Finding Perceptually Closed Paths in Sketches and Drawings

Eric Saund, *Member*, *IEEE*

Abstract—Closed or nearly closed regions are an important form of perceptual structure arising both in natural imagery and in many forms of human-created imagery including sketches, line art, graphics, and formal drawings. This paper presents an effective algorithm especially suited for finding perceptually salient, compact closed region structure in hand-drawn sketches and line art. We start with a graph of curvilinear fragments whose proximal endpoints form junctions. The key problem is to manage the search of possible path continuations through junctions in an effort to find paths satisfying global criteria for closure and figural salience. We identify constraints particular to this domain for ranking path continuations through junctions, based on observations of the ways that junctions arise in line drawings. In particular, we delineate the roles of the principle of good continuation versus maximally turning paths. Best-first bidirectional search checks for the cleanest, most obvious paths first, then reverts to more exhaustive search to find paths cluttered by blind alleys. Results are demonstrated on line drawings from several sources including line art, engineering drawings, sketches on whiteboards, as well as contours from photographic imagery.

Index Terms—Contour closure, closed path, perceptual organization, Gestalt laws, sketch interpretation, line art analysis, graphics recognition.

1 Introduction

The Gestalt laws of perception have long recognized figural closure as one of the primary perceptual phenomena exploited by the human visual system. The detection of closure is fast, automatic, and seemingly effortless, and its role in building intermediate level descriptions is well-recognized [5], [7], [8], [10], [13]. The conventional explanation for why a visual system should seek closed or nearly closed contours is that coherent objects tend to be spatially compact and relatively uniform in surface appearance with respect to the surrounding background. Closure is thus viewed as a key cue for figure/ground segmentation. Abstracting away from cues about interior and exterior region properties, the typical psychophysical demonstration of figural closure employs straight or curved lines defining a region's bounding contour.

As a Gesalt phenomena, figural closure plays a significant role in the perceptual organization even of abstract figures that have no connection to coherent physical objects. This property of the visual system has come to be exploited across a culturally diverse set of conventions people have developed for representing information in graphic media. Referring to Fig. 1, compact closed or nearly closed paths in graphics, line drawings, and sketches can indicate, among other things, individual physical objects, conceptual objects, groupings or collections, logical or other abstract relations, emphasis, looping paths or circuits, symbols and characters (or fragments thereof), and tabular cells.

As a computer vision problem, the detection of compact closed figures in line art and sketches shares much in common with the counterpart problem in natural imagery. The document image domain presents, in many cases, a simpler image at the input level, where foreground markings can be separated from the background medium through relatively straightforward image processing, thresholding, and contour tracing processes. Following this stage, however, a sketch or drawing can present overwhelming clutter, complexity, and noise. See Fig. 2. Mathematically crisp and straightforward constructs from topology or geometry are unlikely to prove adequate to capturing the notion of perceptual salience, especially in the cases of casual, hand-drawn material. Instead, as is common with computational formulations for perceptual organization, the detection of closed figures in sketches and line art necessarily involves uncertainty, ambiguity, heuristics, and judgement calls. As always, we must exploit constraint. The generative processes underlying drawings differ from the imaging of physical objects and call for special consideration of how local image events give rise to larger scale coherent structure in this domain.

In this work, we adopt a framework common to previous computer vision algorithms: Starting from seed curve fragments, construct paths by tracing from one curve to another with which it is associated by end-to-end proximity. Typically, T-junctions, crossings, and higher order confluences of curve endings lead to ambiguity in which curve to choose for tracing. The key problem is to manage the explosive search of possible path continuations through junctions in an effort to find paths satisfying global criteria for closure and figural salience.

We offer two insights that motivate an effective algorithm. First, we observe that there are *two different* kinds of perceptually salient closed paths in drawings, those that tightly circumscribe regions, and those based on smooth continuation. This permits the establishment of two distinct sets of local preferences for tracing. One set of preferences, which we call *maximally turning preferences*, seeks the most tightly closed (or nearly closed) paths. A second set of preferences, *smooth continuation preferences*, seeks paths exhibiting maximal curvilinear smoothness.

• *The author is with the Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304. E-mail: saund@parc.com.*

Manuscript received 31 Oct. 2001; revised 23 May 2002; accepted 24 Oct. 2002.
Recommended for acceptance by D. Jacobs and M. Lindenbaum.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117706.

Fig. 1. A hand-drawn sketch exhibiting various roles for perceptually closed contour paths (see text).

Second, in order to fully exploit these local preferences, it is worthwhile to perform best-first search to grow contour paths *bidirectionally* from each seed contour fragment. This strategy doubles the opportunities for incipient paths to grow to closure, and mitigates garden paths that occur when local preferences misrepresent the trajectories of globally salient paths. The search process results in a redundant set of candidate closed paths due to different seeds finding similar paths. A final consolidation step merges equivalent candidates.

The present algorithm is fast enough and robust enough to serve as a key module in a larger system for perceptual analysis of document images under development at our laboratory. One application of this work is perceptually-supported editing of "rough documents," that is, documents that involve handwritten sketches, doodles, annotations, or otherwise do not obey the constraints of formatted text and graphics [19].

The paper proceeds as follows: Section 2 reviews previous work. Section 3 further explores the phenomenon of figural closure and defines the goals for our algorithm. Sections 4, 5, and 6 describe the algorithm itself, including image processing and data preparation, definition of local and global criteria, search, and consolidation. Finally, Section 7 presents results and Section 8 evaluates key parameters of the algorithm.

2 Previous Work

Early efforts in Computer Vision to detect perceptually compact region structure by spatial aggregation sought partially closed regions, or local convexities. "Local Rotational Symmetries" [6] are proposed by combining local hypotheses of centers formed by edge groups. "Partial Circular Regions" [15] are found by clustering pairs of coarse-scale boundary contours forming "Primitive Partial Regions." These are both multiscale approaches. "U-Structures," [11] suitable for detecting buildings in aerial images, comprise sets of parallel edges joined by a third line on one end.

More recently developed algorithms are based on tracing paths through a graph of linked edgel or contour elements, where search is managed through combinations of local and global constraints and figural goodness criteria. Table 1 compares several approaches. Huttenlocher and Wayner [7] seek strictly convex groups in line segments derived from edge data. Severe restrictions constrain the number of path options at every endpoint, which in turn limits the set of salient regions which potentially can be found. Jacobs [8] seeks strictly convex paths through extended edge data, and incorporates a global criterion on gap size to manage a backtracking search guaranteed to return every convex set meeting specified criteria. Elder and Zucker [5] use locally n-best continuations of edgels based on smoothness and

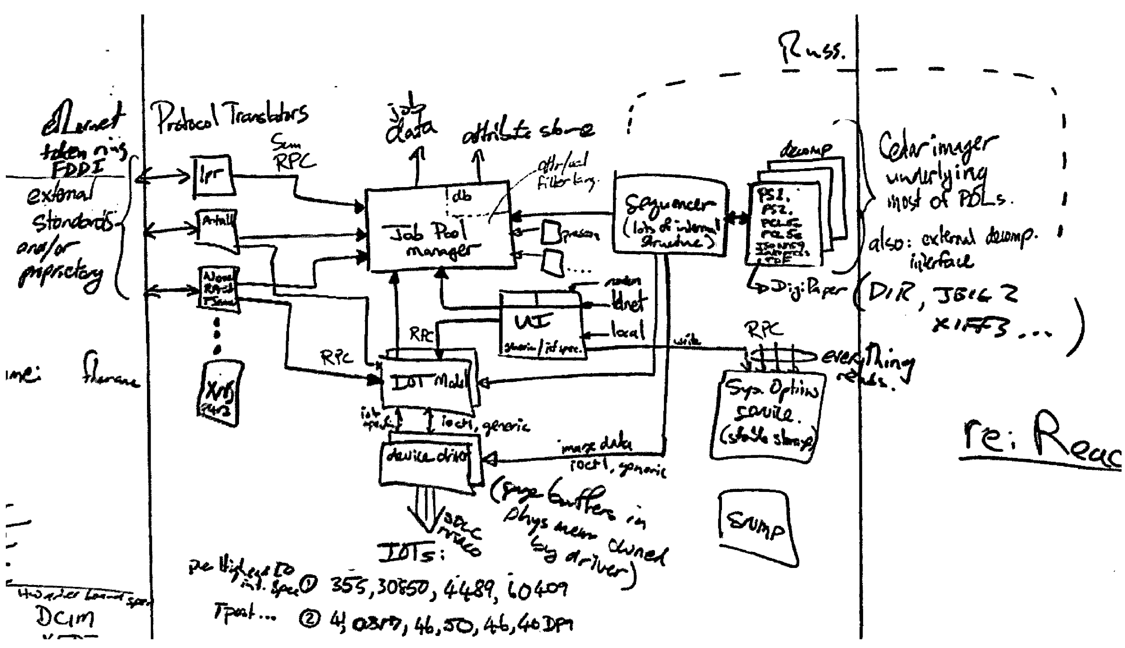


Fig. 2. Portion of a typical complex diagram scanned from large whiteboards using an experimental optical whiteboard capture system in routine operation at our research center [17]. The dimension of this section was drawn at approx. $6' \times 4'$.

TABLE 1
Comparison of Figural Closure Detection Algorithms

	data type	local criteria	global criteria	search method
Huttenlocher & Wayner	linear edge contours	proximity, pairwise convex, best only	strictly convex	connected component aggregation
Jacobs	linear edge contours	pairwise convex	strictly convex, gap/figure length ratio	backtracking
Elder & Zucker	edgels	proximity, smooth or corner	strictly closed	dynamic programming
Casadei & Mitter	curvilinear contours	proximity	strictly closed	recursive path concatenation
Mahamud et al	edgels	smoothness	strictly closed	graph enhancement, then connected component
present paper	curvilinear contours	proximity, smoothness, turning	compact (loosely convex), mostly closed	best-first, bidirectional

corner probabilities, and perform dynamic programming to find optimal strictly closed paths of any shape. Casadei and Mitter [3] construct strictly closed paths on curvilinear contours using recursive composition. Their focus is on efficiently eliminating redundant candidates. Mahamud et al. [10] form a local affinity matrix between edgels, and, then, trace connected components after using a linear algebra technique to enhance links between edgels falling on strictly closed contours.

The present algorithm operates on curvilinear contours and uses two kinds of local preferences, plus flexible global criteria to manage a bidirectional best-first search with backtracking. Because previous algorithms for closure detection are limited to finding either strictly closed or strictly convex paths, they are not suitable for identifying perceptually salient structures which may have significant open ends, and/or may be visually compact, but not strictly convex. We require effective solutions for curvilinear contours, which reduce the data volume from primitive edgels, yet are more expressive than straight-line segments. Finally, the present algorithm is designed to be tailored to the constraints of line art data, although it may nonetheless prove useful for photographic data as well. Canham et al. [2] perform graph search on hand-drawn line art with specific attention to tracing smooth continuations versus turning corners at junctions, but with the goal of finding specific shapes instead of closed paths in general.

3 Phenomenon and Goals

The phenomenon of perceptual closure is commonly interpreted to refer to a collection of curvilinear contours clearly distinguishing an "inside" region from a surrounding "outside." These contours may arise from lightness, texture, color, motion, or other edge detection processes, or they may occur as chains of local or extended contrast ridges. Fig. 3 presents several figures that help to define the

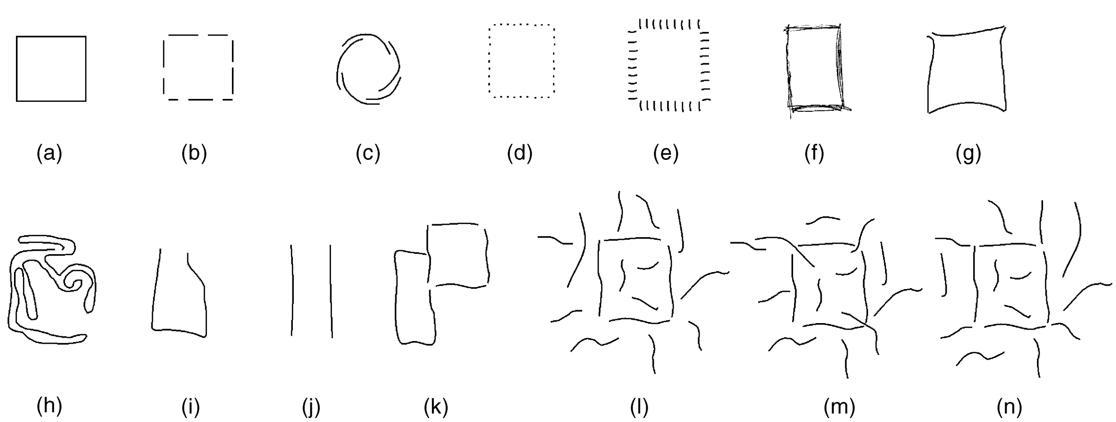
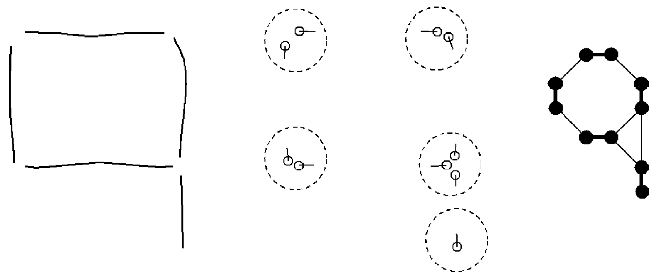
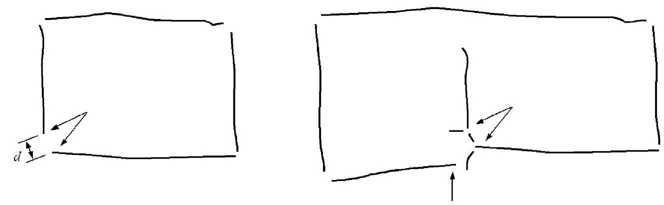


Fig. 3. Variations on closed figures. See text for discussion.



(a) (b) (c)

Fig. 4. Steps in data preparation. (a) Input figure. (b) Curve ends clustered by proximity. (c) Junction graph. Nodes are represented by barbells because every node represents a curve with two ends. Later steps label every link with corner and alignment scores, then with junction preference scores.



(a) (b)

Fig. 5. (a) To form a link between this pair of curve ends in the junction graph, the search for nearby ends must extend for a distance of at least d. (b) The distance d encloses a tangle of curve ends, but not necessarily all the ones relevant to the local trace through curve fragments participating in a salient closed figure (lower arrow). An adaptive clustering algorithm collects all curve ends in the vicinity.

boundaries of the phenomena and permit us to constrain the goals of an algorithm.

Fig. 3a is an ideal reference figure. Fig. 3b shows that contours forming a perceptually closed region need not be connected, but may have sizable gaps. In Fig. 3b, the proximity of nearby endpoints permits easy jumping of gaps and it is easy to imagine how to fill in missing sections of contour. By contrast, the contours in Fig. 3c overlap one another and clear end-to-end linking appears less feasible. Figs. 3d, 3e, and 3f resemble Fig. 3c in that, contour evidence participating in closed figures can be locally discontinuous or noisy. We find it acceptable that our algorithm fails on these examples, including Fig. 3f, which exhibits line quality commonly found in sketches and line drawings. In our view, a more sophisticated, intermediate scale process should operate to discover image contour features defined by essentially texture qualities. For the present purposes, we concentrate on continuous or broken paths delineated by a simple, single-width curvilinear lines, but not fuzzy or textured contours.

Fig. 3g illustrates that perceptually salient closure regions need not be convex, but can include a significant degree of concavity. We equate degree of concavity with perceptual compactness. Fig. 3h is dominated by concavities. Whether the squiggle forms a thin closed figure is not immediately apparent without inspection, while many of its other features are much more significant. We provide a tunable parameter for specifying the degree of concavity deemed salient.

Fig. 3i shows that, in order to be perceptually salient, a closed region need not be truly closed but can have a significant open end. Fig. 3j pushes the point. An "inside" region can be defined as the region between the lines but in most contexts this does not appear to give rise to strong perceptual salience as a closed figure.

Fig. 3k shows that multiple closed regions can share the same segment of contour.

Figs. 3l, 3m, and 3n illustrate perceptual closure of contour paths in the presence of clutter. Clutter that respects the inside/outside relationship is less disruptive than clutter that breaks it. Clutter that attempts to disrupt the closed path by offering alternative paths with better smoothness (good continuation) does not seem to matter.

From consideration of these prototypical cases, as well as real-world examples such as Figs. 1 and 2, we identify desired behavior for an effective algorithm. A good algorithm must find nearly closed paths in the presence of gaps and clutter. The paths must allow some degree of concavity. It must find globally salient paths among contours whose local evidence for directions of contour continuity is misleading. It must find multiple paths that share common contour support. Finally, it must efficiently identify the subset of closed paths that are perceptually significant among the exponential number of valid possibilities occurring in highly connected figures such as a grid.

4 Data Preparation

4.1 Junction Graph

Our starting point is a collection of simple curve fragments. These are relatively straight curvilinear path segments bounded by free ends, corners, or junctions. These may be obtained from source data by any number of means, including local linking of edge or ridge detector outputs, thresholding, thinning, and tracing of scanned line drawing data, or tracking a pen or stylus, then detecting corners and crossings in the resulting digital ink.

Closed paths are to be constructed by tracing sequences of curve fragments linked roughly end-to-end. The set of closed paths that could potentially be entertained by the algorithm is governed by the sets of candidate links between pairs of curve fragments identified in the data preparation stage. These links form a graph whose nodes are the curve fragments themselves. This is the graph that must be searched for chains representing significant, closed contour paths. See Fig. 4.

An engineering tradeoff is required in choosing suitable links to form the junction graph. On the one hand, to form exhaustive links between both ends of all curve fragments would be sure to include all possible closed paths. On the other hand, only fragments proximal to any given curve fragment end are likely to lie on a common perceptually significant closed path, while exhaustive linking increases the search space to unacceptable proportions. Fig. 5 presents dual situations which are both common. In the left half of this figure, two curve fragment pairs must link across a spatial gap of size d, while in the right half of this figure the radius d encompasses a tangle of complex line work containing seven other curve fragment ends. It is important that candidate links be formed between all curve fragment ends within such a tangle because a priori it is not known which sets of fragments may prove to participate in a good path.

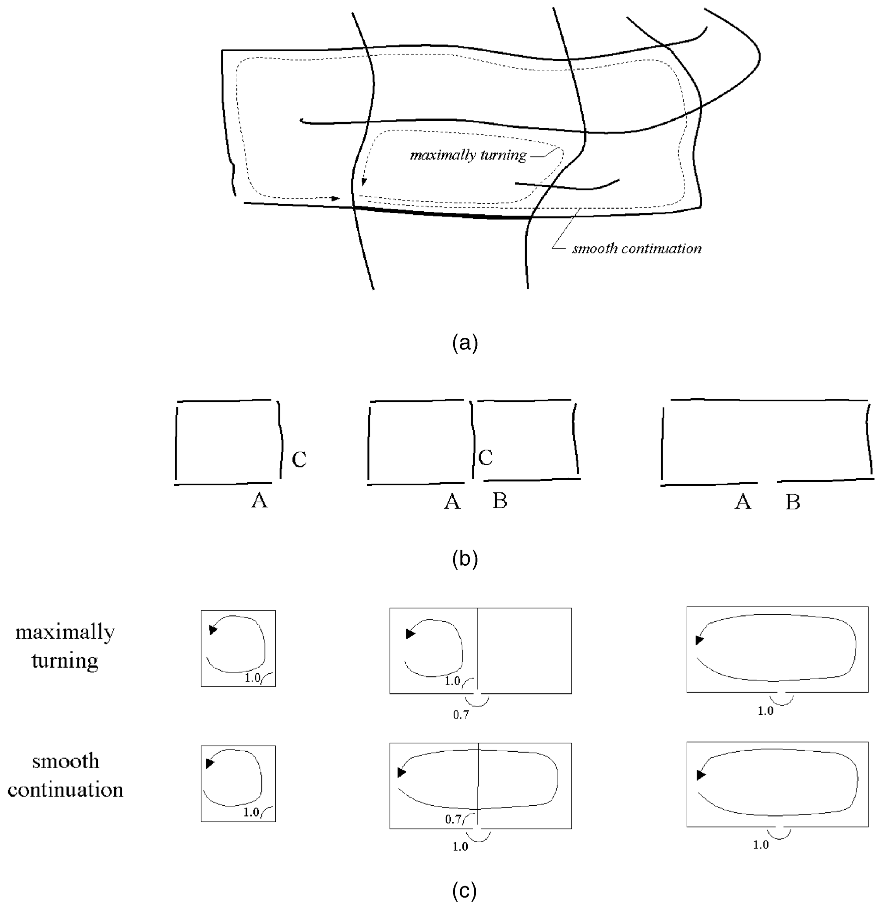


Fig. 6. Two kinds of perceptually salient figural closure. (a) A maximally-turning closure path traces the smallest figure possible, while a smooth continuation closure path prefers smooth continuation traces through junctions. Other closed paths through the seed (thick contour fragment) are perceptually insignificant. (b) Simpler figure illustrating use of junction preference scores. (c) Numbers indicate junction preference scores, under both kinds of path preference, for a path traced between pairs of curve fragments indicated by arcs. Any score below 1.0 indicates a penalty. Arrows indicate minimal penalty paths.

An adaptive link forming process proves advantageous here. Let us denote by S_E the set of curve fragment ends forming a link with seed end E chosen from a seed pool of all curve fragment ends. In our implementation, we choose S_E as the transitive closure of ends occurring near to one another. An iterative clustering algorithm adjusts the threshold distance to limit size of this set of local mutual links.

The final step of data preparation is to score candidate links in the junction graph according to properties of local geometry. Each pair of linked curve ends is evaluated for the degree to which their parent curve fragments form a "corner" relation to one another, or an "alignment" relation, according to heuristic formulae on relative orientation, etc., [4], [12], [16]. Any pair of curve ends not meeting a threshold value on both the corner and alignment scores is deemed to represent nonlinking curve ends and is removed from the junction graph.

5 Closed Path Quality Criteria

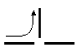
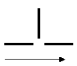
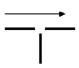
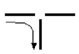
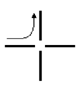
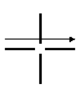
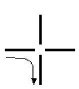
5.1 Local Junction Preferences

The strategy for searching the junction graph for chains representing closed paths is based on choosing nodes (curve fragments) one at a time, then trying to grow a closed path by extending the chain in both directions simultaneously. This search is managed through the use of local criteria for prioritizing the order in which the chains are grown by extending or backtracking. These local criteria reflect domain knowledge about the ways in which curve fragments tend to compose into larger closed paths in the line art/hand-drawn sketch domain we are focused on. Growing chains are evaluated by global figural goodness criteria to decide when search should be terminated or redirected according to success and failure conditions.

Fig. 6 illustrates the fundamental observation that offers leverage in this domain. One kind of perceptually salient closed path is found as a maximally turning path, or one that defines the smallest region enclosed by any contour of the path. In this case, distinct bounded regions are the primary objects of interest. In document images, objects found within a maximally turning region tend to bind together visually, for example, checks found within the cells of a table as shown in Fig. 1.

A second kind of perceptually salient closure is found as a maximally smooth path, where choices of directions through junctions obey the law of good continuation. In line art, these paths normally reflect a single motion of the pen and contour junctions merely reflect "accidental" crossings of distinct pen-stroke objects.

TABLE 2
Table of Junction Preference Scores

junction configuration	max turning CCW	max turning CW	smooth continuation CCW	smooth continuation CW
	1.0	0.7	1.0	0.7
	0.7	1.0	0.7	1.0
	1.0	0.5	0.9	0.5
	0.9	1.0	1.0	1.0
	1.0	0.9	1.0	1.0
	0.5	1.0	0.5	0.9
	1.0	0.5	0.9	0.5
	0.5	1.0	0.5	0.9
	1.0	0.5	0.6	0.5
	0.8	0.8	1.0	1.0
	0.5	1.0	0.5	0.6
	1.0	1.0	1.0	1.0

Columns of scores are presented for path searches attempting to find a closed figure proceeding in the counterclockwise (turning leftward) or clockwise (turning rightward) direction, proceeding through the junction from the left (shaft of the arrow). Columns are presented for both maximally turning and smooth continuation paths. For each row, the arrow indicates a possible path trace through a configuration of corner and alignment links. A score of 1.0 represents maximal preference.

An algorithm seeking Gestalt structure in line drawings must detect both kinds of closure. Note that causal underpinnings for the law of good continuation are found in the physical world as well, for example, due to object occlusion. Therefore, the bifurcation of closure types and the usefulness of detecting both types of closure may well apply to natural scenes in a similar way. We leave this exploration for future work.

To enable a search for both kinds of contour closure, we analyze local contour junctions and score each contour pair entering and exiting the junction for its support of a maximally turning path versus a maximally smooth path. We call these *junction preference scores*. These scores make use of the pairwise contour corner and alignment scores mentioned in Section 4, but they must take into account the local context of all of the contour ends linked to either member of a given pair, not just the geometry of that pair itself. Junction preference scores range from 0 (minimal preference) to 1 (maximal preference). As discussed in Section 6.3, this definition of score permits preferences to accumulate along a candidate trace by a process of attenuation simply by multiplying local preference scores along the way.

Fig. 6 illustrates how this works. Contour ends A-B form a good continuation while ends A-C form a good corner, but a poor smooth continuation. When only the pair A-C is present (left column) or only A-B is present (right column), both maximally turning paths and maximally smooth paths are forced to proceed through these respective contour end pairs. There is no penalty for a maximally closed path going straight, nor for a maximally continuous path making a turn in its preferred direction when there is no other alternative. In the middle column, however, where both junctions are present, pair A-B is scored to reflect a preference for a maximally smooth path while A-C is scored to reflect a maximally turning path.

This kind of analysis permits us to build a table (Table 2) of local junction preference scores that rate end pairs according to their corner and alignment scores in the context of other end pairs occurring at that junction. The table contains four columns. Two columns apply to maximally turning paths, and two to smooth continuation paths. Each set contains scores for paths proceeding to construct a closed path turning in both the clockwise and counterclockwise directions. The scores contained in this table were arrived at on a purely intuitive basis by following the logic exemplified in Fig. 6. Note, for example, that, when several path directions are possible, maximally closed paths prefer to turn in their preferred direction, and are increasingly penalized for choosing directions deviating from that. Maximally smooth paths prefer to keep going straight, they are somewhat willing to turn in their preferred direction and they resist turning in their counter-preferred direction. The exact values in this table are not critical, but varying the ordering of values results in less efficient search or sometimes failure to detect the best closed figure(s) containing a given seed contour. For clarity, the table depicts only corners, crossings, and T-junctions. Y-junctions and junctions involving more than four curve ends are accounted for in the same table, as all pairwise end configurations are labeled as "alignment" or "corner."

Fig. 8 shows how this table comes into play in handling intermediate and ambiguous cases that occur very often in our target domain. In order to trace a maximally turning path, tracing must sometimes choose a good continuation path through a local junction, or even a turning direction opposite that defining the region as a whole. Conversely, in order to trace a maximally smooth path, tracing must sometimes neglect a local good continuation through a junction and take a sharp turn instead. These situations are contended with in the search procedure described in Section 6.

5.2 Global Figural Goodness

While local junction preferences present local criteria for figural goodness that can help guide search, the goal is to find globally salient closed regions. For this reason, we need to define a global figural goodness criterion that indicates when

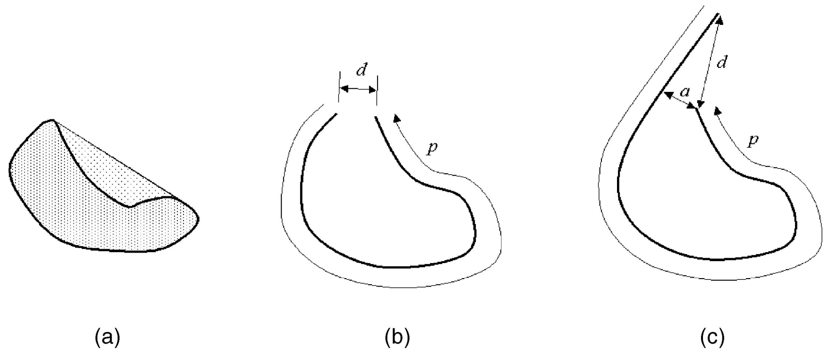


Fig. 7. Factors entering into a measure for global goodness of a candidate closed path. (a) Compactness measured as ratio of enclosed area to area of convex hull. (b) Endpoint distance. (c) Degree to which one end of the path extends beyond the other.

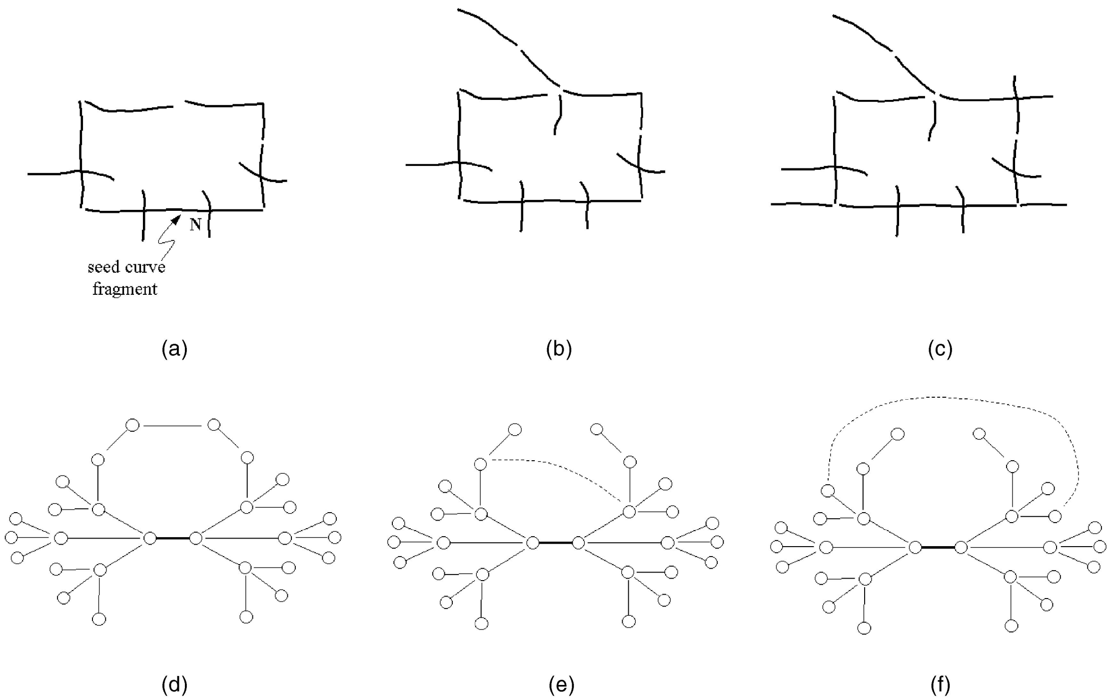
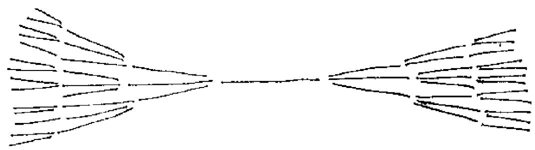


Fig. 8. (a), (b), and (c): Closed paths posing successively greater difficulty for bidirectional search from a seed curve fragment. (d), (e), and (f): Counterpart abstract search graphs (these are abstract illustrations of qualitative behavior and do not map literally onto the example figures above). Locally, best search paths are depicted as the branches bending most toward the other side. In (a) and (d), the target closed figure can be found by extending a branch from just one end of the seed (e.g., the north end). In (b), and (e), the target closed figure can be found by pairwise testing of nodes on the best branch search paths (dotted line). To find the closed path in figures such as (c) and (f) requires expanding locally nonpreferred nodes. The situations depicted here refer to smooth continuation paths. A similar set of figures could be drawn illustrating search for maximally turning closed paths.

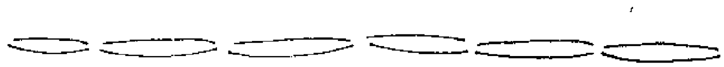
a good closed contour has been found from a given starting seed.

The global figural goodness measure we employ was constructed heuristically through evaluation of hundreds of examples encountered in our image domain. We identify three criteria, all of which are fully satisfied by an ideal closed contour path. These are expressed as three component terms cast as values ranging from a maximally good value of 1 to a minimal value of 0. Their conjunction is expressed by combining them multiplicatively. Fig. 7 illustrates the component criteria we found significant:

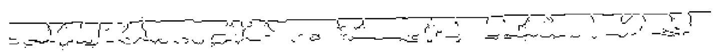
- **Compactness term**. Perceptually salient closed figures tend to be compact. They may contain concavities, but concavities should not dominate their form. We estimate this property by the ratio of area of the figure to the area of its convex hull. This term can be relaxed somewhat to permit increasingly large concavities in the figures found.
- **Endpoint-distance term**. This term gauges the degree to which a path completely closes on itself versus leaving a gaping open end. We estimate this property as $1 - d_e/p$, where d_e is the distance between the endpoints of the path and p is its length.
- **Non-end-nearest-approach term**. This term penalizes paths one of whose endpoints terminates near the body of the path instead of near the other end. See Fig. 7c. Measure the nearest approach of each endpoint to the portion of the curve distal from that endpoint. Call the minimum of these two measurements a. Then,



(a)



(b)



(c)

Fig. 9. Images giving rise to the potential for explosion in search. (a) Even with safeguards described in Algorithm A, search for closed paths seeded from the center curve fragment involves creation of 208 TreeNode structures and potential testing of 757 paths (most of which are filtered by virtue of their ends pointing in opposite directions). (b) The number of smooth continuation paths is 2^N, where N is the length of the chain. A search over these is limited to a linear number by step A.2.2. (c) Curve fragments arising as artifacts of preprocessing the borders of a scanned whiteboard image. In practice, we find that pathological search conditions arise from "image noise" of this sort more often than actual drawn material.

the non-end-nearest-approach term is taken as $min[1, a/(d_e - p/c)]$, where d_e is the distance between the endpoints, p is the curve length, and c is a constant which depends on image resolution.

Under different choices of parameters in the junction preference table, it could become beneficial also to test winding number or for self-crossings, neither of which is representative of the kind of perceptually salient closed paths we are seeking.

Note that the global goodness measure assesses figural "salience" only for a curve in isolation. The true visual salience of a curve depends of course on what else is around (as well as visual task). An ideally scoring rectangle is in fact nearly invisible as an independent object when embedded in a grid. Thus, the goal of our algorithm is not to return as many paths as possible having a high global goodness measure. Rather, the search algorithm itself plays a crucial role in delivering visually salient figures by searching for high global goodness paths under the direction of local preferences. This strategy serves to exclude spurious high global-goodness paths embedded within dense linework.

6 Bidirectional Search Procedure

6.1 Design Principles for the Search Procedure

Search for closed paths is based on growing each candidate path in both directions starting from a seed contour fragment. The strategy for our search procedure is informed by the following design principles:

1. Pick up the easy structure first and fast.
2. Do not repeat work already done.
3. When the going gets tough, pursue the most promising leads first.
4. Construct a wealth of hypotheses, then combine or prune later.

These principles reflect the goal of our algorithm, which is to achieve fast and robust detection of closed contour paths primarily in sketches and line drawings, using currently available computing hardware and programming languages. The opposing goals of speed and thoroughness of search mandate engineering tradeoffs. Alternative design decisions may be appropriate for algorithms attempting different objectives.

Once launched from a seed fragment, the search procedure may return zero, one, or several closed path candidates. Search from different seeds may return identical or very similar paths. This accords with design principle 4, but violates design principle 2. Section 6.5 describes how redundant paths are consolidated.

To minimize the number of redundant paths found, we employ a strategy to eliminate seeds by judiciously choosing and marking contour fragments. Most salient closed paths can be found by searching from a small subset of the potential seed fragments forming nodes in the junction graph. This is discussed in Section 6.4.

The core bidirectional search procedure itself adheres to design principles 1 and 3. By proceeding best-first, it rapidly finds the most perfectly formed closed paths in time linear in the number of contour fragments contained in a path. Then, successively less preferred avenues are tried to discover closed paths that occur among increasingly misleading garden paths.

6.2 Search Tree Representation

Figs. 8a, 8b, and 8c illustrate a series of closed paths whose detection requires an escalating amount of effort. Each run of the search procedure is executed with one of four parameter settings governing closure type and turn direction. Search will seek either maximally turning paths or smooth continuation paths and it will attempt to find paths closing on either one side or the other of a given seed fragment. Let us label one end of the seed fragment, "North." Then, paths on the west side of this fragment will be sought by searching for paths proceeding counter-

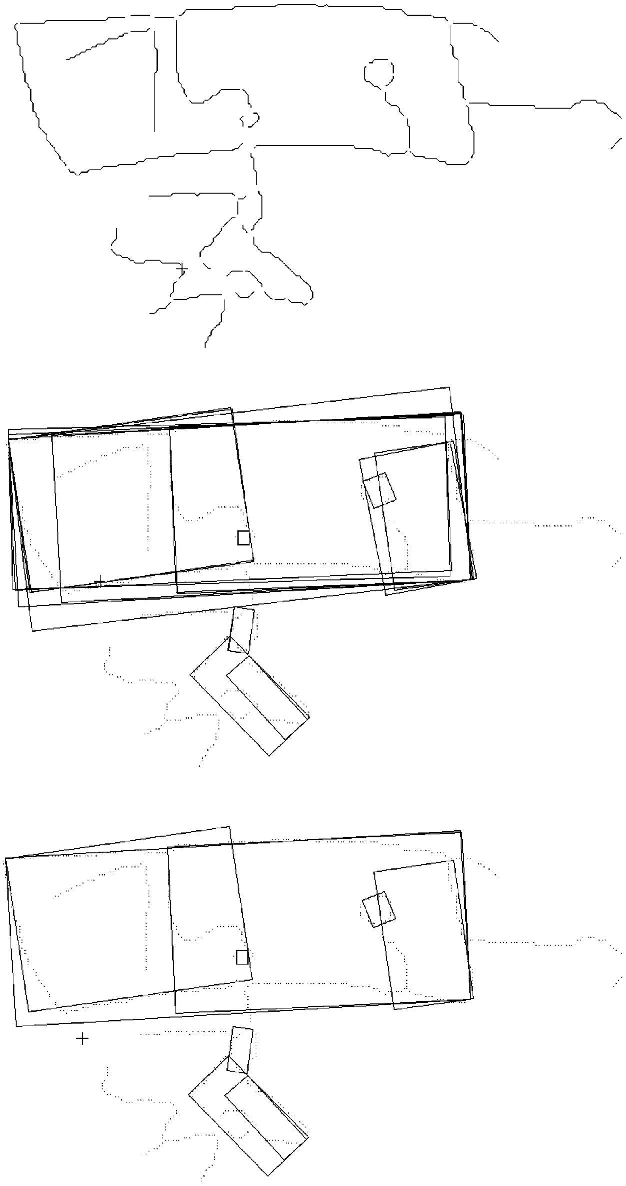


Fig. 10. Consolidation of redundant paths using pose clustering. (a) Input curve fragments. (b) Poses (oriented bounding boxes) of the 17 paths found by the search process (five maximally-turning and 12 smooth continuation). (c) The nine closed paths returned after consolidating these (see text).

clockwise heading north from the seed, and clockwise heading south from the seed. These parameter settings of the search direct which column of Table 2 to use in extending the growing contour according to local junction preferences. For the sake of the following discussion, let us refer to Fig. 8, assuming that search is seeking smooth continuation paths lying to the west of the seed contour.

The quality of the search spaces for paths of three degrees of difficulty are depicted abstractly in Figs. 8d, 8e, and 8f. The search tree has two roots, one extending from the north of the seed curve fragment, and one extending from the south end. In general, the search consists of expanding the tree on one or both sides to generate new nodes. A node in the search tree corresponds to a curve fragment, and the branches emanating from a node correspond to the choices among path continuations on the distal junction of that curve fragment. Any north-subtree/south-subtree pair of nodes thus represents a path of curve fragments linked-end-to-end through the junction link graph. To any such pair of nodes we can apply the closed path goodness measure.

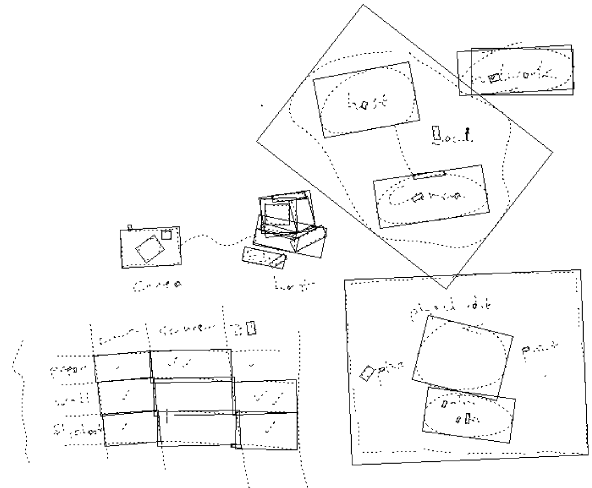


Fig. 11. The 62 closed paths returned by the closed path finding algorithm on 467 curve fragments extracted from Fig. 1. Paths are depicted by their oriented bounding boxes. Processing time: Three seconds.

The abstract search trees are drawn in Figs. 8d, 8e, and 8f, such that branches are ordered by local junction preference, the most preferred junction trace being the one bending the most toward the other half of the tree. For example, for a smooth continuation closed path search, the branches emanating from each node on the north half of the tree would be arranged left to right in order of local junction preference score as looked up in column "smooth continuation CCW" of Table 2.

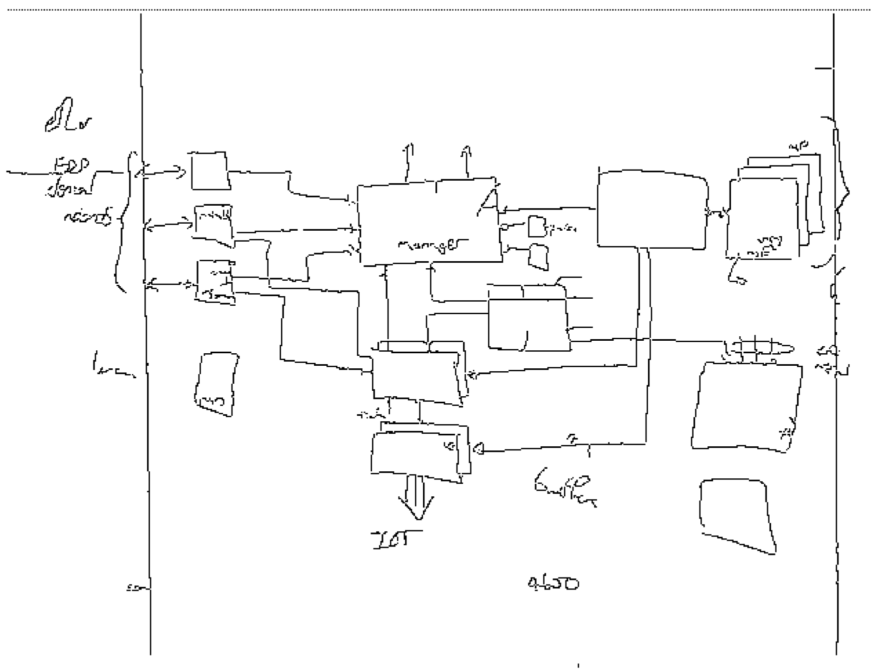
6.3 Search Procedure

The closed path of Fig. 8a is well-behaved with respect to the smooth continuation path criteria. It is fully closed in the sense that all participating curve fragments are linked end-to-end all the way around the path. Furthermore, the smooth continuation local junction preference of each of the curve fragment ends leads to a next fragment that is in fact on the correct closed path. (This figure is not well-behaved with respect to maximally closing paths because local turns lead to dead ends.)

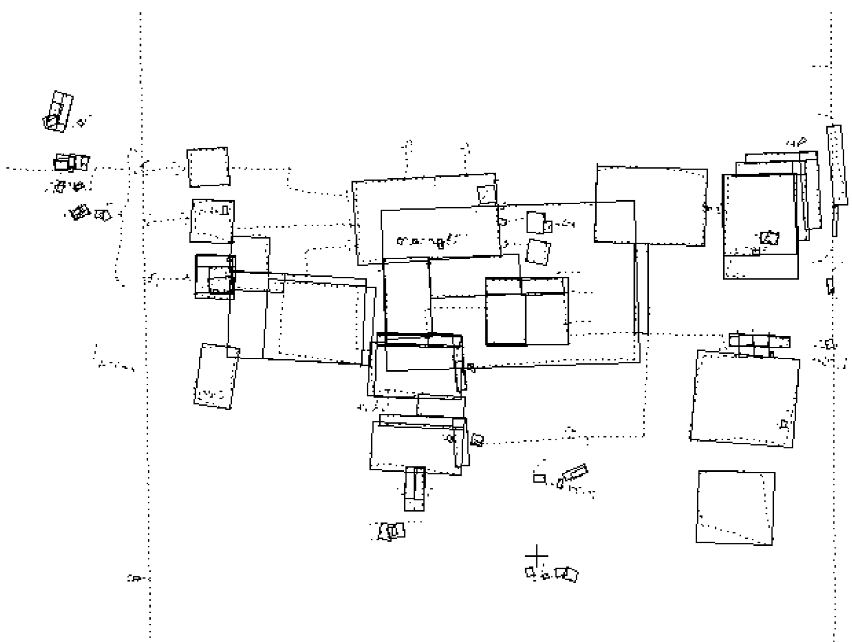
Correspondingly, the search path for this closed figure is quite simple as shown in Fig. 8d. In fact, the closed path can be found by expanding nodes from just one end of the bidirectional tree until the path closes on itself. (For simplicity, not all of these branch choices are illustrated in Fig. 8d).

In cleanly drawn line drawings, a significant fraction of high-goodness closed paths are of this type. The search procedure is designed to find these rapidly by growing one branch first along the best-first path.

The closed path of Fig. 8b is somewhat more difficult to detect than Fig. 8a. Its counterpart search tree quality is illustrated in Fig. 8e. The dotted line links the pair of tree nodes representing the best quality closed path available between the north and south subtrees. This closed path cannot be found by best-first extension from either the north



(a)



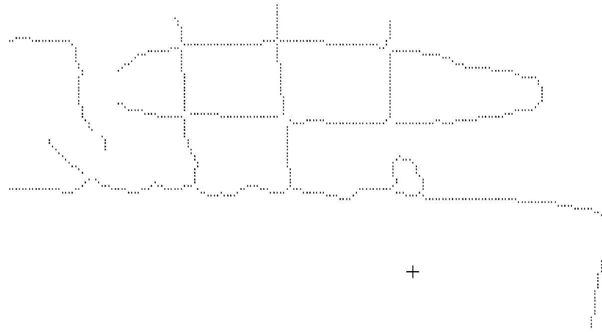
(b)

Fig. 12. (a) Input data: 702 curve fragments. (b) Results: 125 closed paths. Processing time: Eight seconds.

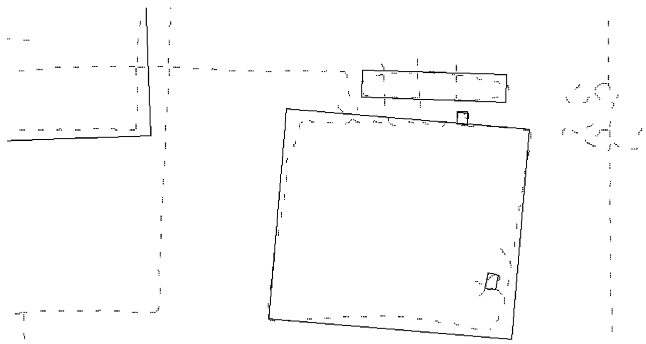
or south end of the seed alone because it is not fully linked; both north and south subtrees must be expanded to generate the required nodes representing the best goodness closed path. Nodes from the North and South branches have to be tested pairwise.

Closed paths of a nature illustrated in Fig. 8c require even more exhaustive search to detect. Not only do these paths require expansion in both north and south directions from the seed curve fragment root, but they require exploration of branches in these trees that reflect locally nonpreferred choices in trace direction. Observe the garden paths where local smooth continuation criteria lead away from the closed figure. In Fig. 8f, the dotted line shows that the target closed path is represented by node pairs found somewhere in the bowels of the search subtrees. We must take care to mitigate the potentially exponential explosion of search paths.

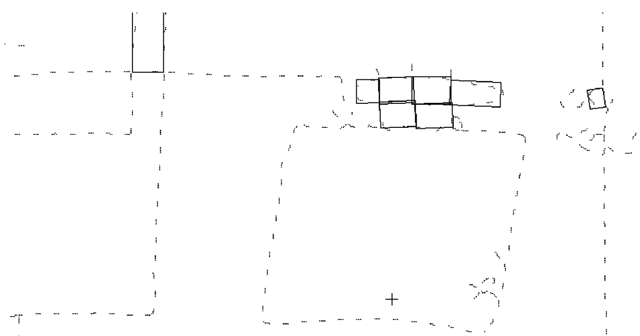
To do this, the search strategy is based on straightforward, best-first search with pruning. As with A* and other search techniques, we require an estimate of the quality of



(a)



(b)



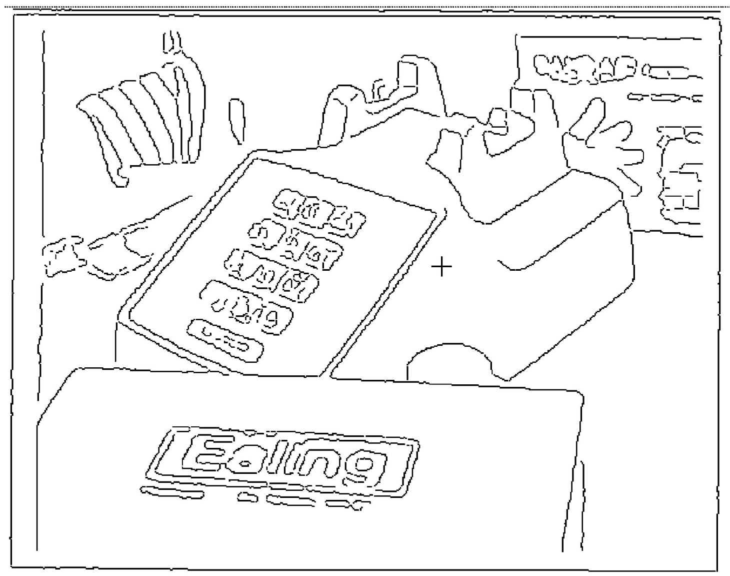
(c)

Fig. 13. (a) Detail from Fig. 12, showing smooth-continuation paths (b) and maximally-turning paths (c).

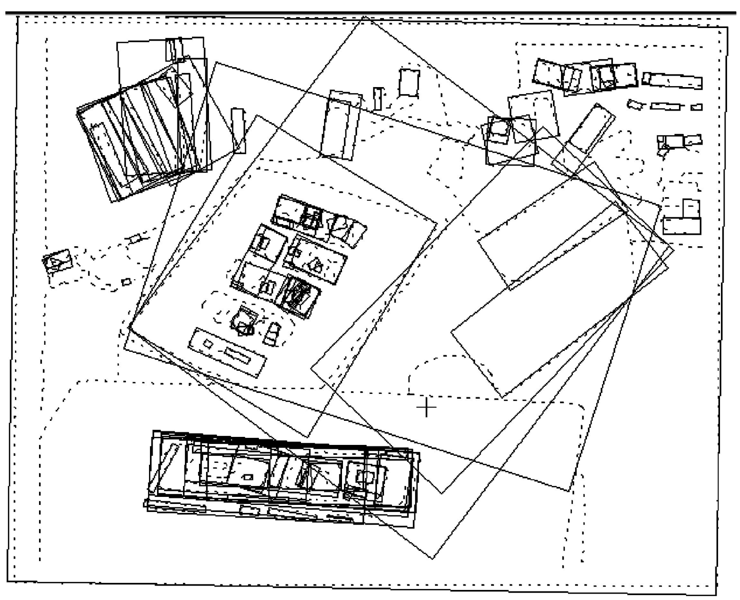
incomplete paths. In our case, this obtains from local junction preference scores as described in Section 5.1. We assume that globally better quality closed paths will, in general, be those reflecting preferable choices in local tracing through curve fragment junctions, according to the closed path type sought (maximally-turning or smooth-continuation). Because junction preference scores in Table 2 are scaled from 0 (unpreferred) to 1 (most preferred), we can combine these by multiplication. Proceeding outward from the root, each north or south partial path represented by an expanded node in the subtree suffers attenuation in estimated quality for every nonpreferred direction choice along the way. This quality estimate is used both to select which node to expand next, and to prune unpromising nodes.

Data Structure: TREENODE: Indicates a node in the search tree. A TreeNode Contains the following fields:

- Parent-TreeNode: pointer to the parent TreeNode (toward the root).
- Outward-curve-fragment-end: the curve-fragment end in the junction graph corresponding to this node in the search tree, distal to the seed curve fragment.
- Cumulative-junction-preference-score: The product of local junction preference scores tracing back to the root node (seed curve fragment).
- Child-curve-fragment-list: A list of curve fragment ends linked to this TreeNode's outward-curve-fragment-end, along with each's local junction preference score. This list is maintained in descending order of junction preference score, so that if the current TreeNode is expanded further, the most preferred continuation of the path will be expanded first.



(a)



(b)

Fig. 14. (a) Input data: 534 curve fragments. (b) Results: 169 closed paths. Processing time: Six seconds.

- Best-child-score: The cumulative junction preference score of the first child in child-curve-fragment-list. This is simply the product of this TreeNode's cumulative-junction-preference score and the local junction preference score of the best extension from this TreeNode.
- Depth: The depth of this TreeNode in the search tree.

The search algorithm maintains two lists of TreeNodes that have been expanded so far, one set branching form the north end of the seed fragment, the other set branching from the south end. These lists of TreeNodes are maintained in descending order of best-child-score.

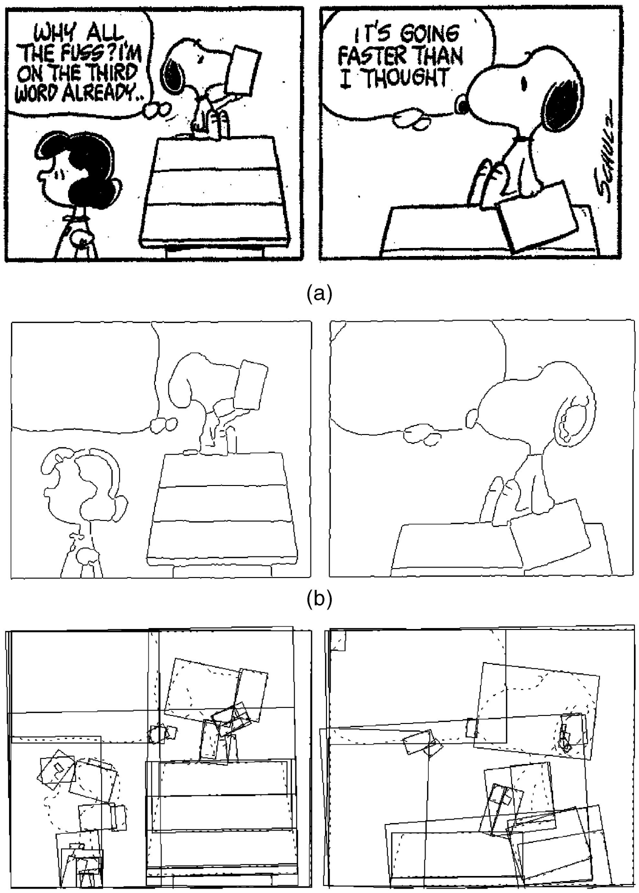
Algorithm A: Search

A.1. Initialize the north and south TreeNode lists with a TreeNode representing the north end and south end of the seed curve fragment, respectively.

A.2. Loop while the best-child-score of the first member of the north TreeNode list (or south TreeNode list) is greater than a threshold value, T_{cumatt}. We use $T_{cumatt} = .6$ for smooth continuation search and $T_{cumatt} = .9$ for maximally turning search. (See discussion in Section 8.)

A.2.1. Expand the best scoring TreeNode from the north (south) TreeNode list. This returns a new TreeNode representing the extension of the search path down the most preferred path step beyond the curve path represented by the expanded TreeNode.

A.2.2. Compare the newly created TreeNode with every other TreeNode on the north (south) TreeNode list. If this new TreeNode's outward-curve-fragment-end matches that of any other TreeNode



(c)

Fig. 15. (a) Input image. (b) Curve data: 265 curve fragments. (b) Results: 73 closed paths.

on this side, then abandon the expansion of this TreeNode and proceed to Step A.2.1.

A.2.3. Apply the global curve path goodness test between this TreeNode and each TreeNode on the south (north) TreeNode list. Before applying the global goodness test to a path, we use a pretest on the relative orientation of the two TreeNodes' outward-curve-fragment-ends. We discard immediately paths whose ends are directed away from one another.

A.2.4. If the goodness measure achieves a threshold value T_{accept}, then accept this path as a candidate closed path.

A.2.5. If the path achieves a more strict threshold value T_{done}, then terminate the search and return the candidate closed paths found so far.

A.2.6. Add the newly created TreeNode to the north (south) TreeNode list, in its proper location to keep this list sorted in decreasing order of best-child-score.

A.2.7. If the newly created TreeNode exceeds a predetermined limit on search tree depth, proceed to step A.2.1.

A.2.8. If the north (south) TreeNode list length exceeds a maximum size, then remove its last element (the TreeNode with the smallest best-child-score).

A.3. Return the list of candidate paths accumulated.

Note the use of two thresholds on the acceptance of candidate closed paths. A lower threshold T_{accept} is used in accumulating closed paths of good enough quality that they should be reserved for further consideration, but the search for better paths continues. Paths meeting a higher quality threshold T_{done} are deemed good enough to declare success and terminate the search for any more paths from this seed (under these turning/continuation/west/east search parameters). The settings of these parameters are to be determined according to any particular application's data and speed/quality tradeoff. By and large, for images containing clean, well-formed closed paths, performance can be maximized by setting T_{done} lower, while highly connected images containing many spurious paths require more exhaustive search controlled by raising this value. Similarly, search depth limit and maximum north or south TreeNode list size are application dependent. For the first of these parameters, our implementation uses the value 20 chain steps which permits detection of closed paths in significantly cluttered and broken curvilinear data. We limit TreeNode list size to 20 TreeNodes. This is mainly to limit the time searching pathological configurations as shown in Fig. 9a. Step A.2.2 is also needed only for pathological configurations, as shown in Fig. 9b. While these situations can occur occasionally as deliberate drawing content, more often the pathological cases that catch the algorithm without these safeguards arise in certain kinds of of image noise such as Fig. 9c.

6.4 Seed Selection and Elimination

The most thorough algorithm for finding high global goodness-measure closed paths would perform the foregoing search algorithm repeatedly using every contour fragment as a seed, for all four search conditions (maximally-turning/smooth-continuation, west closures and east closures). This would result in repeated detection of the same contour, seeded at every curve fragment along its length. In practice this is very inefficient and not necessary.

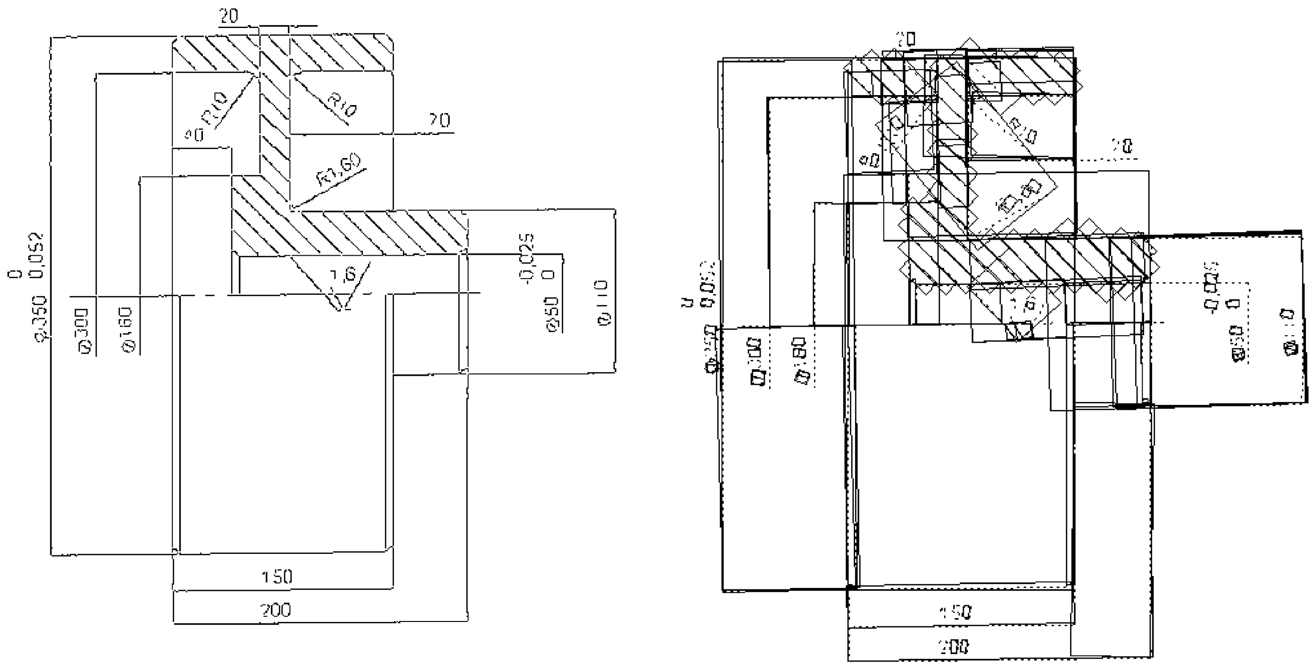
To avoid this extra work, we mark each contour fragment every time it is found to participate in an above threshold closed path (T_{accept}). The mark applies to only one of the four search conditions, depending on the path tracing direction.

To amplify the benefit of this strategy for eliminating likely redundant seeds, it pays to be judicious in choosing seeds early that will be likely to yield closed paths and therefore eliminate many other curve fragments from further consideration. To this end, we separate seeds forming an isolated corner with another curve fragment from those in the interior of X or T-junctions. Search for smooth continuation paths is run on the corner-forming seeds first. Search for both smooth continuation and maximally turning paths is run next on interior seeds.

6.5 Candidate Consolidation

The search procedure in general, returns a set of candidate closed paths that may contain multiple representations of the same or nearly the same path. These need to be consolidated down to a nonredundant set of high-goodness closed figures.

This is done by clustering closed path candidates according to similarity of pose. Pose is the five-parameter vector specifying the oriented bounding box enclosing the path.



(a) (b)

Fig. 16. (a) Input data: 409 curve fragments (scanned from [1]). (b) Results: 126 closed paths. Processing time: Seven seconds.

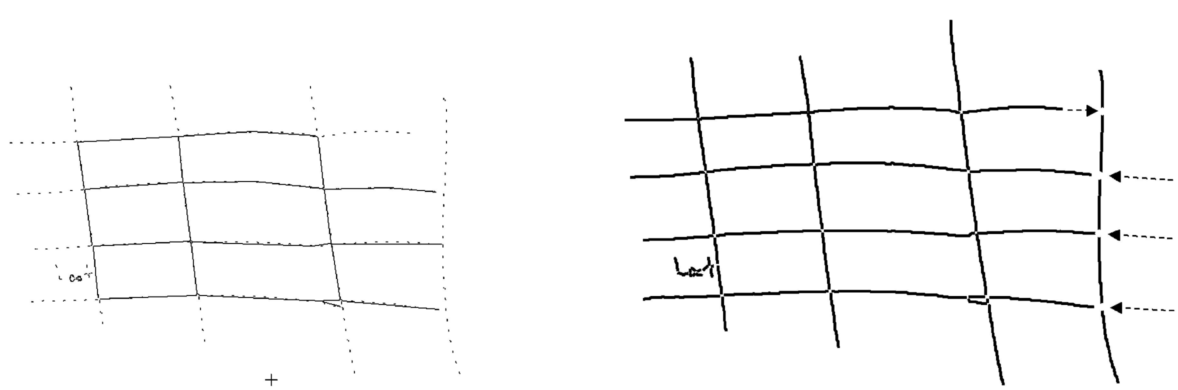
See Fig. 10. Within each cluster, consolidation proceeds in two steps. First, any path is eliminated that is subsumed by any other path. Path A is subsumed by Path B if Path A's global goodness is no greater than Path B's and its support is a subset of Path B's. Second, any path is eliminated whose pose is sufficiently similar to another path whose goodness score is greater. Search for candidate paths of similar pose is very fast, mediated by a spatially- and scale-indexed data structure [14]. See [18] for a suitable pose similarity measure.

7 Performance

Figs. 11, 12, 13, 14, 15, and 16 present results of the closed path finding algorithm on various types of data. Runtimes are stated for the closed path finding algorithm itself, and do not include image preprocessing and data preparation stages. Timing is for a Java implementation running on a 700 MHz Pentium. Statistics for Fig. 12 are typical for a complex image:

input curve fragments	734
time	8 seconds
smooth continuation paths found initially	293
maximally turning paths found initially	81
paths after consolidation	139
TreeNodes created	11,592
paths potentially tested	10,286
times search algorithm called	1,032

These figures exercise a number of the objectives discussed in Section 3. The algorithm detects partially



(a) (b)

Fig. 17. (a) Closed paths found for grid region of Fig. 11a, depicted as approximations to the curve fragments themselves instead of oriented bounding boxes. (b) A data preparation step to find virtual T-junctions would break curve fragments into pieces permitting the closed paths representing the right-side table cells to be closed on four sides.

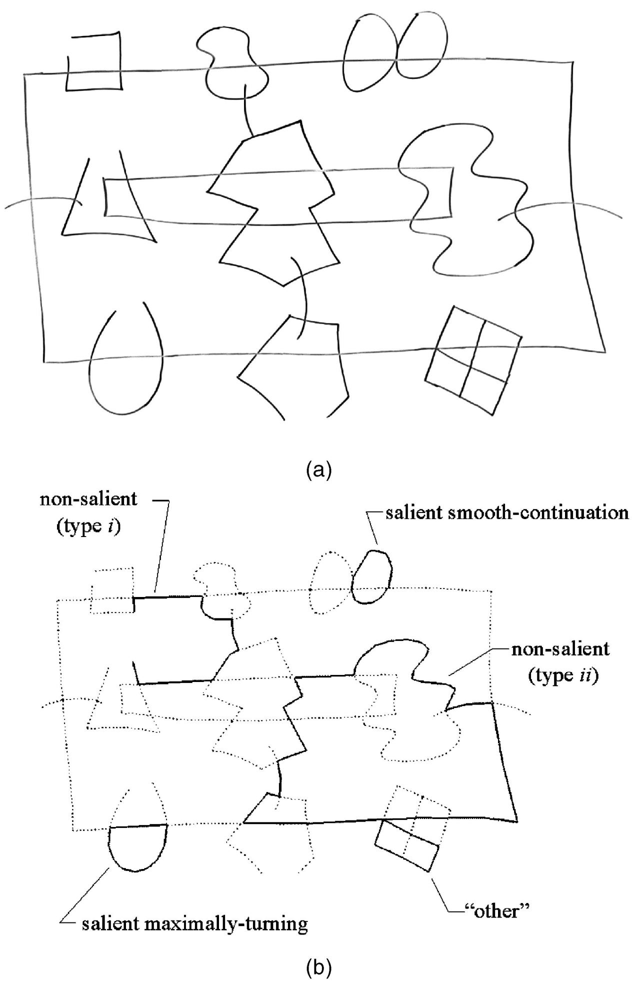


Fig. 18. (a) Test figure. (b) Examples of types of closed paths that can be traced through this figure.

closed contours containing gaps and/or a single large opening. It finds partially concave, as well as strictly convex paths. It identifies multiple closed paths that share contours in common. It tolerates misleading local evidence about path continuation. It delivers the perceptually salient closed paths occurring in a grid among the presence of many irrelevant paths satisfying the goodness measure. Fig. 13 offers a detail from Fig. 12, showing how smooth-continuation paths are distinguished from maximally-turning paths.

Fig. 11 exhibits a subtle effect which may be considered a system failure. Fig. 17a presents a closeup view of the paths found in the vicinity of the grid. Here, closed paths are rendered not by their oriented bounding boxes, but by approximations to the curve fragments that comprise them. Note that the two rightmost paths do not include curve fragments corresponding to the right side of their rectangular shapes. This is because the rightmost vertical stroke in the grid is broken into only two curve fragments, none of which correspond to suitable pieces to complete these cells. A more elaborate data preparation stage which contains mechanisms to introduce curve breaks at “virtual” T-junctions, as suggested in Fig. 17b, would alleviate this issue.

A second failure mode (not shown) occurs with wiggly or zig-zag contours forming closed regions. Such contours are normally broken into large numbers of small fragments joined end-to-end. When these joins are interpreted as corner junctions, a closed path search would be required to make a large number of turns opposite the preferred direction for a given clockwise/counterclockwise turn direction. These are penalized according to values in the junction preference table and the search can be extinguished before a good closed path is found. In our view, wiggly paths fall under the same category as “sketchy” curves (Fig. 3f) and should be dealt with by multiscale contour texture processes turning these into single contour “chunks.”

8 Discussion

The closed paths delivered by this algorithm are a subset of the plethora of available paths that simply satisfy the global goodness measure. By design, the operation of search itself seeks the more salient paths first according to local junction preferences, and these normally lead to termination of search before spurious paths are entertained.

The effects of varying T_{cumatt}, as well as the significance of the smooth-continuation and maximally-turning constraints, are shown in Fig. 18 and Table 3. This test figure contains ten nominal contour figures (two oval figures are tangent) intersected by two rectangles, yielding nominally twelve salient closed paths satisfying a smooth-continuation constraint. Additionally, the figure contains approximately 25 compact maximally-turning paths. The present algorithm finds these closed paths quickly, as shown in the first column of the table. In addition to the salient contours, the algorithm returns two “nonsalient” contours. These are paths that satisfy the global figural goodness criteria, but are not perceptually salient because, as shown in Fig. 18b, they are incomplete or enclose “ground” (nonsalient type *i*) or because they take inconsistent paths through junctions (nonsalient type *ii*). Both nonsalient contours returned by the algorithm on Fig. 18a are of type *i*. “Other” closed paths are classified as those that may be neither strictly smooth-continuation nor maximally-turning, yet are perceptually accessible, such as adjacent pairs of squares in the grid figure element. By these criteria, approximately 75 percent of the closed paths found in each of Figs. 14, 15, and 16 would be considered salient, 25 percent nonsalient.

Column 2 shows that the algorithmic distinction between smooth-continuation and maximally-turning contours is indeed critical, both in search performance and in results. Here, we tested a simplified version of the algorithm in which all junction preference scores were set uniformly to to 1. The effects of this are to remove systematic selection of child nodes to expand during search and to eliminate pruning of unpromising nodes in the tree based on path characteristics. Also, it is not meaningful in this situation to do sophisticated seed selection as described in Section 6.4. Not only is this

TABLE 3
Performance and Results of the Present Algorithm and Variations (See Text)

	default algorithm	impoverished algorithm (see text)	$T_{cumatt,\ S.C.} = .3$ $T_{cumatt,\ M.T.} = .45$	$T_{cumatt,\ S.C.} = .8$ $T_{cumatt,\ M.T.} = .95$
seconds	2	25	15	1
# TreeNodes created	1570	14979	7884	1217
paths potentially tested	1245	22737	20978	496
# times search called	165	127	165	165
salient max-turning paths returned	24	22	25	24
salient smooth-continuation paths returned	12	8	12	11
non-salient paths returned	2	44	57	2
"other" paths returned	6	1	6	6

impoverished algorithm much slower, but it returns many nonsalient paths (many nonsensical, of type *ii*) while it fails to find several significant smooth-continuation paths.

Columns 3 and 4 show the effects of varying the thresholds on cumulative junction preference score, T_{cumatt} (Section 6.3). These parameter govern how many "wrong turns" a path may take. Reducing T_{cumatt} results in discovery of more paths, while increasing it reduces the number of paths delivered that violate local junction preferences for smooth-continuation or maximal-turning. When T_{cumatt} is cut by half (column 3), the algorithm performs significantly more search. It finds all of the target smooth-continuation and maximally-turning paths, while it also returns a large number of nonsalient paths. These nonsalient paths are of type *i* however, and, as such, they tend to make sense in terms of consistency of tracing through junctions. When T_{cumatt} is increased by half toward its maximum value (column 4), search is terminated more quickly, but at the cost of failing to discover all of the salient closed figures (in this case it fails to find the jagged object in the center of the figure).

In the context of our larger research program, to bring the philosophy and techniques of Perceptual Organization to bear on applications surrounding informal documents [20], the performance of this algorithm is quite satisfactory. For example, closed paths represent a very convenient intermediate stage for shape classification leading to beautification of sketches in terms of rectangles, triangles, ellipses, and other formal graphic objects.

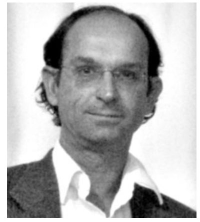
To the extent that edge detection or other preprocessing steps deliver significant intensity, color, or texture boundaries in photographic scenes in terms of extended curvilinear boundary fragments, the present approach can prove effective beyond the domain of line art as suggested by Figs. 14 and 15.

However, the detection of closed regions based on curve tracing suffers an inherent drawback. In order to be discovered by search, closed paths must be represented in the junction graph by links between contour ends. These may not be easy to construct, especially when perceptually compact regions arise from blurry, broken up, or weakly defined boundaries. A complementary approach might detect locally concave regions, then group pairs or larger sets of these whose concavities all face one another or share space in common. A role for automatic tracing processes has been proposed for human intermediate level vision [21]. It may be entirely appropriate for Perceptual Organization stages of computer vision systems to employ multiple strategies for detecting perceptual closure, including the one presented here.

REFERENCES

[1] S. Ablamayko, A. Gorelik, and S. Medvedev, "From Recognized Engineering Drawings to 3D Object Reconstruction," *Proc. Third IAPR Int'l Workshop Graphics Recognition,* pp. 313-319, 1999.
[2] R.O. Canham, S.L. Smith, and A.M. Tyrrell, "Recognition and Grading of Severely Distorted Geometric Shapes from within a Complex Figure," *Pattern Analysis and Applications,* vol. 3, no. 4, pp. 335-347, 2000.
[3] S. Casadei and S. Mitter, "Beyond the Uniqueness Assumption: Ambiguity Representation and Redundancy Elimination in the Computation of a Covering Sample of Salient Contour Cycles," *Computer Vision and Image Understanding,* vol. 76, no. 1, pp. 19-35, 1999.
[4] J. Dolan and E. Riseman, "Computing Curvilinear Structure by Token-Based Grouping," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 264-270, 1992.
[5] J. Elder and S. Zucker, "Computing Contour Closure," *Proc. European Conf. Computer Vision,* pp. 399-412, 1996.
[6] M. Fleck, "Local Rotational Symmetries," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 332-337, 1986.
[7] D. Huttenlocher and P. Wayner, "Finding Convex Edge Groupings in an Image," *Int'l J. Computer Vision,* vol. 8, no. 1, pp. 7-27, 1992.
[8] D. Jacobs, "Robust and Efficient Detection of Salient Convex Groups," *IEEE Trans.Pattern Analysis and Machine Intelligence,* vol. 18, no. 1, pp. 23-37, Jan. 1996.
[9] J. Jorge and M. Fonseca, "A Simple Approach to Recognize Geometric Shapes Interactively," *Proc. Third IAPR Int'l Workshop Graphics Recognition,* pp. 251-258, 1999.
[10] S. Mahamud, K. Thornber, and L. Williams, "Segmentation of Salient Closed Contours From Real Images," *Proc. Int'l Conf. Computer Vision,* 1999.
[11] R. Mohan and R. Nevaita, "Perceptual Organization for Scene Segmentation and Description," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 11, pp. 1121-1139, Nov. 1992.
[12] T. Pavlidis, "An Automatic Beautifier for Drawings and Illustrations," *Proc. ACM SIGGRAPH '85,* vol. 19, no. 3, pp. 225-234, 1985.
[13] S. Sarkar and K. Boyer, "Integration, Inference, and Management of Spatial Information Using Bayesian Networks: Perceptual Organization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 3, pp. 256-274, 1993.
[14] E. Saund, "Symbolic Construction of a 2D Scale-Space Image," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 8, pp. 817-830, 1990.
[15] E. Saund, "Putting Knowledge Into a Visual Shape Representation," *Artificial Intelligence,* vol. 54, pp. 71-119, 1992.
[16] E. Saund, "Labeling of Curvilinear Structure Across Scales By Token Grouping," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 257-263, 1992.
[17] E. Saund, "Bringing the Marks on a Whiteboard to Electronic Life," *Proc. Cooperative Buildings—Integrating Information, Organizations, and Architecture: Second Int'l Workshop, CoBuild '99 (Lecture Notes in Computer Science 1670),* N. Streitz, J. Siegel, V. Hartkopf, and S. Konimi, eds., Springer, 1999.

[18] E. Saund and T. Moran, "A Perceptually-Supported Sketch Editor," *Proc. ACM Symp. User Interface Software and Technology,* pp. 175-184, 1994.
[19] E. Saund and T. Moran, "Perceptual Organization in an Interactive Sketch Editing Application," *Proc. Int'l Conf. Computer Vision,* pp. 597-604, 1995.
[20] E. Saund, J. Mahoney, D. Fleet, and D. Larner, "Perceptual Organization as a Foundation for Graphics Recognition," *Proc. Fourth IAPR Int'l Workshop Graphics Recognition,* Sept. 2001.
[21] S. Ullman, "Visual Routines," *Cognition,* vol. 18, pp. 97-159, 1984.



Eric Saund received the BS degree in engineering and applied science from the California Institute of Technology and a PhD degree in cognitive science from the Massachusetts Institute of Technology. He is a senior member of the research staff at the Palo Alto Research Center. His research in computational vision specializes in perceptual organization in the domain of document images, with applications in ubiquitous document imaging, diagrammatic user interfaces, and perceptually-supported image editing. He is a member of the IEEE.