, distance transform, and thinned shape. Indeed, the concept of a skeleton has been used to refer to a number of different constructs. The basic premise, however, is to reduce a planar shape in a digital image to a line drawing or stick figure. The stick figure allows for easier extraction of shape information.

Skeleton algorithms are used in numerous applications including: printed circuit board inspection, asbestos fiber counting, character recognition, chromosome shape analysis, soil cracking patterns, fingerprint classification, facsimile, and data reduction for map storage. Since Rosenfeld and Pfaltz's paper in 1967 [14] there have been many algorithms proposed to develop the skeleton of a discrete planar region [15]–[25].



(a) (b)

Fig. 6. Sample isthmii. (a) Positive istmii. (b) Negative isthmii.

The various skeleton algorithms, however, do not produce the same results. Each algorithm produces slightly different skeletons. The fundamental problem with the development of the actual Euclidean skeleton in a digital image is the difficulty of measuring the equality of distance between pixels in a grid (the equidistant property problem). The jigsaw puzzle problem requires a skeleton algorithm that uses Euclidean distances in order for the skeleton to be rotation invariant.

Discrete skeletons are dependant on the distance transform used. Danielsson in [25], Borgefors [24], and Dorst [18] proposed "nearly" Euclidean distance transforms that can be used to generate the discrete skeleton. We have used both the Danielsson algorithm [25] and our own [3]. Within the jigsaw puzzle problem domain both algorithms yield satisfactory results, i.e., discrete Euclidean skeletons that preserve connectedness. The proof of connectedness for the Danielsson algorithm appears in [25].

V. Hardware and Software Environment

The vision system used is the Imaging Technology FG101 image processing system. The frame buffer provides 1024×1024 pixel resolution with an 12 bit grey shade. A Quasar CCD camera is used as the input device. A Digital Equipment Corporation (DEC) VR241-A RGB color monitor is the output device of the vision system. A Sun Sparcstation® 330 computer is the main computer vision controller. The Sun Sparcstation 330 is configured with 16 Megabytes of memory, two 699-megabyte hard discs, and a cartridge tape drive. The hardware is depicted in Fig. 7.

The operating system is UNIX® (with Openwindows). The vision boards are addressed via a device driver. The UNIX device driver and all the application software is written in the "C" language.

VI. The Puzzle Matching Process

The process starts by placing each puzzle piece at a random orientation in the field of view of the camera. A backlight box is used to minimize the effects of shadowing. This produces a crisp binary image. After a frame grab operation is performed, the next task is to find the object's border. The puzzle piece's location in the image is not known a priori. A simple border following algorithm returns the array of border points and the area of interest box denoted by (max-x, max-y, min-x, min-y). The border points are stored in the system's main data structure.

® Sparcstation and Openwindows are trademarks of the Sun Microsystems Corporation.

® Unix is a trademark of AT&T.

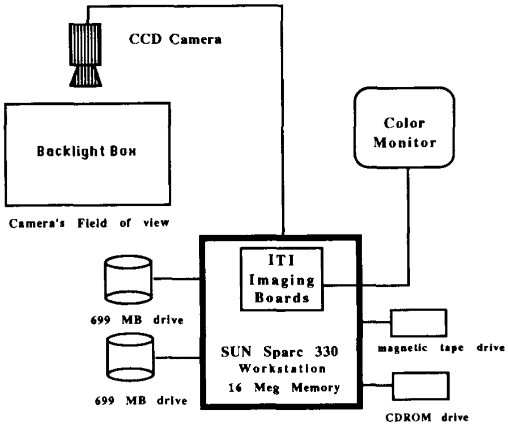


Fig. 7. Hardware environment.

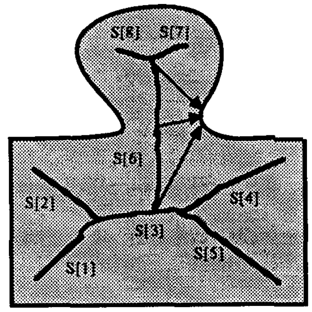


Fig. 8. The skeleton line segments. The arrows show the elevations to the nearest border point along skeletal segment $S[6]$. A local minima of elevations along a skeleton segment indicates an isthmus point.

Computing Isthmus Points from the Skeleton

Using the area of interest box a skeleton algorithm is then performed on the piece. The next step is to recursively traverse the branches of the skeleton looking for local minima. A local distance minima is defined as follows: Let S be a skeleton. A skeleton is a line drawing comprised of line segments. All such segments are connected. Let a skeleton segment $S[i]$ be the union of all skeletal points between an endpoint and an intersection point. The union of all $S[1], S[2], S[3] \cdots S[n]$ for $1 \leq i \leq n$ of $S[i] = S$ (see Fig. 8). Also, $S[i]$ intersected with $S[j]$, for all $i <> j$, is a special point called an intersection point. Along each skeleton segment is a function that maps the position on the segment into elevations. A local minima point along a skeleton segment indicates an isthmus point. The isthmus critical points (ICP) are computed as the endpoints of the isthmus line (see Fig. 4).

The algorithm then marks those border points on the puzzle piece that are isthmus critical points. This process recursively traverses the skeleton segments and stores all the ICP's in the main data structure and accents them on the monitor. Fig. 9 shows the Euclidean skeleton extracted from the Euclidean distances and the positive isthmus critical points. The process of extracting the negative isthmus critical points (Fig. 10) from the exoskeleton is also performed and these points are stored in the system data structure.

The endoskeleton is the internal skeleton of a planar region. The exoskeleton is the external skeleton of a planar region (the skeleton

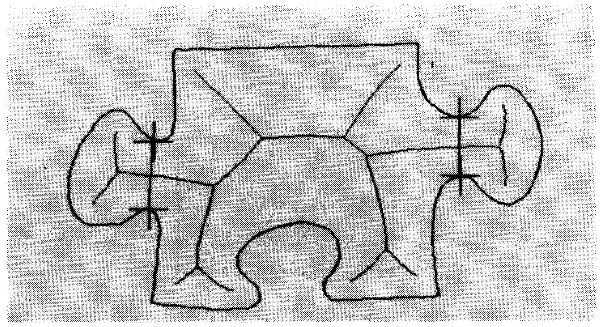


Fig. 9. Euclidean endoskeleton and positive isthmus critical points.

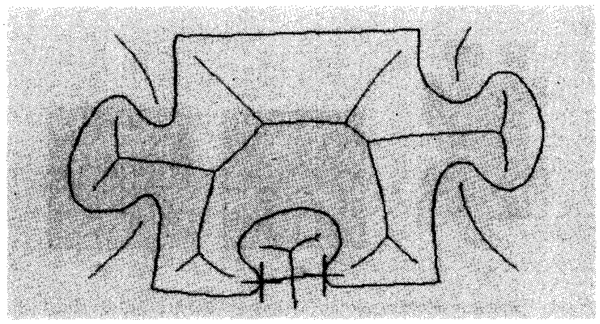


Fig. 10. Euclidean exoskeleton and negative isthmus critical points.

of the area outside the object). The exoskeleton is bounded by boxing the object and calculating the skeleton inside this boxed area. This window or box must be large enough to compute the exoskeleton correctly.

For each puzzle piece, the main data structure stores:

1) The border points $B(x, y)$.
2) The total number of border points.
3) The endoskeleton and exoskeleton.
4) All the isthmus critical points (positive and negative).
5) Features about the isthmus points.

Reducing the Search Space—Heuristic Matching

The matching algorithm begins by attempting to mate positive isthmus match segments with negative isthmus match segments. This yields candidate match segments.

The number of matches of all positive match segments to all negative match segments can be very large. In a typical 24-piece jigsaw puzzle there can be more than 2000 matches to process. To reduce the search space many matches can be eliminated with some heuristic processing. For example: IF the distance between the negative isthmus critical point pair is less than the distance between the positive isthmus critical Point pair THEN discard from the matching process. The reason for this is that Isthmus points are the points of narrowest necking. If the negative isthmus point opening is less than the positive's isthmus distance it can't possibly fit.

Another heuristic used is: IF the absolute value of the difference of the two path lengths (of the match segments) is greater than 50% of the longer segment THEN discard from the match. This heuristic is used to simply weed out two match segments in which their respective lengths are not even close and thus stand little chance of being matched.

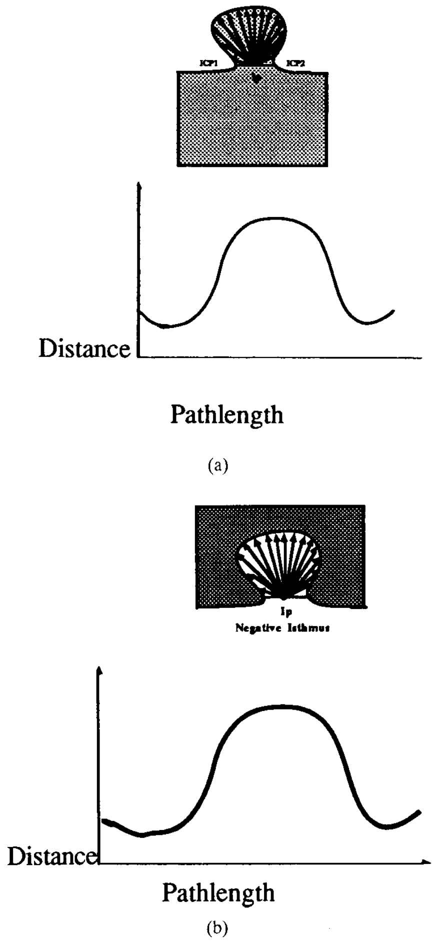


Fig. 11. (a) The isthmus distance function of a positive isthmus. (b) The isthmus distance function of a negative isthmus.

After the heuristic preprocessor has removed a number of potentially bad candidate match segments then the viable candidate matches are stored in the system data structure. Next, some method of partial boundary matching is necessary. The method used here is a correlation of two functions we call the isthmus distance function (IDF).

The IDF is derived by taking the distance to the border points from the isthmus point. The IDF is a function of the match segment. The pair of isthmus critical points (which defines the match segment) serves as the start and stop points of the IDF function. Recall, an Isthmus defines a pair of isthmus critical points that are bounded together by the Isthmus extraction process. The isthmus distance function is derived for both positive and negative isthmii. Fig. 11 illustrates the IDF of two isthmii that mate.

The IDF's are then correlated to determine best fit. The correlation procedure depicted in (1) is used. The maximum value stored in $R(m)$ indicates the offsets where $g(x-m)$ best fits $f(x)$:

$$R(m) = \frac{\sum \left[f(x) - \frac{\sum f(x)}{n}\right]\left[g(x-m) - \frac{\sum g(x-m)}{n}\right]}{\sqrt{\sum \left[f(x) - \frac{\sum f(x)}{n}\right]^2 \sum \left[g(x-m) - \frac{\sum g(x-m)^2}{n}\right]}}. \tag{1}$$

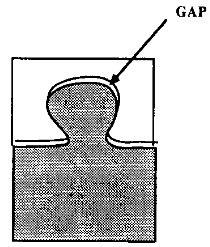


Fig. 12. The GAP is the area in between two puzzle pieces when assembled.

In (1) $R(m)$ is the correlation array $f(x)$ and $g(x-m)$ are the two IDF functions to be correlated. The summations are taken over the range where $g(x-m)$ is defined; $R(m)$ is scaled to the range -1 to 1. The length of the functions $f(x)$ and $g(x)$ is determined by the match segment.

The correlation values are then used to determine where the best fit occurs within the two match segments. The offset produced by the correlation function is used to align the two match segments. The GAP is then calculated. The GAP measure is the area or amount of space in between the two puzzle pieces when they are fitted or assembled (Fig. 12). This amounts to computing the sum of both the positive gap that is the overlapping portions (area) of the match and negative gap that is the underlapping area over the match segment.

The GAP represents how well the shapes (pieces) fit. Normally, one would expect that the smaller the gap, the better the match. However, it is possible that a short match segment will have a small GAP but not necessarily be a good fit. Thus, the average gap as a function of the match segment is calculated.

The average gap is represented by taking the gap divided by the path length (PATH-LENGTH) of the match segment: AVE-GAP = GAP / PATH-LENGTH. The potential matches are then sorted, in ascending order, by AVE-GAP. The matching algorithm takes the smallest value of AVE-GAP for the best fit. The program then rotates and translates the puzzle pieces into assembled position. The correlation offset is used to properly align the two mating puzzle pieces.

After the program has mated two match segments it removes them from the list of candidate matches. It also gets rid of impossibilities, i.e. after mating piece 1 with piece 2 one can not thereafter have a match of piece 2 with piece 1 with different isthmii. A rudimentary check is then performed to determine if there is too much overlap or underlap. The program proceeds down the list of sorted AVE-GAP values until all puzzle pieces are mated or a stopping condition arises. The matching algorithm is shown in Fig. 13.

VII. Experimental Results

The 24-piece jigsaw puzzle used as a test case is depicted in Fig. 14. The puzzle was chosen at random from a set of conventional off the shelf jigsaw puzzles. This puzzle is a difficult one because it contains many different isthmii and many different match segments. There are some match segments that appear, to the human eye, to be very similar.

The shape matching program calculated 2377 possible matches. The heuristic processing reduced the search space to 126 possible matches (2251 matches eliminated). For each of the 126 possible matches the two corresponding match segments were correlated as per the equation in (1). Correlation is expensive so it is important to reduce the search space via the heuristics (reduced from 2377 to 126 possible matches).

```
Algorithm Puzzle Piece Matching.
Begin
  For each Negative Isthmus on each Piece do
  Begin
    For each Positive Isthmus on the N-1 pieces
    Begin
      If any Heuristic Tests Fail Then
        Discard from the matching process.
      Else
      Begin
        Correlate the two Isthmus Distance
         Functions.
        Store Highest Correlation value.
        Compute AVE-GAP.
      End {else}.
    End {for other N-1 pieces}
  End {for each Negative}

  QuickSort all candidate matches by AVE-GAP.
  For each candidate match:
  Begin
    Print Puzzle Piece Mating.
    Move Puzzle Pieces (Rotate and Translate
     in the Image).
    Eliminate impossibilities—check intolerable
     overlap condition.
  End {for each }
End. {end Algorithm }
```

Fig. 13. Puzzle matching algorithm.

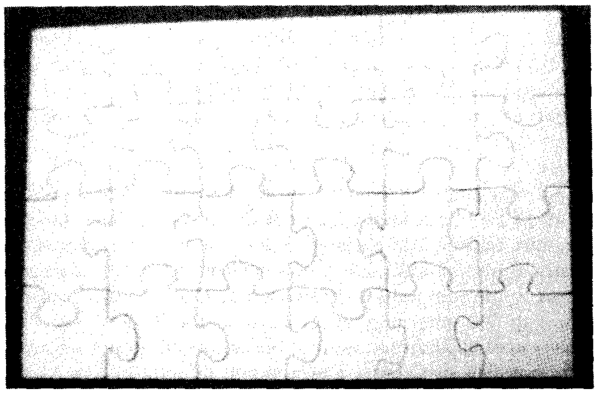


Fig. 14. 24-piece jigsaw puzzle used as test case.

After the correlation of the 126, additional heuristics can be performed to further reduce the search space. Correlation values less than 0.80 are discarded from the space because this suggests a poor correlation (poor fit).

The GAP is then calculated on the reduced set of 126. Recall, the GAP is calculated as: GAP = POS-GAP + NEG-GAP where POS-GAP is the amount of overlapping area and NEG-GAP is the underlapping area of two pieces assembled together. All matches with a GAP to POS-GAP ratio of greater than 50% can be eliminated. This is because any match in which more than 50% of its GAP is positive constitutes an intolerable overlapping condition.

After these additional heuristics are performed, the puzzle matching algorithm generated 37 candidate matches from the 126 possible.

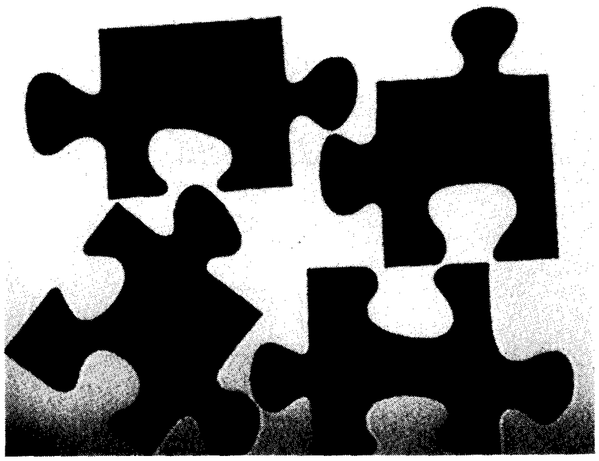


Fig. 15. Unoriented sample jigsaw puzzle pieces.

Puzzle Matching

A sampling of four pieces from the 24-piece puzzle is shown in Fig. 15. These puzzle pieces were chosen because they represent one of the more complex matches of the puzzle. An inspection of the sample pieces shows that there exist match segments that are similar.

The puzzle matching algorithm proceeds by selecting the best puzzle piece match, i.e., the one with the lowest AVE-GAP value. The piece mating is then printed and the pieces are rotated and translated into the correct assembled position.

Fig. 16 shows the four sample puzzle pieces, their Euclidean skeletons and their isthmus critical points. The program correctly assembles these sample puzzle pieces and performs the actual rotations and translations in the image (see Fig. 17).

Fig. 18 illustrates the effects of digitization error on the matching process. The puzzle pieces (in Fig. 18) were assembled together and placed in view of the camera. For each piece, the other $(N-1)$ pieces were carefully removed from the field of view without moving the remaining piece. A freeze frame operation was then performed. Thus, the pieces were input into the system in their assembled position. When the puzzle pieces are displayed they should be perfectly assembled with no gaps or area holes.

The four pieces in Fig. 18 should show as a perfect match (already assembled). They do, however, show a small gap between them. This gap is the best that the vision system would be able to obtain.

The puzzle matching algorithm can not be used blindly, however. As the algorithm assembles more and more pieces eventually it will incorrectly assemble two pieces. As with human puzzle assembly, if two pieces actually do fit together they will be assembled. Later the human discovers that this is an incorrect match because there are pieces that do not fit anywhere in the present configuration or two puzzle sections can not be coagulated. The human performs a version of backtracking to discover the mismatch.

Our computer program correctly assembled almost all of the 37 actual matches in the 24-piece puzzle used as a test case. The one mismatch caused a coagulation problem that forced a stopping condition. This mismatch can be handled with backtracking. Backtracking must be performed because of computation errors, digitization errors and because many isthmii are very similar, hence producing similar computational values. Fig. 19 shows the mismatch assembly. Fig. 20 shows the entire puzzle assembled with backtracking implemented. The puzzle pieces are linearly scaled down in x and y to fit on the

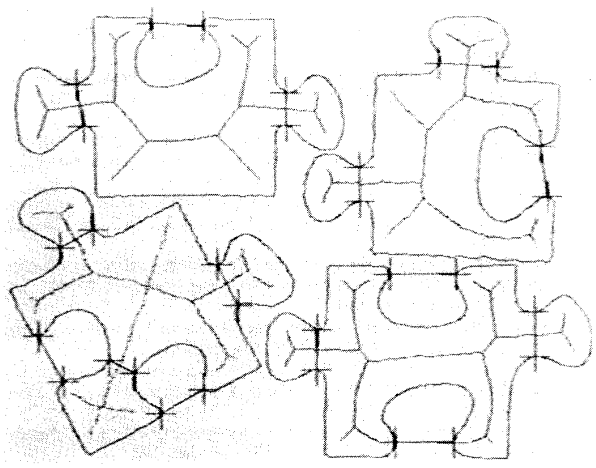


Fig. 16. Euclidean skeleton and isthmus critical points.

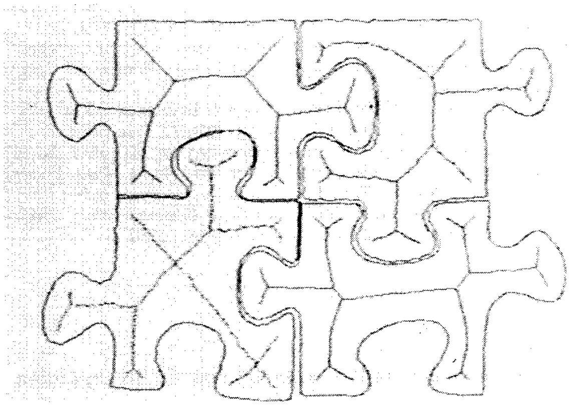


Fig. 17. Assembly of four sample pieces.

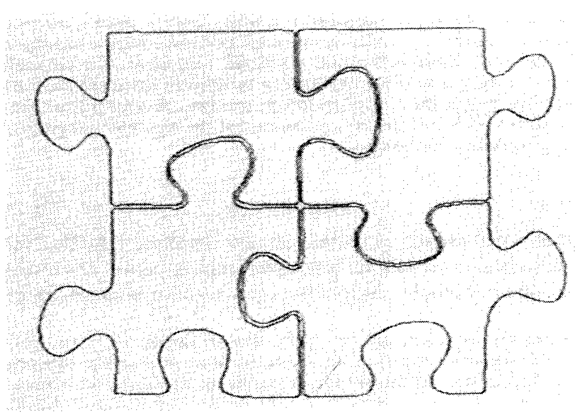


Fig. 18. Digitization effects of piece assembly.

monitor.

Although the test case 24-piece puzzle is a completely interlocking puzzle, we expect that the puzzle matching algorithm will work on a subset of noncompletely interlocking puzzles as well. It is

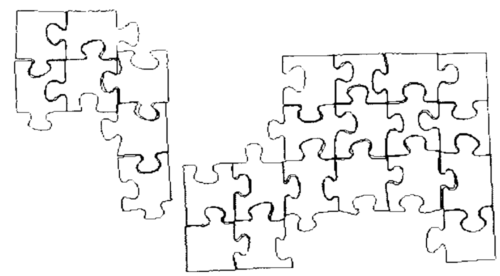


Fig. 19. Assembly mismatch.

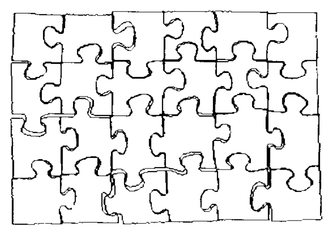


Fig. 20. Completely assembled puzzle.

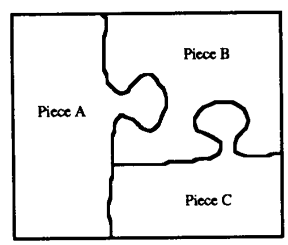


Fig. 21. Noncompletely interlocking pieces.

possible that the assembly of interlocking puzzle pieces will assemble a noninterlocking match segment.

For example, Fig. 21 depicts three pieces in which piece A is assembled with piece C even though they are not interlocking. This is because piece A is assembled to piece B and piece B to piece C. Pieces A and B are interlocking. Pieces B and C are interlocking. However, piece A to C is not interlocking. It is therefore possible that the algorithms described in this paper will correctly assemble some noncompletely interlocking jigsaw puzzles as well.

VIII. Conclusion

The techniques and algorithms described in this paper were implemented in a set of computer programs and applied to the partial boundary matching problem of jigsaw puzzle fitting. An illustration of an actual assembly of a 24-piece jigsaw puzzle by these programs was given.

The methodology presented here uses a higher order entity called isthmus critical points rather than just single critical points (e.g. sharp corners, inflection points), which helps control the number of matches to evaluate. The class of jigsaw puzzles that this method can solve is conventional interlocking puzzles in which each match segment contains a unique isthmus. Nonunique isthmii (match segments) can be resolved with the use of backtracking.

In addition to an effective solution to the jigsaw puzzle problem, we have advanced the notion of a global feature of a planar object, called an isthmus. A method for reliably computing the isthmus feature from the Euclidean skeleton of an object and deriving a new set of critical points (isthmus critical points) that describe the feature has been presented.

Suggestions for Future Work

A set of programs that integrates many different critical points (sharp corners, inflection points, curvature maxima, and isthmus points) may broaden the class of puzzles solvable. Thus, if a solution is not possible using isthmus points, the algorithm could then use sharp corners or other points as the critical points.

The use of parallel processing to mate match segments could be used. It may be possible to store each puzzle piece's set of match segments in a small processor and then in one probe determine which of the other pieces best match. This could speed up the matching process. Future work might also include using a robot to physically assemble the puzzle. The robot end effector could be a suction or vacuum device to manipulate the pieces. The vision system could instruct the robot to perform the actual piece rotations and translations.

Acknowledgment

The authors wish to thank the IEEE reviewers for their very insightful suggestions, Dr. Giorgio Ingargiola and Dr. Charles Kapps of the Department of Computer Science at Temple University, Dr. Paul Ross of Millersville University and Dr. Marvin Ziskin of the Temple University School of Medicine and for their comments on earlier drafts of this paper.

References

[1] H. Freeman, "Boundary encoding and processing," in *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1970, pp. 101–120.

[2] G. Radack and N. Badler, "Jigsaw puzzle matching using a boundary-centered polar encoding," *Computer Graphics, and Image Processing*, vol. 19, pp. 1–2, May, 1982.

[3] R. Webster, "Partial boundary matching and shape fitting using the medial axis transformation," Ph.D. Dissertation, Temple Univ., Philadelphia, PA, May 1988, pp. 1–208 (also available through University Microfilms International, Ann Arbor, MI).

[4] H. Freeman and L. Garder, "Apictorial jigsaw puzzles: The computer solution to a problem in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-13, pp. 118–127, Apr. 1964.

[5] K. Hirota and Y. Ohto, "Image recognition in jigsaw puzzle assembly robot system (JPARS)," in Bull. Coll. Eng., Hosei Univ., Japan, pp. 87–93, Mar. 1986.

[6] K. Nagura, K, Sato, H. Maekawa, T. Morita, and K. Fujii, "Partial contour processing using curvature function—Assembly of jigsaw puzzle and recognition of moving figures," *Syst. Computing*,vol. 2, pp. 30–39, Feb. 1986.

[7] H. Blum, "A transformation for extracting new descriptors of shape," in *Proc. Symp. Models for Perception of Speech and Visual Form*,Weiant Whaten-Dunn, Eds. Cambridge, MA: MIT Press, 1967, pp. 362–380.

[8] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 170–179, May 1977.

[9] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 179–187, Feb., 1962.

[10] E. Wong and E. L. Hall, "Scene matching with invariant moments," *Computer Graphics, and Image Processing*, vol. 8, pp. 16–24, 1978.

[11] T. Nguyen and J. Sklansky, "A fast skeleton finder for coronary arteries," in *Proc. IEEE Int. Conf. Pattern Recog.*, Paris, France, Oct. 27–31, 1986, pp. 481–48.

[12] H. Blum, "A geometry for biology," in *Mathematical Analysis of Fundamental Biological Phenomena, Ann. New York Acad. Sci.*, vol. 231, pp. 19–30, 1974.

[13] You, Z. and K. Anil, "Performance evaluation of shape matching via chord length distribution," *Computer Vision, Graphics, and Image Processing*, vol. 28, pp. 185–198, Nov. 1984.

[14] J. L. Pfaltz and A. Rosenfeld, "Computer representation of planar regions by their skeletons," *Commun. ACM*, vol. 10, no.2, pp. 119–122, Feb. 1967.

[15] C. Arcelli, L. Cordella, and S. Levaldi, "From local maxima to connected skeletons," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, no. 2, pp. 134–143, Mar. 1981.

[16] U. Montanari, "A method for obtaining skeletons using a quasi-euclidean distance," *J. ACM*, vol. 16, pp. 534–549, 1969.

[17] B. Shapiro, J. Pisa, and J. Sklansky, "Skeleton generation from x,y boundary sequences," *Computer Vision, Graphics, and Image Processing*, vol. 15, pp. 136–153, 1981.

[18] L. Dorst, "Pseudo Euclidean skeletons," in *Proc. IEEE Int.Conf. Pattern Recog.*, Paris, France, Oct. 27–31, 1986, pp. 286–288.

[19] C. Lee, "Modified distance transform and linking algorithm for image skeletonization," in *Proc. SPIE*, vol. 415, Apr. 1983, pp. 147–154.

[20] B. Zvolanek, and C. Lee, "Image skeletonization for object position measurement," in *Proc. SPIE*, vol. 359, Aug. 1982, pp. 24–27.

[21] D. Lee, "Medial axis of a planar shape," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, no. 4, pp. 363–369, July, 1982.

[22] S. Peleg and A. Rosenfeld, "A min-max medial axis transformation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, no. 2, pp. 208–210, Mar., 1981.

[23] P. V. DeSouza, and P. Houghton, "Computer location of medial axes," *Comput. Biomed. Res.*, vol. 10, no. 4, pp. 333–343, Aug. 1977.

[24] G. Borgefors, "A new distance transformation approximating the euclidean distance," *Proc. IEEE Int. Conf. Pattern Recog.*, Paris, France, Oct. 27–31, 1986, pp. 336–338.

[25] P. Danielsson, "Euclidean distance mapping," *Computer Vision, Graphics, and Image Processing*, vol. 14, pp. 227–248, 1980.

An Algorithm for Point Clustering and Grid Generation

Marsha Berger and Isidore Rigoutsos

Abstract— **The paper describes a special purpose point clustering algorithm, and its application to automatic grid generation, a technique used to solve partial differential equations. Extensions of techniques common in computer vision and pattern recognition literature are used to partition points into a set of enclosing rectangles. Examples from two-dimensional (2-D) calculations are shown, but the algorithm generalizes readily to three dimensions.**

I. Introduction

This paper presents a special purpose clustering algorithm for the automatic generation of grids when solving partial differential equations using adaptive mesh refinement. Adaptive mesh refinement

Manuscript received April 25, 1990; revised January 26, 1991. This work was supported in part by the Air Force Office of Scientific Research under Contract No. AFOSR-86-0148, in part by the NSF Presidential Young Investigator Award ASC-8858101, in part by the Department of Energy Contract No. DEAC0276ER03077-V, and in part by Cray Research and Grumman Aerospace, who gave matching funds in support of the PYI award.

The authors are with the Courant Institute of Mathematical Sciences, 251 Mercer St., New York, NY 10012.

IEEE Log Number 9100401.

0018–9472/91$01.00 © 1991 IEEE