The Use of Neighbourhood Matching in Constructing Hidden Object Topology

Peter Varley

***Abstract*—My interest is automatic interpretation of natural line drawings of engineering objects. Such drawings only show that part of an object visible from a particular viewpoint, and the major challenge in interpreting them is deducing the remaining, unseen part of the topology. Since engineers routinely use sketched drawings as a means of communicating structure, using well-established (if unstated) conventions to deduce the invisible topology, it should be possible to identify these conventions and use them to interpret such drawings automatically.**

One practical approach to constructing the hidden topology is to treat the construction process as a search through the space of possible complete topologies. When this approach is used, the most frequent problem is that of mismatched vertex neighbourhoods: the side of an edge which is convex at one end is concave at the other.

This paper describes a simple method for overcoming the problem of mismatched neighbourhoods, and analyses the results which are obtained when this method is incorporated in the construction process.

Insights gained from this work may also be applicable to other fields, most obviously perception psychology and the "healing" of CAD models.

***Keywords*—Hidden Topology, Neighbourhood Matching, Boundary Representation Models**

I. Introduction

A. Objective

My overall goal is to create a 3D solid model automatically from a single 2D drawing. A tool which could quickly interpret line drawings of engineering objects as boundary representation CAD models would be of significant benefit in the process of engineering design. It would enable designers to spend more time on the creative aspects of their job and less on the routine aspects, it would reduce time spent correcting mistakes by allowing instant visualisation, and the simpler "what you draw is what you imagine" interface will be less distracting than an array of menus and icons.

B. Terminology

A *drawing* depicts an *object*. The *junctions*, *lines* and *regions* of the drawing often, but not always, correspond to the *vertices*, *edges* and *faces* of the object. A drawing is a *natural line drawing* if it depicts only those parts of the object visible from some chosen viewpoint.

A vertex is *trihedral* if exactly three edges meet at it. An object is *trihedral* if all of its vertices are trihedral.

My main interest is in *solid objects*, the faces of which bound a single continuous finite volume. A solid object is a *polyhedron* if all of its faces are planar. A polyhedron is a *normalon* if all of its edges and face normals are aligned with one of three mutually orthogonal axes, or a *quasi-normalon* if all of its vertices terminate at least one edge aligned with one of the three mutually orthogonal axes.

An object or drawing is described by its *topology* (discrete data such as vertex/edge connectivity) and *geometry* (continuous data such as vertex coordinates and edge lengths).

A drawing is from a *general viewpoint* if no small change in the viewpoint changes the topology of the drawing. I assume that all drawings are from general viewpoints.

C. Problem Statement and Applicability

The problem of creating hidden topology is this: given the frontal geometry of an object (sometimes called a 2½D model), determine the topology of the complete object, including those parts not visible to the viewer.

This paper does not discuss how frontal geometry is produced from natural line drawings. Instead, it takes a correct frontal geometry (or nearly correct, allowing for the possibility of drawing inaccuracies) and creates a complete object from it.

The techniques described in this paper are intended for use as part of a system which reconstructs boundary representation CAD models from 2D natural line drawings.

The ideas underlying these techniques may also be of use in other areas. Since our aim is to emulate the human ability to interpret natural line drawings, perception psychology is one obvious area in which they should be considered. Since our methods apply to incomplete wireframe models, they may be relevant to other applications which also deal with incomplete models, such as reverse engineering and the "healing" of broken models.

Peter Varley is with the Department of Mechanical Engineering and Construction, Universitat Jaume I, Castellón de la Plana, Spain. E-mail: varley@emc.uji.es
This work was funded by financial support from the Ramon y Cajal Scholarship Programme.

D. Structure of Paper

Section 2 describes existing approaches to constructing hidden topology. Section 3 describes the original contribution of this paper: a method for detecting mismatched neighbourhoods, a frequent cause of failure. Section 4 summarises my conclusions.

II. Constructing Hidden Topology

This section describes important contributions to the problem of constructing hidden topology and places the current paper's contribution in context.

Currently, the most successful methods are CSG-based methods. Two possibilities have been suggested.

The first, suggested by Grimstead [3], is to treat planes as half-space separators, faces as patches of planes, and edges as half-space operators (convex edges are "intersection" and concave edges are "union"). My preliminary investigations found this idea to be hard to implement even for normalons, and the results were unimpressive.

The second, suggested as long ago as Roberts [6] and shown by Suh [7] to be a practical approach to constructing hidden topology, is to use simple polyhedra as half-spaces (Roberts suggested using cuboids; Suh's implementation uses extrusions of polygonal end-caps).

Suh's implementation assumes an accurate drawing. It remains to be seen whether or not it can be extended successfully to freehand drawings, which will inevitably contain drawing errors.

Also, Suh's approach is inevitably suboptimal as a design tool. On the one hand, it is not an ideal method of interpreting line drawings, since it is inherently restricted to drawings of objects which can be decomposed into extrusions of polygons. On the other hand, it is not an ideal tool for design engineers, since if we are to model objects as unions and intersections of extrusions we should allow the design engineer to enter the extrusions in 3D the first place rather than insisting that he draw a 2D natural line drawing which the program interprets as extrusions.

The major alternative to CSG-based methods aims at constructing boundary representation topology in two stages: firstly, creating a complete wireframe, and secondly, filling in the remaining faces. The second stage is, in practice, much simpler than the related problem of finding all of the faces in a sketched 2D wireframe, since, given a 2D natural line drawing, we can deduce some of the faces directly from the drawing [8]. The unresolved problem is constructing a suitable wireframe.

The first attempt at creating a complete wireframe was that of Grimstead [3], whose fundamental assumption was that the objective was to find the simplest possible reconstruction. Kyratzi and Sapidis [4] have recently contributed a similar idea which makes extensive use of graph-theoretical ideas. However, the assumption can be challenged: in many cases, the simplest possible reconstruction does not lead to the object which a human would perceive the drawing as representing. The dodecahedron (Fig 1) is an obvious example of this: the simplest possible reconstruction would add three, not five, hidden vertices.

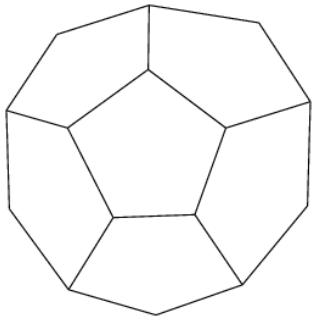

Figure 1: *Dodecahedron*

More usefully, Cao et al [1] propose a method based on human perception principles. Essentially, their idea is to create all reasonable complete wireframes, and then assess the results using heuristics derived from human perception principles, choosing the best. It can be argued that Cao et al's choice of heuristics is limited and should be wider, but the most serious objection to their idea is that the resulting algorithm appears to be of factorial order (the number of possible topological completions of a 2½D wireframe is believed to be factorial in the number of incomplete vertices [8]), so although successful for "toy" problems, it would be unacceptably slow for anything more complicated.

Since exhaustive evaluation is impractical, it is necessary to search through the space of possible completions selectively. The idea which seems to have most potential is the tree-search approach of [8]. The process of seeking the most plausible topology is implemented as a search through a *directed acyclic graph* (or *tree*) of possible topologies.

Each branch of the tree represents the addition of new topology to the wireframe: either addition of a single vertex, linked by at least two edges to existing vertices, or addition of a single edge joining two existing vertices. Thus, each node of the tree includes more topology (either a new edge, or a new vertex and two or three new edges) than its parent node.

Identical nodes reached by different routes are not merged.

Branches are generated by hypotheses about the incomplete wireframe. For example, where an existing vertex is met by only two edges, one must hypothesise that a third edge exists, and where such hypothesised edges meet, it is reasonable to hypothesise the existence of a new vertex.

[8] also discussed the utility of "macro branches" which reconstruct in a single branch the entire topology required to satisfy a hypothesis (for example, that of bilateral symmetry); these "macro branches" are not considered here.

In considering search approaches, [8] found no improvements on a standard *greedy* approach. Adding backtracking is, in general, not helpful: the result of a mistaken branch is, more often than not, a topologically valid but unexpected solid, not a topologically invalid solid, and in the absence of a detectable error condition there is nothing which triggers a backtrack.

Branches are compared on the basis of merit figures, derived from: (a) the merit of the hypothesis which generated the branch (for example, the hypothesis that the third edge of an incomplete vertex of a normalon is aligned with the unused major axis is very strong); (b) the number of competing hypotheses (for example, a vertex which is the only possible termination of a hypothesised edge is a stronger branch than a vertex which is one of several possible terminations of a hypothesised edge); (c) the number of reinforcing hypotheses (when several hypotheses suggest that the same new vertex should be added, the branch is very strong); and (d) the plausibility of the topology which would result from taking this branch (for example, if a hypothesised new vertex would be in a location which would make it visible in the original drawing, the branch is weak).

Although [8] considers the creation and assessment of hypothesis in detail, it is less thorough when considering how the tree of hypotheses may be pruned. Some hypotheses produce additional topology which cannot be reconciled with the existing topology, and such hypotheses should be rejected immediately, not fed into the assessment process.

For example, given knowledge of whether the existing and proposed edges at a vertex are convex or concave, it is usually possible to tell whether the proposed edge lies above or below the face containing the existing edges. Table I, from [8], is an exhaustive list of the possibilities at a trihedral vertex.

By way of illustration, it can be seen (in row two of the table) that at any vertex met by two convex edges and one concave edge, the turn at the corner of the face where the two convex edges meet is concave (a left-hand turn, assuming a clockwise circuit), the edge leaving the face is concave, and it is below the plane of the face (where "inside the object" is below and "outside the object" is above).

Table I: *Possibilities at a Trihedral Vertex*

Incoming edge	Outgoing edge	Turn	Leaving edge	Direction
convex	convex	right	convex	below
convex	convex	left	concave	below
convex	concave	right	convex	above
concave	convex	right	convex	above
concave	concave	left	convex	above
concave	convex	right	concave	below
convex	concave	right	concave	below
concave	concave	right	concave	above

Ensuring that these conditions are met for all proposed additional edges allows us to reject some hypotheses which are clearly erroneous. However, the idea is limited in application. It is appropriate for use for all trihedral objects, not only normalons and quasi-normalons, but there are several problems which make it difficult to extend beyond the domain of trihedral objects. It relies on line labelling, which is only reliable for trihedral and extended trihedral objects. Even given a reliable line-labelled drawing, the above/below implications of 4-hedral vertices are tedious to enumerate, and the above/below implications of higher-order vertices unlikely ever to be enumerated.

Section 3 proposes a new method for pruning out erroneous branches, based solely on geometric reasoning.

III. Mismatched Neighbourhoods

Consider, as an example, the erroneous edge introduced into the lower drawing in Fig 2. A cross-section at the left-hand end would show a quarter empty and three-quarters solid. A cross-section at the right-hand end would show a quarter solid and three-quarters empty. The part of the cross-section which is solid at one end is empty at the other, and vice versa. The **local neighbourhoods** at the two ends do not match.

Such mismatched neighbourhood problems are comparatively common: [8] found that they were the largest single category of error made by the tree-search approach.

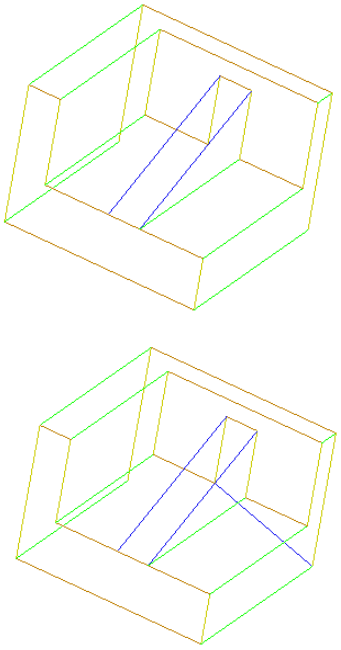

Figure 2: *Mismatched Neighbourhoods*

This section introduces an idea for avoiding the problem of mismatched neighbourhoods. What is desired is a method by which such branches, thus ruling them out of consideration entirely, not heuristics which make them less plausible, in order that some other branch of the search tree will be preferred.

It is assumed that, by analysing the drawing, the process of determining the frontal geometry can identify groups of 2D lines which correspond to edges aligned with the three major 3D object axes, as shown in Fig 3. In reality, this process is not entirely reliable: it is trivial for normalons, but there are some quasi-normalon drawings which still present problems.

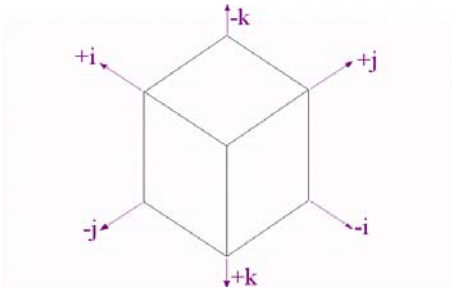

Figure 3: *Three Major Object Axes*

Secondly, it is assumed that, by analysing the drawing, the process of determining frontal geometry can determine which lines are convex, which are concave, and which are occluding (the *line-labelling* problem [2, 5]). Although, as with approaches to determining the major object axes, there is no entirely satisfactory general-case line-labelling algorithm [9], we are nearer to solving the line-labelling problem than we are to solving the problem of constructing hidden topology. For example, if the inflation process produces correct 2½D geometry, it is possible to determine geometrically which edges are convex and which concave, bypassing traditional line-labelling algorithms. For the purposes of this investigation, the remaining problems posed by line-labelling are ignored.

A. Zones

The idea described here subdivides the neighbourhood surrounding each vertex into eight *zones*, each of which is an infinitesimal axially-aligned cube. In a normalon, each of these zones must be *full* (entirely within the object) or *empty* (entirely outside the object). While parts of the object structure remain unknown, zones may also be *unknown*. In non-normalons and non-trihedral solids, zones may be *mixed* (a face of the object intersects the zone, which is neither fully inside nor fully outside the object).

For notational convenience, I label the eight zones *efg*, *efk*, *ejg*, *ejk*, *ifg*, *ifk*, *ijg* and *ijk*. Zone *efg* is the zone nearest the viewer; zone *ijk* is that furthest from the viewer.

Assume initially that we are dealing with a normalon.

At every visible vertex, the zone *efg* must necessarily be *empty* as this is the zone which includes the line-of-sight.

Each edge must be aligned with one of the three *ijk* axes. Axis-alignment of hypothesised edges is part of the existing scheme [8] (it is one of the criteria used to create hypotheses in the first place), so this information is already available. Each line must also be convex, concave, occluding with the solid face on the left, or occluding with the solid face on the right; this information comes from line-labelling. This leads to a total of twelve possible cases, as listed in Table II.

In columns 7 to 12, the position of the occluding solid (*left* or *right*) in the final six columns is as viewed in 2D. For *i*-aligned and *j*-aligned edges, this is the same as the position of the occluding solid as viewed along the edge, but this is not the case for *k*-aligned edges, since *k*-aligned edges run from top/near to bottom/far. In column 9, the occluding solid is on the *left* as viewed in 2D but on the *right* relative to the edge direction, and vice versa for column 12.

Table II: *Local Neighbourhoods for Normalons, by Edge Type and Alignment*

Edge Type	Con-vex	Con-vex	Con-vex	Con-cave	Con-cave	Con-cave	Oc-clud-ing Solid on Left	Oc-clud-ing Solid on Left	Oc-clud-ing Solid on Left	Oc-clud-ing Solid on Right	Oc-clud-ing Solid on Right	Oc-clud-ing Solid on Right
Align	*i*	*j*	*k*	*i*	*J*	*k*	*i*	*j*	*k*	*i*	*j*	*k*
Near *efg*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*
Near *efk*			*Empty*			*Empty*			*Empty*			*Empty*
Near *ejg*		*Empty*			*Empty*			*Empty*			*Empty*	
Near *ejk*		*Empty*	*Empty*		*Full*	*Full*		*Empty*	*Empty*		*Full*	*Full*
Near *ifg*	*Empty*			*Empty*			*Empty*			*Empty*		
Near *ifk*	*Empty*		*Empty*	*Full*		*Full*	*Full*		*Full*	*Empty*		*Empty*
Near *ijg*	*Empty*	*Empty*		*Full*	*Full*		*Empty*	*Full*		*Full*	*Empty*	
Near *ijk*	*Full*	*Full*	*Full*	*Full*	*Full*	*Full*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*
Far *efg*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*	*Empty*
Far *efk*	*Empty*	*Empty*		*Full*	*Full*		*Full*	*Empty*		*Empty*	*Full*	
Far *ejg*	*Empty*		*Empty*	*Full*		*Full*	*Empty*		*Empty*	*Full*		*Full*
Far *ejk*	*Full*			*Full*			*Empty*			*Empty*		
Far *ifg*		*Empty*	*Empty*		*Full*	*Full*		*Full*	*Full*		*Empty*	*Empty*
Far *ifk*		*Full*			*Full*			*Empty*			*Empty*	
Far *ijg*			*Full*			*Full*			*Empty*			*Empty*
Far *ijk*												

Ignoring T-junctions, there are eleven possible ways that a trihedral vertex can be labelled. Table III lists some additional deductions which can be made from these labels.

We can use the rules in Tables II and III to assign initial values to the eight zones surrounding each vertex. What do these rules tell us about non-normalons?

Clearly, they also apply to any *cubic corner* (a trihedral vertex where the three edges are aligned with the major object axes), regardless of what the rest of the object may contain.

Similarly, with one modification, they can be applied to any vertex which terminates an axis-aligned edge. The modification is that, where the previous section identified a zone as *empty*, all we can say about the corresponding zone of a general (non-cubic-corner) vertex is that it is not *full*, and where the previous section identified a zone as *full*, all that we can say is that it is not *empty*.

B. Hypotheses

This section describes how neighbourhood matching can be used in assessing hypotheses. As previously noted, I only consider here hypotheses which create topology incrementally (new edge, or new vertex plus two or three edges), and not macro-hypotheses (e.g. complete as much topology as possible from a hypothesised mirror plane).

If any new edge is i-aligned, the zones *efg*, *efk*, *ejg* and *ejk* of the "far" vertex must be the consistent with the zones *ifg*, *ifk*, *ijg* and *ijk* of the "near" vertex. If any of this creates a conflict, the hypothesis should be rejected.

Similarly, for j-aligned edges, the zones *efg*, *efk*, *ifg* and *ifk* of the "far" vertex must be consistent with zones *ejg*, *ejk*, *ijg* and *ijk* of the "near" vertex, and for k-aligned edges, the zones *efg*, *ejg*, *ifg* and *ijg* of the "near" vertex must be consistent with zones *efk*, *ejk*, *ifk* and *ijk* of the "far" vertex.

In the case of new-vertex hypotheses, if zone *efg* of the new vertex is empty, the merit of the hypothesis should be reduced as this implies that the vertex would be visible since the line-of-sight to the vertex passes through this zone.

The neighbourhoods of *T*-junctions are ignored; it is those of the true vertices which are important.

IV. Conclusions

In this paper, I present a reasoning process which can be used to identify the majority of erroneous hypotheses which are generated when attempting to complete partially-complete wireframe.

Table III: *Local Neighbourhood Reasoning for Normalons, by Junction Shape and Label*

Vertex Shape	+	-	Consequences
Y	3	0	zone *ijk* is full and all of the other seven zones are empty
Y	2	1	Three of the zones are full; the other five are empty
Y	0	3	Zone efg is empty and all of the other seven zones are full
L, W	3	0	One of the zones is full; the other seven are empty
L, W	2	1	Three of the zones are full; the other five are empty
L, W	1	2	Five of the zones are full; the other three are empty
6-way	3	3	Zones *ijk*, *ijg*, *ifk* and *ejk* are full; zones *efg*, *efk*, *ejg* and *ifg* are empty

References

[1] L. Cao, J. Liu and X. Tang, *What the Back of the Object Looks Like: 3D Reconstruction from Line Drawings without Hidden Lines*, IEEE Transactions on Pattern Analysis and Machine Intelligence 30(3), March 2008, 507–517.

[2] M. B. Clowes, On Seeing Things, Artificial Intelligence 2, 79-116, 1970.

[3] I. J. Grimstead, *Interactive Sketch Input of Boundary Representation Solid Models*, PhD Thesis, Cardiff University, 1997

[4] S. Kyratzi and N. Sapidis, *Extracting a polyhedron from a single-view sketch: Topological construction of a wireframe sketch with minimal hidden elements*, to appear in Computers & Graphics.

[5] D. A. Huffman, *Impossible Objects as Nonsense Sentences*, Machine Intelligence **6**, 295-323, New York American Elsevier, 1971.

[6] L. G. Roberts, *Machine Perception of Three-Dimensional Solids*. PhD Thesis, MIT, 1963.

[7] Y.S. Suh, *Reconstructing 3D Feature-Based CAD Models By Recognizing Extrusions From A Single-View Drawing*, Proc. IDETC/CIE 2007.

[8] P. A. C. Varley, A*utomatic Creation of Boundary-Representation Models from Single Line Drawings*, PhD Thesis, University of Wales, 2003.

[9] P. A. C. Varley, R. R. Martin and H. Suzuki, *Frontal Geometry from Sketches of Engineering Objects: Is Line Labelling Necessary*?, Computer Aided Design **37** (12), 1285-1307, 2005.