Interpreting Line Drawings of Objects with K-Vertices

P. A. C. Varley, H. Suzuki
Department of Precision Engineering, The University of Tokyo, Tokyo, Japan
{pvarley, suzuki}@cim.pe.u-tokyo.ac.jp

R. R. Martin
School of Computer Science, Cardiff University, Cardiff, Wales, UK
ralph@cs.cf.ac.uk

Abstract

As part of the goal of automatic creation of B-rep models of engineering objects from freehand sketches, we seek to take a single line drawing (with hidden lines removed), and from it deduce an initial 3D geometric realisation of the visible part of the drawn object. Junction and line labels, and provisional depth coordinates, are key parts of this frontal geometry.

Many methods for producing frontal geometry only work correctly for drawings of trihedral objects. However, non-trihedral K-vertices commonly occur in engineering objects. We analyse the performance of a line-labelling method applied to K-vertices, and show why methods ignoring geometric considerations are inadequate.

We give a new approach which produces both junction labels and provisional depth coordinates without any prior knowledge. Our results show that even a naïve implementation outperforms previous methods.

1 Introduction

Automatic creation of B-rep models of engineering objects from freehand sketches would be of obvious benefit to designers. Conversion of freehand sketches to line drawings is relatively straightforward (e.g. see Mitani et al [10]), so here we consider line drawings.

We wish our process to be as simple as possible for a designer to use, so we use use a single drawing, rather than multiple drawings, as input. For the same reason, we choose to interpret *natural line drawings* rather than *wireframe drawings*, in which lines occluded by the material of the solid object are nevertheless shown in the drawing.

To aid later stages of this process, we wish to deduce as much as we can from the drawing before adjusting or adding to it. In particular, we seek a *frontal geometry* of the object, a 3D geometric realisation of the visible part of the object in the drawing. This depth information is used to compare various hypotheses made about the drawing by later stages of processing. Junction and line labels [1, 3] and provisional depth coordinates are two important components of this frontal geometry. The depth coordinates (and the user's x-y-coordinates) may be adjusted later to match constraints (e.g. implied by symmetry or axis-alignment); the presence of such constraints is assessed using the frontal geometry.

Clearly, it is impossible to uniquely calculate the depths of junctions in a single line drawing—projection discards information. Nevertheless, humans can interpret line drawings, and usually agree about the depth implications. Most readers will interpret drawings such as those in Figures 1–32 in the same way.

Interpretation of single natural line drawings only becomes tractable if one makes assumptions. Most important of these is the commonly-made assumption that the drawing is made from a *general position* viewpoint [3]: no small change in the viewpoint will alter the drawing topology. We also assume that the designer has a real object in mind. This avoids the question of *realisability*: whether the line drawing is realisable as a 3D object or not.

Previous work has also often made the overly-restrictive assumption that the object portrayed is trihedral, sidestepping several problems arising when creating the frontal geometry of real engineering objects. Here we examine some of these problems using drawings of objects containing K-vertices (defined in Section 4), the most common type of non-trihedral vertex in engineering objects—we estimate that about two-thirds of non-trihedral vertices are K-vertices. We conclude that the most serious problem is that, in treating line-labelling solely as a local constraint satisfaction problem, geometry is ignored.

As existing line-labelling methods fail through ignoring geometry, we propose a new approach to the line-labelling problem based initially on geometry rather than constraint

satisfaction. Our tests show that even a naïve implementation of this is 20%–30% better at labelling drawings of non-trihedral objects than previous methods.

The purpose of this paper is twofold: to demonstrate a deficiency in existing methods, and suggest an alternative. Sections 2 and 3 consider various approaches to the problems of line-labelling and production of a preliminary frontal geometry. Sections 4 and 5 explain what K-vertices are, and examine why existing methods are inappropriate for drawings containing K-vertices. Sections 6, 7 and 8 outline our new approach to line-labelling and inflation, give a practical example, and give test results. Section 9 gives our conclusions and plans for further work.

2 Line Labelling Review

When analysing a line drawing, we assume that the 2D coordinates x_j, y_j of each junction j are known (from the original drawing—they may or may not be accurate), as are the junction pairs joined by each line, and the loops of junctions and lines forming each region.

Line-labelling labels all lines in the drawing as convex, concave or occluding; it is a well-known, often used initial stage in the interpretation of such drawings. The standard Huffman [3] and Clowes [1] approach has the advantages that it requires as input no information other than the above, and that it produces as output useful information about both the frontal geometry and the topology of the hidden part of the object. The non-silhouette lines of Figure 2 have been labelled in accordance with the Clowes-Huffman convention as convex (+), concave (-) or occluding (arrow).

Many implementations of Clowes-Huffman line labelling (e.g. [5]), use *catalogue labelling*. The catalogue contains all possible junction labels, reducing line labelling to a discrete constraint satisfaction problem: all junctions must satisfy a 1-node constraint—the label appears in the catalogue, and all lines must satisfy a 2-node constraint—the line has the same label at both ends.

However, even for trihedral objects described by the Clowes-Huffman catalogue, line-labelling is not truly a 1-node and 2-node constraint problem. In order to label objects correctly, one must also satisfy non-local (and perhaps vaguer) geometric constraints—the object must be realisable, and psychological ones—that the object must be the one the user intended. However, in the trihedral world, the advantages of catalogue-labelling (it usually runs in $O(n)$ time [11]), mean that occasional failures are tolerated.

In the non-trihedral world, the advantages are lost but the disadvantages remain. For example, the catalogue of tetrahedral vertices for polyhedra [15] is not sparse, so catalogue-based labelling is often too slow [18] to be useful.

Use of the tetrahedral vertex catalogue also results in many more possible solutions to the constraint satisfaction problem, and choosing a good solution is difficult [18]: often labellings which are valid solutions to the constraint satisfaction problem may not be realisable geometrically.

For trihedral drawings, Sugihara [14] suggested that labelling should be used to obtain a reasonably small number of candidate interpretations, each corresponding to a legal labelling, whose geometric realisability could then be determined by subsequent stages of processing. In a previous paper [18], we explained why this idea is inappropriate for general non-trihedral drawings. In reconsidering the idea for drawings where the only permitted non-trihedral vertices are K-vertices, we note that (i) for more than half of the test drawings in this paper, the number of legal labellings is no greater than five, but (ii) Figures 18 and 29 have 1177 legal labellings each, Figures 15 and 19 1208 each, and Figure 30 has 10398. Processing so many candidate interpretations is impractical. The search and selection processes must be aided by heuristics based on psychological plausibility [18].

As well as geometric considerations which already exist in the trihedral world (e.g. if a pocket is as deep as the surrounding material, it should be a through hole, not a pocket), there are geometric considerations unique to K-vertices (see Section 4). For objects with K-vertices, reducing the labelling problem to one of simple 1-node and 2-node constraints is inappropriate.

Other approaches exist for labelling, even for trihedral objects (see [16] for a summary). However, they continue to treat the problem as one of local constraints, and are also inappropriate for labelling drawings with K-vertices.

2.1 Relaxation Methods

Previously [18], we gave a "relaxation algorithm" approach to solving the 1-node and 2-node discrete constraint satisfaction problem by probabilistic rather than deterministic means. This approach is used as a benchmark against which we test the new idea described in Section 6.

Relaxation has two advantages over e.g. Kanatani's algorithm [5]. Firstly, it is considerably faster when using the non-trihedral junction catalogue. Secondly, although less reliable overall for labelling than other methods, it *is* more reliable for drawings with single K-vertices [16]. This seems to be because relaxation methods gradually accumulate data from other parts of the drawing beyond the immediate 1- and 2-node neighbourhoods, whereas heuristic-based selection methods have no heuristics to use (one K-vertex is as good as another) and are effectively selecting at random.

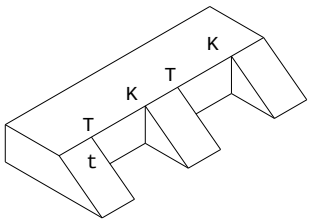

Figure 1.

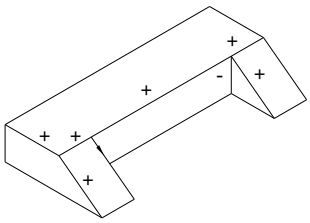

Figure 2.

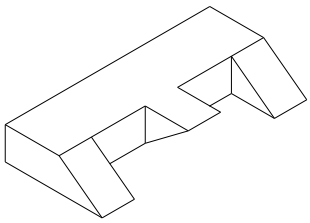

Figure 3.

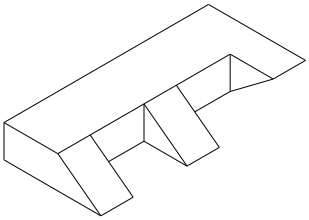

Figure 4.

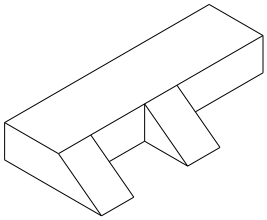

Figure 5.

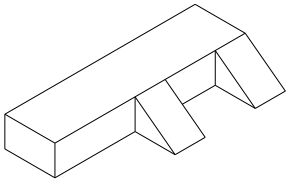

Figure 6.

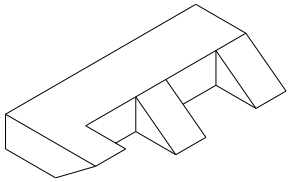

Figure 7.

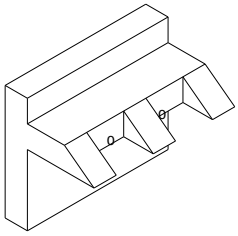

Figure 8.

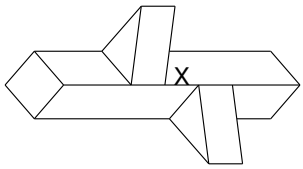

Figure 9.

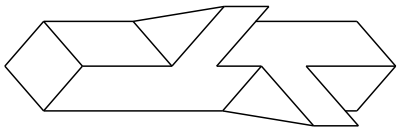

Figure 10.

3 Inflation Overview and History

Inflation is the process of adding depth to the visible part of the object in the drawing. Relevant available information is encoded using *compliance functions* to give a system of equations. The best solution to the resulting system provides as outputs the depth (z-) coordinates for each visible vertex (e.g. those marked K and T in Figure 1) and for each point at which a partially-occluded edge disappears from view (such as the one marked t in Figure 1).

We have, as a minimum available input, the same information as that for line-labelling. If inflation is not the first stage of an overall system, we may have additional information (e.g. the output of line-labelling); certain compliance functions rely on such information being available.

There are two basic approaches to inflation. We may choose to only use information which can be translated into a linear system of equations, and find the optimum solution by linear algebra. Instead, we may also include non-linear equations, and find the solution using iterative optimisation. In practice, the simpler, quicker linear method is generally used when an approximate geometry suffices at an early stage. Slower and better non-linear optimisation methods are more suitable when a fully-correct, "beautified" geometric realisation of an object is required at a later stage of processing [17], so are not considered further here.

Previously [17], and here, we use the simplest linear method, a system of equations linear in just the set of variables z_v, where z_v is the depth coordinate of vertex v. Grimstead [2] used a more complex linear approach based on equations constraining vertices to lie in faces: $P_f x_v + Q_f y_v + z_v + C_f = 0$; the output variables are P_f, Q_f and C_f for each face and z_v for each vertex. The extra complexity does not seem worthwhile.

Table 1, based on the list in Lipson and Shpitalni [7] with additions, lists various compliance functions. The table indicates for each compliance function: what input information it needs, whether or not it can be used in either linear system approach, and whether it is, by itself, inflationary (some compliance functions are not: e.g. line parallelism has the trivial solution $z_v = 0$ for all vertices). For example, "Prismatic face" [7] requires the knowledge that the drawing is of an extrusion, and comprises a number of compliance functions, some but not all of which are inherently inflationary. Although "Face planarity" [7] cannot be translated directly into an equation linear in z-coordinates, it can

Name	Ref.	Requires	Linear (PQzC)	Linear (Z)	Inflate
face planarity	[7]	Label	Y	*	N
4-vertex planarity	[17]	Label	Y	Y	N
line parallelism	[7]	Parallel	Y	Y	N
line verticality	[7]	(none)	Y	Y	N
isometry	[7]	(none)	Y	Y	N
corner orthogonality	[12]	(none)	Y	Y	Y
junction label pairs	[17]	Label	Y	Y	Y
skewed facial orthogonality	[4]	(none)	Y	N	Y
skewed facial symmetry	[4]	Symmetry	Y	N	Y
line orthogonality	[7]	(none)	N	N	Y
minimum standard deviation of angles	[9]	(none)	N	N	Y
face perpendicularity	[7]	(none)	N	N	Y
"prismatic face"	[7]	Extruded	Y	Y	*
line collinearity	[7]	(none)	Y	Y	N
planarity of skewed chains	[7]	Symmetry	N	N	Y
mirror symmetry	[19]	Symmetry	N	N	Y

Table 1. Compliance Functions for Inflation

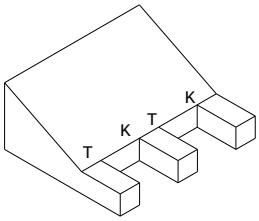

Figure 11.

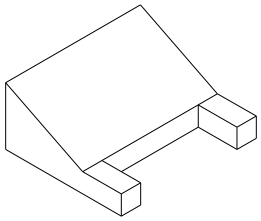

Figure 12.

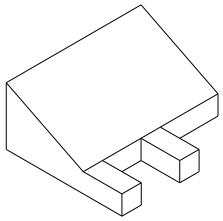

Figure 13.

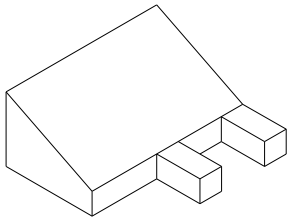

Figure 14.

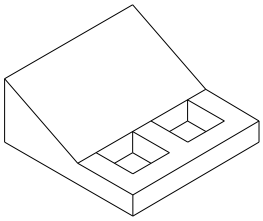

Figure 15.

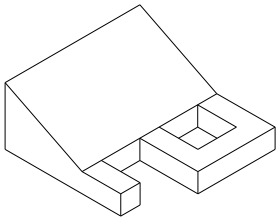

Figure 16.

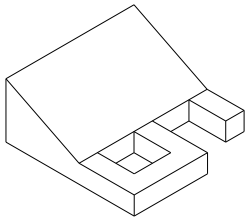

Figure 17.

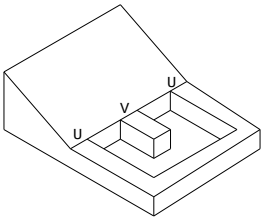

Figure 18.

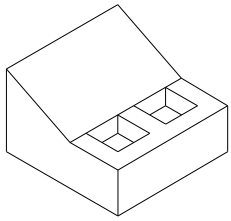

Figure 19.

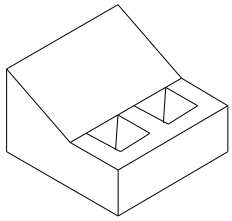

Figure 20.

be used in the simpler linear approach if expressed as several "4-Vertex planarity" compliance functions.

Deducing whether pairs of lines should be parallel in 3D, as required for the line parallelism compliance function, is only straightforward under a strong interpretation of the "general viewpoint" rule such as that used by Sugihara [14]. However, such a strong interpretation does not allow for freehand drawing errors—lines which are almost parallel in 2D need not be almost parallel in 3D—so we prefer a weaker interpretation that no small change in viewpoint results in a topological change to the line drawing. Given this, there is no entirely reliable method of determin-

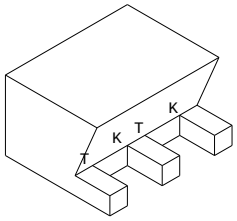

Figure 21.

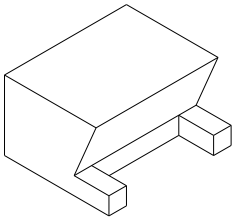

Figure 22.

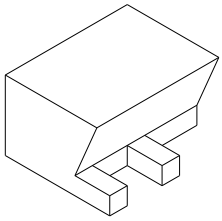

Figure 23.

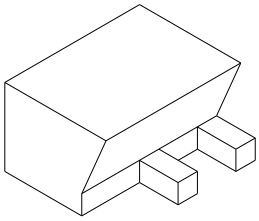

Figure 24.

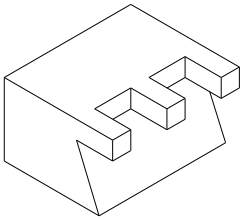

Figure 25.

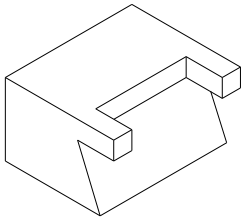

Figure 26.

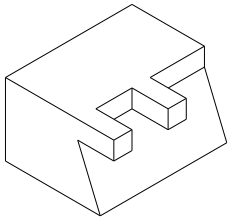

Figure 27.

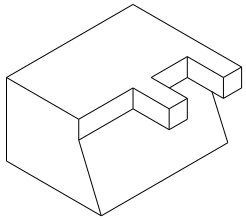

Figure 28.

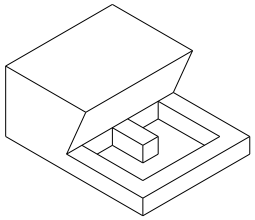

Figure 29.

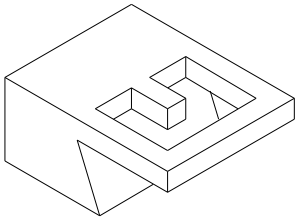

Figure 30.

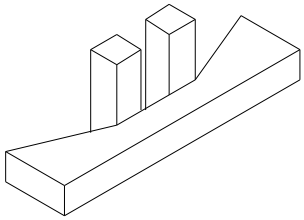

Figure 31.

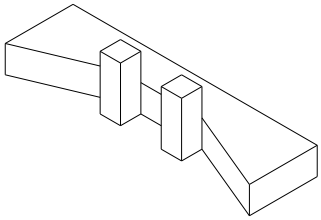

Figure 32.

ing whether the user *intended* two lines to be parallel.

Determination of whether or not a drawing represents an extrusion, or a mirror-symmetric object, as required by some compliance functions, is even less straightforward, and we do not consider such approaches here.

4 K-Vertices

If one breaks down engineering objects into a union of adjacent "building blocks", the two most common blocks are cuboids and axis-aligned wedges [13]. Although all vertices of *isolated* cuboids or wedges are trihedral, when they are combined, non-trihedral vertices can appear as a result of coinciding edges. All of the test objects shown here can be assembled from cuboids and axially-aligned wedges.

We use the terminology: a *K-vertex* is a tetrahedral vertex produced by coinciding edges of cuboids and an axially-aligned wedge; a *K-junction* is a 2D junction in which all four edges touching a K-vertex can be seen (and which resembles in shape the letter **K**). All K-vertices have 4 edges with at least 1 convex and 1 concave edge; a K-vertex with 3 convex edges is said to be *convex*, with 3 concave edges is said to be *concave*, and otherwise is said to be *mixed*.

Our test set includes ten drawings of objects featuring each of the three main types of K-junction, plus two drawings of objects including the "occluded" mixed K-junction. None of these drawings is in the set of over 500 drawings [16] used to tune the relaxation algorithm.

Figure 1 shows four convex K-vertices seen from two different viewpoints, appearing as K- or T-junctions as annotated. Figure 2 shows a similar but simpler object, and Figure 3 has two of the K-vertices on the underside of the object. Figures 4–7 are similar.

Figure 8 shows that a line which separates two regions corresponding to parallel faces *must* occlude one or the other—it cannot be convex or concave. There are two such lines in this figure, annoted O. Figures 9 and 10 illustrate another point: the central line, annotated X, corresponds to an edge with an axis of symmetry through its mid-point, so for reasons of symmetry as well as geometry that edge cannot be occluding.

Figure 11 shows a drawing in which four mixed K-vertices can be seen from two different viewpoints, appearing as K-or T-junctions. Figures 12–14 show simpler objects, and Figures 15–18 show similar objects, still with mixed K-vertices, and with one or two through holes. The

internal features of Figure 19 are the same as those of Figure 15, but the bottom three junctions have moved. Nevertheless, it is the internal features which should be labelled differently, as pockets rather than through holes, while the labels of the bottom three junctions remain the same.

While similar to Figures 15 and 19, Figure 20 should be easier to label, since it is not necessary to determine whether the internal features are holes or pockets. It also shows that so-called "drawing errors" may not necessarily result from carelessness by the user, but may be needed to avoid accidental coincidences—this figure cannot be drawn in isometric projection without breaking the general position rule.

Figure 21 shows an object with four concave K-vertices seen from two different viewpoints, appearing as K- or T-junctions as annotated. Figures 22–24 show simpler objects. Figure 25 shows another view of the object in Figure 21. Again, Figures 26–28 are similar, simpler objects.

Figures 29 and 30 show similar objects, still with concave K-vertices, which include a through hole. As another illustration of the non-local nature of the labelling problem, although it is possible to label the object in Figure 29 as having a through hole or a pocket, the labelling must be consistent: either "all hole" or "all pocket". Figure 30 is particularly tricky in that two of the regions of the drawing correspond to the same face of the object. Figure 31 shows an object in which two occluded K-vertices can be seen from different viewpoints. Figure 32 shows a similar object from another viewpoint. Note that from no viewpoint is it possible to see all four edges meeting this sort of K-vertex.

5 Labelling Discussion

5.1 Benchmark Method

In testing the ability of relaxation methods to label Figures 1–32 correctly, we have used (with minor modifications) the algorithm described in [18]. That paper used an analogy with crystallisation to describe its behaviour, and noted that the crystallisation process was often too quick. This had advantages for drawings which included one particularly unusual junction amidst many common junctions, but was detrimental on the whole.

A reviewer of [18] suggested that, to avoid this over-rapid crystallisation, when propagating vertex label probabilities to edges, the update factor should be the maximum vertex label probability contributing to a particular edge label, not the sum of all vertex label probabilities which require that edge label. This resulted in about a 10% reduction in mislabellings, but also made the results more sensitive to the initial values of vertex label probabilities. We therefore pursued this idea no further.

As an alternative for avoiding too-rapid crystallisation, we tried under-relaxation, reducing the influence of neighbours at each iteration step. Our tests suggest that under-relaxation is useful when determining junction labels from edge labels but not vice versa.

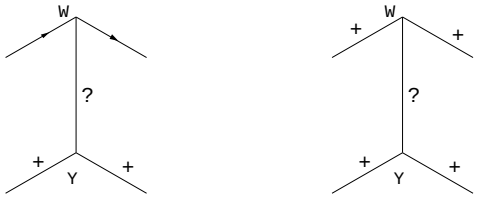

Figure 33. What Label?

A further addition was made to the algorithm in [18]. Consider the configuration of lines shown on the left of Figure 33. The vertex at W lies in the plane of both faces, as does the vertex at Y. The edge joining them must therefore also lie in the plane of both faces, and thus cannot occlude either. The line must thus be labelled convex, as a Y-shaped junction with two convex lines and one concave is not possible. Generalising, if neither of two junction labels allows the corresponding vertex to occlude a face, an edge joining them cannot occlude the face either, so the corresponding line cannot be labelled as occluding that region. This can be translated into a 2-node constraint on neighbouring junction labels.

However, incorporating this constraint poses a further problem. Consider the unknown line shown on the right of Figure 33. It cannot be convex—no possible vertex produces an all-convex W-junction on projection. Similarly, it cannot be concave—no possible vertex produces a Y-junction with two convex lines and one concave. Yet it cannot occlude either face—the vertices at either end of the edge (and hence also the edge) are clearly in the plane of both faces. In the absence of a backtracking mechanism, all that can be done is report that the given configuration of line labels is invalid, and (unhelpfully) give up.

Note that adding the under-relaxation factors described above does not remove the problem of an edge having no possible labels, but does significantly reduce the chance of the problem occurring. Although it helps, adding under-relaxation is not a full solution.

5.2 Geometric Constraints

Our drawings illustrate a number of geometric constraints which go beyond the local constraints used by existing labelling algorithms.

Firstly, in Figure 18, consider the two edges marked U where the U-shaped rim face meets the sloping face. Clearly, since these two edges are collinear, they must have the same label. Generalising, if two or more edges lie between the same two faces, (i) if such edges are collinear, the

labels must be the same, (ii) if such edges are non-collinear, the labels must be different, and at least one must be labelled as occluding. This is still a 2-node constraint (but a non-local one) between two edges, derived from information which can be obtained directly from the drawing (we may allow for user error by using a probability function to determine how likely two lines are to be collinear).

Secondly, in Figure 18, the edge marked V, where the smaller central protrusion meets the sloping face, must have the same label as the two edges previously discussed. This may seem to be a straightforward generalisation: any rule applying to a single face also applies to two coplanar faces. However, deciding whether two faces are coplanar can not be done simply using only information in the drawing—lines collinear in 3D must also be collinear in 2D, but there is no direct way of identifying in 2D which faces are coplanar in 3D. This kind of constraint on line labels must be satisfied by any correct labelling, but it cannot be imposed by any constraint satisfaction algorithm which takes only the lines and junctions of a drawing as its input.

In a similar way, extra geometric reasoning is also needed before line labelling in other cases. A line which separates two regions corresponding to parallel faces *must* occlude one or the other. Deciding that the three protrusions in Figure 11) are parallel is simple, but in cases like Figure 8, reasoning purely from the lines in the drawing that the two faces are parallel is beyond the capabilities of any known method. Their parallelism must be determined geometrically.

Consider now the central line in Figures 9 and 10. Reasoning as before shows that it cannot be occluding, but with a small change in the angles of the lines in the drawing, it *would* be reasonable to label this line as occluding. Such reasoning is also beyond the capabilities of any known method. However, even if it were geometrically plausible to label this line as occluding, to do so would be wrong. An axis of rotational symmetry passes through the centre of the line, and labelling the line as occluding would break this symmetry. Our process must be tolerant of drawing errors, and so it is surely more important here to preserve symmetry than to allow *anything* which the geometry permits. Again, detecting potential symmetries requires more input than simply the lines and junctions of the original drawing.

The latter, however, is an exceptional case. The main problem with current labelling mechanisms is that *geometric information* is not taken into account. An approach is needed which makes use of geometric information which can immediately be deduced from the line drawing.

6 Inflation and Labelling: A New Approach

In order to inflate a labelled drawing, Lamb et al [6] assume that lines aligned with one of the three main axes in the drawing correspond to edges aligned with the three main axis in object (i, j, k) space (and implicitly assume, less justifiably, that determining these axes is straightforward).

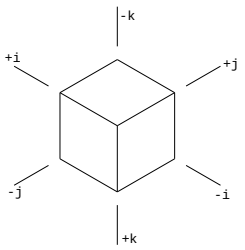

Figure 34. Main Object Axes

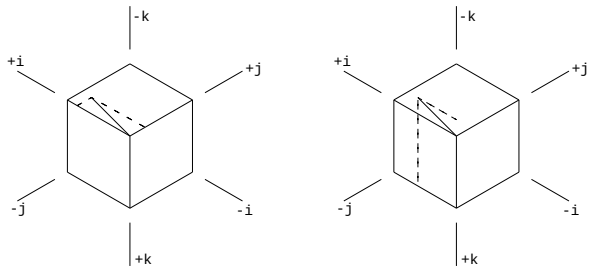

Figure 35. Non-Axially-Aligned Lines in 3D

We apply this idea, with further extensions, to unlabelled drawings, to provide a preliminary frontal geometry which can then be used to predict line labels. Methods for determining the three main axes from a 2D drawing will be given in a later paper; here we simply assume that they correspond to the six specific equispaced directions in Figure 34 with the k axis being drawn vertically.

The axes shown in Figure 34 correspond to a view of the object from above, rather than below. Experimental evidence [8] suggests that humans prefer views from above when interpreting line drawings, justifying this heuristic.

Every other line, aligned *between* any two of the six directions (e.g. $+j$, $-k$), is assumed to correspond to an edge which lies on a plane perpendicular to the remaining axis in space (e.g. i); the direction of the edge in that plane is given by its angle relative to the two directions. For drawings of *normalons* (objects in which all edges and face normals are aligned with one of the three major axes), deviations are due to drawing errors. For drawings of semi-normalons such as those considered in this paper, our assumption should in principle be correct half of the time, as can be seen by reference to Figure 35: the non-axially-aligned line is likely to be in either the ij plane (as we assume, left-hand drawing) or the ik plane (right-hand drawing), but in either case the i-component of the edge vector is larger.

The example in Section 7 illustrates how this idea is used in practice.

The suggestion of Lipson et al [7] that a line which is vertical in the drawing corresponds to a vertical edge in 3D space is a special case of the same idea. Both Lamb et al, and Lipson et al, observe that lines which are the same length in a drawing should correspond to edges of the same length in 3D space. This is correct for the special case of isometric projection which we assume here. Other cases will be discussed in a future paper.

The outline of our new inflation approach is as follows:

• Attempt to identify the three main axes. In this paper, we use the equispaced lines shown in Figure 34 (this is adequate for all but two of the test drawings in this paper). Finding axes by grouping together lines intended to be parallel will be considered in a later paper.

• Create three sets of linear equations (for vertex i-, j- and k-coordinates), by using the assumption of isometricity and by interpolating 3D line directions as above to give the relative (i, j, k) coordinates of the start and end vertices of each line. Solve the three equation systems to obtain vertex positions in (i, j, k) space.

• Determine the best transformation from (i, j, k) space to (x, y, z) space, using the vertex coordinates in (i, j, k) space and equivalent junction coordinates in (x, y) space. Use this to determine the z-coordinate for each vertex, assuming for now that all junctions correspond to vertices.

• If z-coordinates have the wrong sense, they must be inverted. The edges running inwards from the drawing boundary should mainly be coming towards the viewer. If more point away from the viewer, invert all z-coordinates. (Some objects, e.g. pyramids with concave bases, can be drawn violating this assumption, but such drawings would also defeat traditional approaches.)

• Find a best-fit plane corresponding to each region, again assuming that all junctions correspond to vertices and that there is no occlusion. At and near occluding T-junctions this assumption is incorrect, as associated vertices are not in the plane of the face; this will be addressed in future work.

• If a line does not lie in the plane of a region, it occludes that region. Also, if, at a line mid-point, one region is further from the viewer than the other, the line occludes the further region (in practice such decisions cannot be made with certainty, so use probabilities, basing the figure on the difference in angle or z-coordinates). In other cases, find the convexity of the line using the two regions' normal vectors.

This simple method allows many possible extensions, which we will consider in future work.

In the last stage of our approach, we produce probabilities of, rather than decisions about, each line being occluding. Also, our approach makes no use of junction catalogues, useful constraints which we do not wish to abandon. For these reasons, rather than use the line labels predicted by this method directly, we use them to seed the line label probabilities for the relaxation labelling method described above.

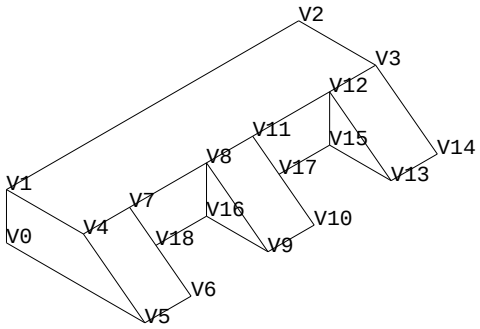

Figure 36.

7 Example

We illustrate the method described in the previous section using Figure 1 as an example. We refer to junctions in this drawing as shown in Figure 36.

For each line of this drawing, we determine the relative i-, j- and k-coordinates of its two junctions from the 2D line angle, as shown in Table 2 (the units are arbitrary). Fixing

Line			Distance		
¿From	To	Length	i-axis	j-axis	k-axis
V0	V1	300	0.00	0.00	-300.00
V1	V2	1900	0.00	1899.61	0.00
V2	V3	500	-499.99	0.00	0.00
V7	V8	500	0.00	499.99	0.00
V4	V1	500	499.99	0.00	0.00
V4	V5	500	-399.53	0.00	300.24
V5	V0	900	899.63	0.00	0.00
V6	V5	300	0.00	-300.17	0.00
V18	V6	286	-228.63	0.00	171.68
V8	V9	500	-399.53	0.00	300.24
V10	V9	300	0.00	-300.17	0.00
V17	V10	286	-228.63	0.00	171.68
V12	V13	500	-399.53	0.00	300.24
V14	V13	300	0.00	-300.17	0.00
V3	V14	500	-400.68	0.00	299.66
V13	V15	400	399.64	0.00	0.00
V15	V12	300	0.00	0.00	-300.00
V9	V16	400	399.64	0.00	0.00
V16	V8	300	0.00	0.00	-300.00
V15	V17	329	0.00	-328.82	0.00
V16	V18	329	0.00	-328.82	0.00
V4	V7	300	0.00	300.17	0.00
V11	V12	500	0.00	499.99	0.00
V8	V11	300	0.00	300.17	0.00
V3	V12	300	0.00	-299.30	0.00
V11	V17	214	-170.90	0.00	128.55
V7	V18	214	-170.90	0.00	128.55

Table 2. Distances along Lines

	Initial		IJK			
Vertex	x	y	i	j	k	z
0	0	1250	0.00	0.00	0.00	1000.00
1	0	950	-10.37	-10.35	-310.45	765.80
2	1645	0	-33.13	1866.46	-333.33	2060.60
3	2078	250	-555.87	1843.67	-356.21	1658.72
4	433	1200	-497.97	2.08	-298.02	438.65
5	779	1700	-889.26	10.35	10.45	386.01
6	1039	1550	-870.66	329.15	29.14	637.79
7	693	1050	-493.82	306.42	-293.84	659.74
8	1126	800	-530.96	769.22	-331.09	934.38
9	1472	1300	-935.83	763.90	-36.24	852.88
10	1732	1150	-913.84	1086.08	-14.15	1111.86
11	1386	650	-530.19	1070.17	-330.30	1148.27
12	1819	400	-567.29	1533.01	-367.52	1422.99
13	2165	900	-979.24	1520.65	-79.68	1326.55
14	2425	750	-967.90	1832.24	-68.11	1563.08
15	1819	700	-603.37	1496.85	-103.65	1558.52
16	1126	1100	-563.52	736.57	-63.73	1077.34
17	1534	864	-663.22	1108.09	-163.74	1198.82
18	841	1264	-623.42	347.77	-123.85	717.56

Table 3. Junction Coordinates

one reference vertex and solving the above as three linear systems gives the set of vertex coordinates shown in the i-, j- and k-columns of Table 3. Rotating the resulting structure to fit, as closely as possible, the original junction x- and y-coordinates produces the set of z-coordinates show in the final column of Table 3. Fitting face planes to each region is straightforward. Finally, the initial line and junction probabilities of a relaxation labeller are derived from the geometry at each line mid-point (omitted due to lack of space).

As noted in Section 8 below, for this example, both the initial probabilities produced above and even the unmodified relaxation labeller probabilities of [18] are sufficient to label the drawing correctly.

8 Test Results

Table 4 uses the Figures to compare the behaviour of (i) the baseline relaxation labelling algorithm from Section 5, and (ii) the same algorithm seeded using the methods from Section 6. It lists the number of incorrectly-labelled lines *(Errors)* and the number of T-junctions misidentified as occluding when not, or vice versa. (The latter make it difficult to reconstruct the topology of the object, so are more serious than mislabelling the convexity of a line).

Although relaxation labelling copes effectively with drawings with a single K-junction [16], it is no more effective than other existing methods for drawings with several adjacent K-junctions. Our new approach is better, reducing the number of mislabellings by over 20%.

However, clearly much more is needed to find a satisfactory method of labelling drawings with multiple K-junctions. Three recurring problems can be seen. Firstly, planes of partial faces in drawings such as Figures 13 and 23 are misplaced because of the incorrect assumption that the T-junction lies on the face; as a result, lines connecting such faces to their neighbours are labelled occluding rather than convex or concave.

Drawing	Previous method		New method	
	Errors	T-errors	Errors	T-errors
Fig. 1	0	0	0	0
Fig. 2	0	0	0	0
Fig. 3	1	1	1	0
Fig. 4	0	0	2	1
Fig. 5	1	0	1	0
Fig. 6	0	0	0	0
Fig. 7	0	0	3	2
Fig. 8	5	1	3	0
Fig. 9	1	0	3	0
Fig. 10	1	0	1	0
Fig. 11	1	0	1	0
Fig. 12	0	0	0	0
Fig. 13	2	0	2	0
Fig. 14	1	0	1	0
Fig. 15	8	0	5	0
Fig. 16	5	0	3	0
Fig. 17	4	0	3	0
Fig. 18	6	0	5	0
Fig. 19	8	0	1	0
Fig. 20	1	0	1	0
Fig. 21	3	0	2	0
Fig. 22	1	0	1	0
Fig. 23	3	0	2	0
Fig. 24	1	0	1	0
Fig. 25	4	1	2	1
Fig. 26	3	1	1	1
Fig. 27	1	0	1	0
Fig. 28	3	1	1	1
Fig. 29	8	0	5	0
Fig. 30	2	0	2	0
Fig. 31	2	1	2	1
Fig. 32	1	2	3	1
Totals	77	8	59	8

Table 4. Test Results

Secondly, determination of those lines which do not lie in the planes of the regions they separate does not result in a strong enough signal that the lines should be labelled occluding, with the result that "obviously" occluding lines in drawings such as Figure 25 are labelled concave; the misplacing of face planes also contributes to this problem.

Thirdly, our approach at present is no more able than any previous approach to distinguish pockets and through holes, and as a result lines in drawings such as Figures 15–17 which appear to be at the bottom of holes are labelled concave, not occluding.

Speed is satisfactory: both approaches can label any of the drawings in a fraction of a second (on a Dell Optiplex SX270 with 3GHz Pentium 4 CPU).

9 Conclusions and Future Work

We have shown that local-constraint methods of line labelling are inadequate for a sizeable class of engineering objects. We have outlined an alternative method, based on creating provisional depth information.

Our initial investigations show this method to be promising. Going beyond the test results already given here, while using a 558-drawing test set [16] to tune numerical parameters, our approach produced 30% fewer mislabellings than did relaxation labelling.

Perhaps the most important advantage of our new approach is that when it goes wrong, the reasons for its failure, being geometric, are apparent and comprehensible. It will thus be easier to adapt and improve than algorithms based entirely on probability propagation.

As noted, there are several possible improvements to the basic implementation given here of our concept as an algorithm, and these will be investigated in our future work.

10 Acknowledgements

Funding for this investigation was provided by Japan Society for the Promotion of Science Fellowship number P03717; this support is acknowledged with gratitude.

References

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

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

[3] D.A. Huffman, *Impossible Objects as Nonsense Sentences,* Machine Intelligence **6** 295–323, 1971.

[4] T. Kanade, *Recovery of the Three-Dimensional Shape of an Object from a Single View,* Artificial Intelligence **17**, 409–460, 1981.

[5] K. Kanatani, Group-Theoretical Methods in Image Understanding, Number 20 in Springer Series in Information Sciences, Springer-Verlag, 1990.

[6] D. Lamb and A. Bandopadhay, Interpreting a 3D Object From a Rough 2D Line Drawing. In ed. A.E.Kaufman, Proceedings of the First IEEE Conference on Visualization '90, 59-66, IEEE, 1990.

[7] H. Lipson and M. Shpitalni, *Optimization-based Reconstruction of a 3D Object from a Single Freehand Line Drawing,* Computer-Aided Design **28**, 651–663, 1996.

[8] P. Mamassian and M.S. Landy, *Observer Biases in the 3D Interpretation of Line Drawings,* Vision Research **38** (18), 2817–2832, 1998.

[9] T. Marill, *Emulating the Human Interpretation of Line-Drawings as Three-Dimensional Objects,* International Journal of Computer Vision **6** (2) 147–161, 1991.

[10] J. Mitani, H. Suzuki and F. Kimura, *3D SKETCH: Sketch Based Model Reconstruction and Rendering*, in ed. U. Cugini and M.J. Wozny, ¿From Geometric Modeling to Shape Modeling, IFIP TC5 WG5.2 Seventh Workshop on Geometric Modeling: Fundamentals and Applications, October 2-4, 2000, Parma, Italy, 85–98, Kluwer, 2001.

[11] P. Parodi, R. Lancewicki, A. Vijh and J.K. Tsotsos, Empirically-Derived Estimates of the Complexity of Labeling Line Drawings of Polyhedral Scenes, Artificial Intelligence 105, 47-75, 1998.

[12] D.N. Perkins, *Cubic Corners,* Quarterly Progress Report 89, 207–214, MIT Research Laboratory of Electronics, 1968.

[13] M.M. Samuel, A.A.G. Requicha and S.A. Elkind, *Methodology and Results of an Industrial Parts Survey*. Technical Memorandum 21, Production Automation Project, University of Rochester NY USA, 1976.

[14] K. Sugihara, *Machine Interpretation of Line Drawings,* MIT Press, 1986.

[15] P.A.C. Varley and R.R. Martin, *The Junction Catalogue for Labelling Line Drawings of Polyhedra with Tetrahedral Vertices,* International Journal of Shape Modelling **7** (1), 23–44, 2001.

[16] P.A.C. Varley, *Automatic Creation of Boundary-Representation Models from Single Line Drawings,* PhD Thesis, Cardiff University, 2002.

[17] P.A.C. Varley and R.R. Martin, *Estimating Depth from Line Drawings*. In Ed. K.Lee and N.Patrikalakis, Proc. 7th ACM Symposium on Solid Modeling and Applications, SM02, 180–191, ACM Press, 2002.

[18] P.A.C. Varley and R.R. Martin, *Deterministic and Probabilistic Approaches to Labelling Line Drawings of Engineering Objects,* International Journal of Shape Modelling **9** (1), 79–99, 2003.

[19] T. Vetter and T. Poggio, *Symmetric 3D objects are an easy case for 2D object recognition*, Human Symmetry Perception, ed C.W.Tyler, 349–359, 1996.