Making the Most of Using Depth Reasoning to Label Line Drawings of Engineering Objects

P. A. C. Varley[1], R.R.Martin[2], H.Suzuki[1]

[1] Department of Precision Engineering, The University of Tokyo, Tokyo, Japan
[2] School of Computer Science, Cardiff University, Cardiff, Wales, UK

Abstract

Automatic creation of B-rep models of engineering objects from freehand sketches would benefit designers. A subgoal is 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 object. Junction and line labels, and provisional depth coordinates, are important components of this frontal geometry.

Most methods for producing frontal geometry use line labelling, but this takes little or no account of geometry. As a result, the line labels produced can be unreliable.

Previously, we proposed an approach which inflates a drawing to produce provisional depth coordinates, and uses these to make deductions about line labels. Even a naïve implementation can outperform previous line labelling methods in certain cases.

In this paper, we further enhance this approach. We extend the algorithm to non-isometric-projection drawings, consider improved ways of realising some of the concepts, and also consider how to combine this approach with other labelling techniques to gain the benefits of each.

We test our approach using to be drawings of what we consider representative samples of engineering objects; these exemplify difficulties not considered in many previous papers on line labelling. Our results, based on this test set, show that the enhancements result in significant benefits.

1. Introduction

Our area of interest is the process of engineering design. Studies such as [7] have shown that designers routinely sketch their new designs on paper before entering them into a CAD package. An automated process for interpreting a sketch as a 3D solid model would enable designers to spend more of their time creatively [7], and, if done within a second or two, would give helpful feedback, further enhancing the designer's creativity [4].

Here we consider specifically the automated production of solid models from *line drawings* which show the *visible edges* of *polyhedral objects* when viewed from a *general position*. Systems for converting freehand *sketches* to line drawings (e.g. see [3]); are not discussed.
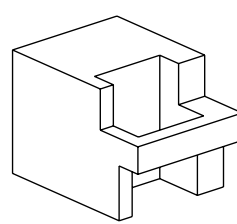
A polyhedron is *trihedral* if three faces meet at each vertex. It is *extended trihedral* [16] if three planes meet at each vertex; there may be four or more faces if some are coplanar. It is *tetrahedral* if no more than four faces meet at any vertex. It is a *normalon* if all edges and face normals are aligned with one of three main perpendicular axes.

Junctions of different shapes are identified by letter: junctions where two lines meet are *L-junctions*, junctions of three lines may be *T-junctions*, *W-junctions* or *Y-junctions*, and junctions of four lines may be *K-junctions*, *M-junctions* or *X-junctions*. Vertex shapes follow a similar convention: for example, when all four edges of a *K-vertex* are visible, the drawing has four lines meeting at a *K*-junction. We assume that vertices in typical engineering objects may be any of the trihedral or tetrahedral types, or may be one of a few common and highly-symmetrical pentahedral or hexahedral types.

We consider the *correct* frontal geometry to be the one which a human would take as the most plausible interpretation of a drawing. We aim to quickly find this correct frontal geometry for line drawings of typical engineering objects. Correctness, in this sense, is surprisingly uncontroversial—for example, the authors of the papers [19, 20] from which our test drawings are taken assume, correctly, that their readers can interpret such drawings as solid objects. This interpretation process is so easy for humans that we do it automatically; it is only when attempting to program a computer to replicate this human skill that we realise its difficulty (see, for example, Palmer [15] for a summary of the current understanding of human perception).

We believe that the reported successes of some approaches to line drawing interpretation can be due to their consideration of cases which are too specific or simple, such as tri-
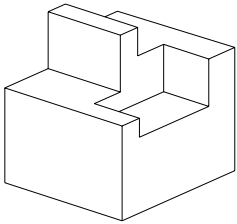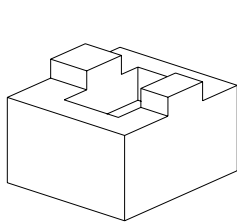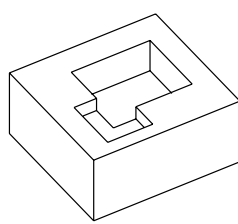
Figure 1:	**Figure 2:**	**Figure 3:**	**Figure 4:**	**Figure 5:**
Figure 6:	**Figure 7:**	**Figure 8:**	**Figure 9:**	**Figure 10:**
Figure 11:	**Figure 12:**	**Figure 13:**	**Figure 14:**	**Figure 15:**
Figure 16:	**Figure 17:**	**Figure 18:**	**Figure 19:**	**Figure 20:**
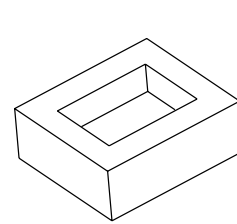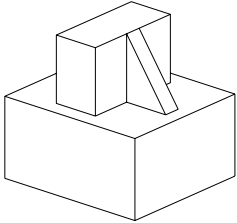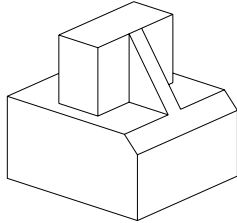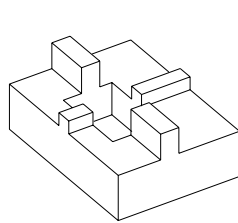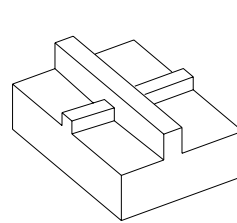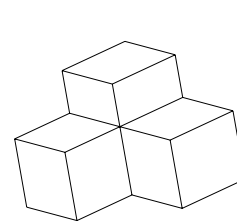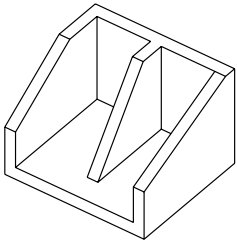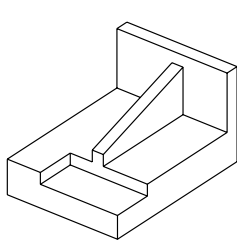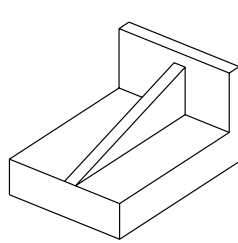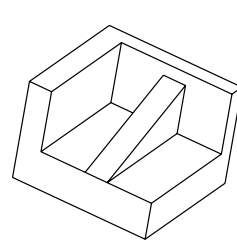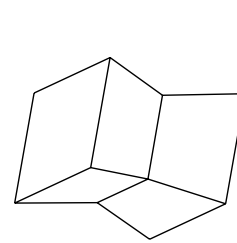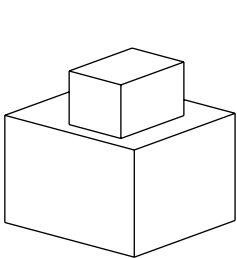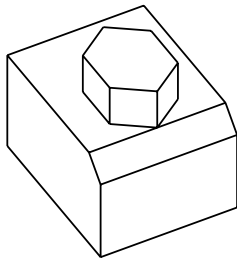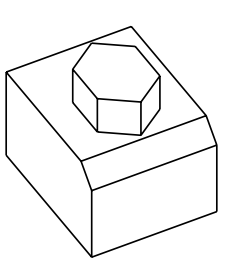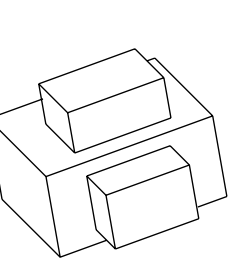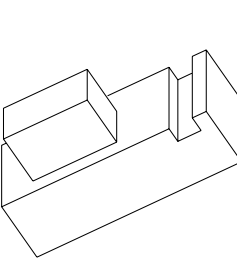
Line Drawings from Sashikumar et al [19, 20]

hedral polyhedra and normalons, rather than to a range of realistic engineering objects. Our test data, shown in Figures 1–20 (and available as the Second Test Set from `http://ralph.cs.cf.ac.uk/Data/sketch.html`), are chosen to avoid this (see Section 7.1).

Limiting our investigations to polyhedral objects is not overly restrictive: by far the most common non-polyhedral features in engineering objects are cylindrical through holes [14] and blends (so common that they are often not included in feature surveys). It is arguably simpler to add these using a CAD package *after* the main polyhedral shape has been created from the sketch. Figures 16 and 19 may not look like engineering objects in their "raw" polyhedral state, but would do with blends and holes added.

For simplicity, we further assume that the user is trying to draw

a real object, not trying to fool the computer by drawing one of several well-known "impossible" objects. Also, we assume that the object is drawn from the "most informative viewpoint"—there is nothing hidden which could not reasonably be deduced from what is visible.

However, we cannot assume that the drawing is perfect—our methods must be tolerant of geometric inaccuracy in the input. Indeed, although all of our test drawings appear acceptable to the naked eye, none of them is mathematically perfect. Typically, the errors in junction positions are far short of the accuracy required by CAD packages.

The correct frontal geometry must conform as far as is geometrically possible to the implications of the lines in the drawing. Sev-

eral methods (see [24]) exist for inferring information from a drawing. These include:

- *Region Identification*: Division of the drawing into 2D areas bounded by loops of lines is trivial. A region may correspond to an entire face of the portrayed object—but they may also correspond to partially-occluded faces or to the background as seen through a hole in the object.
- *Feature recognition*: This is not discussed in this paper except to note its use in Section 5.1.
- *Line Labelling*: Determining whether lines in the sketch correspond to convex, concave or occluding edges is a major topic of this paper.
- *Grouping of parallel lines*, This is a non-trivial problem, discussed in Section 4.1.
- *Inflation*: The addition of *z*-coordinates to the *x-y* coordinates of junctions in the sketch is the other main subject of this paper.

A complete system uses these methods in roughly the order given, so e.g. the outputs of region identification and feature recognition are available as inputs to line labelling and inflation.

A serious problem with existing approaches to line labelling is that the results may not be geometrically realisable [24, 26]. The "traditional" approach [2, 6] uses a catalogue of valid junction labels, and treats line labelling as a local discrete constraint satisfaction problem: the constraints are either 1-node (each junction label must be in the catalogue) or 2-node (each line must have the same label at either end). We [26] have given several examples illustrating that ignoring geometry in this way is inadequate even for some drawings of trihedral objects, and that it is unacceptable when the non-trihedral catalogue is used.

Previously [26], we presented an approach to labelling line drawings which is geometrically-based, and showed that even a simple implementation improves on existing methods when applied to the restricted field of drawings of objects containing *K*-vertices.

We now show (a) how to make further significant enhancements to this concept and (b) that the enhanced concept results in significant benefits when applied to drawings representative of real engineering objects.

Section 2 discusses line labelling: why it is needed, and some of the methods used in previous work. Section 3 outlines our new approach. The various components of our approach are discussed in more detail in Sections 4–6. Section 7 describes our set of twenty test drawings in more detail and gives the results of labelling them using our preferred implementation of this approach. Finally, Section 8 presents our conclusions and suggests paths for future work.

2. Line Labelling

Line-labelling is a well-established preliminary stage of interpreting line drawings [2, 5, 6]. All lines are labelled as either convex (**+**), concave (**−**) or occluding (→). By convention, occluding lines are labelled with the occluded face on the left side of the arrow. Figure 21 shows two labelled drawings.

Here, we discuss why labelling is desirable (Section 2.1) and outline methods for labelling (Section 2.2).
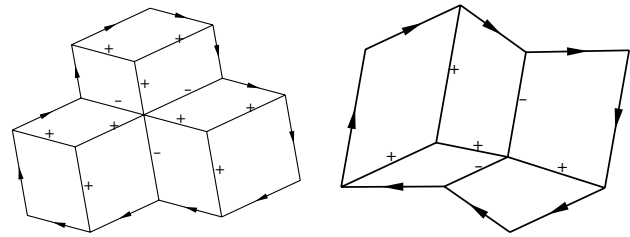
Figure 21: *Line-labelled Drawings*

2.1. Why Label?

The original purpose of line labelling was as a method of identifying and rejecting impossible drawings (a function which, as already noted, does not interest us). However, labelling produces a number of incidental benefits which justify its use in any approach for interpreting line drawings.

Successful labelling provides useful information about the object drawn. Firstly, the line labels indicate which edges bound the visible faces or partial faces of the object and which merely occlude them. For example, labelling Figure 1 indicates that both of the *T*-junctions are occluding, and from this it can easily be deduced that three of the regions correspond to partially-occluded faces.

Secondly, the underlying vertex types implied by the junction labels constrain the possibilities when attempting to reconstruct the hidden topology of the object. In the example of Figure 1, the minimum needed to complete the topology is that two partially-occluded edges must be extended, and seven additional edges must be added to complete the vertices at *L*-junctions.

Thirdly, the junction labels constrain the geometry of any edges to be extended or added. In Figure 1, these constraints, combined with the results of inflation, make it obvious where these nine edges meet. It is clear that this minimum reconstruction is the best one, and that Figure 1 can be interpreted on the basis of a correct labelling and some straightforward deductions based on that labelling.

Determining how to combine the additional edges required to complete Figure 4 is not so straightforward—but it is clear that interpreting a drawing such as this would be much harder without the several clues provided by labelling.

We have also shown that labelling can be a useful input to the process of inflation [23, 24]. The method described there can also be used to improve provisional frontal geometries obtained using our new method.

However, we believe that the most important function performed by labelling is that of distinguishing occluding *T*-junctions from non-occluding *T*-junctions. The differences, both topological and geometric, between the two ways of interpreting *T*-junctions are so fundamental to reconstruction that they must be made at an early stage in the process. For example, no topology can be deduced from an occluding *T*-junction (all we know is that a line is partially-occluded) since there is no vertex at the *xy*-coordinates of an occluding *T*-junction; we can deduce the presence of a non-trihedral vertex (either extended-trihedral or *K*-vertex) at the *xy*-coordinates of a non-occluding *T*-junction. Geometrically, the occluded and

occluding lines have *different* z-coordinates at an occluding T-junction, but have the *same* z-coordinates at a non-occluding T-junction

Even if no other labelling is performed, this distinction must be made in order to create a sensible frontal geometry. We investigate an alternative approach: distinguishing occluding from non-occluding T-junctions without labelling, elsewhere [27].

2.2. Line Labelling History

The usual method of labelling line drawings is by means of a list of valid junction labels, or *junction catalogue* [2, 6]. Combinations of labelled lines meeting at a junction which do not produce a valid junction label can be rejected. The task is thus translated into a discrete constraint satisfaction problem: each line must have the same label throughout its length, and each junction a label in the catalogue.

The Clowes-Huffman [2, 6] catalogue for *trihedral* polyhedra (such as those in Figures 1 and 2) is well-established. Although the limitation to trihedral vertices is somewhat restrictive, Clowes-Huffman line-labelling has been used successfully in applications similar to our own [4].

However, real engineering objects are often not trihedral (Figures 6, 7, 11, 13, 14, 15 and 17 are not). Various extended junction catalogue have been proposed, including ones for simple curved objects [13], for extended trihedral vertices (e.g. [16]), and for all tetrahedral vertices [22]. See [24] for more details of these, and for an overview of non-catalogue-based approaches.

At the core of our labelling methods is the following algorithm, which derives from Kanatani [8]:

- *(Initialisation)*:
- For each junction, candidate labels = set of all valid labels for that junction type;
- For each line, candidate labels = {occluding, such that outside is occluded} if the line is on the drawing boundary, {occluding to left, occluding to right, convex, concave} otherwise
- *(Processing)*
- Loop
- – For each junction with no unique label, eliminate from the sets of candidate labels for neighbouring lines any line labels inconsistent with the remaining candidate labels for this junction
  - For each line with no unique label, eliminate from the sets of candidate labels for the neighbouring junctions any junction labels inconsistent with the remaining candidate labels for this line
  - Exit the loop if a unique labelling has been obtained
  - Exit the loop if the set of candidate labels for any junction or line is empty (no valid labelling can be obtained given the starting conditions)
  - Exit the loop if no candidate labels were eliminated in this iteration (it is likely that there will be multiple valid labellings compatible with the starting conditions)
- End Loop.

Given the "most informative viewpoint" assumption, if there is no non-trihedral junction visible in the drawing, we believe it reasonable to attempt to label the drawing using this algorithm and the

Clowes-Huffman catalogue (see Table 2, first column). If this fails, or if the drawing contains non-trihedral junctions, an approach which can label drawings of non-trihedral objects is required.

The tetrahedral catalogue in principle permits catalogue-based line labelling to be used for drawings of objects with tetrahedral vertices. When using the trihedral catalogue, the proportion of valid junction labellings is so low that most trihedral drawings have only one valid labelling. However, this proportion is much higher when using the tetrahedral catalogue, resulting in a dramatic increase in the number of valid labellings. For example, Figure 1 has only one valid labelling if the Clowes-Huffman catalogue is used, and Figure 2 has two (the central depression is either a pocket or a through hole), but Figure 6 has 337 valid labellings if the catalogue for K-vertices is added (and nearly 1.4 million if the full tetrahedral catalogue is used).

Our experience is that the time taken by traditional labelling algorithms depends more on the number of valid labellings than on the theoretical order of the algorithm. For an interactive system, algorithms which generate all valid labellings are impractical.

We have experimented with two ideas based on the core algorithm presented above. The first is that whenever the core algorithm terminates with the "multiple labels" condition, we use heuristics to select the most promising of the available options to investigate first, and to discard entirely the least promising options. This dependence on heuristics is unsatisfactory, and also this idea can be very slow. The second is to replace the boolean condition (label is possible/impossible) by a probability measure, and boolean elimination by probability multiplication. The resulting probabilistic relaxation algorithm is fast but comparatively unreliable (see Table 2, second column) and the optimal initial probability values are difficult to explain. See [24] for a more detailed analysis of these two approaches.

3. Outline of Approach

We have previously [26] outlined a method which produces both a provisional frontal geometry and suggested line labels which we now summarise, then discuss in more detail:

- We assume that the three main axes (i, j, k) of the object correspond to identifiable groups of (almost-)parallel lines in the drawing. Attempt to identify these groups of lines, as described in Section 4.
- Create three sets of linear equations (vertex i-, j- and $k-$ coordinates) based on line lengths along these axes, as described in Section 5. Solve the three sets of equations to obtain vertex positions in (i, j, k) space.
- Determine the best transformation from (i, j, k) space to (x, y, z) space by minimisation of least-square (x, y) differences, given that we now know vertex coordinates in (i, j, k) space and the equivalent junction coordinates in (x, y) space. Use this to determine a z-coordinate for each vertex (assuming for now that all junctions correspond to vertices).
- There is the possibility that z-coordinates have the wrong sense. Test for this as follows: consider the edges running from the drawing boundary inwards. These ought, in general, to be coming towards the viewer. If they are not, negate the z-coordinates.
- Find a best-fit plane corresponding to each region (again assuming for now that all junctions correspond to vertices and that

there is no occlusion). As it is uncertain whether T-junctions are in the plane of the face, we use a lower weighting for these than for other junctions (a tuning constant in the range 0–1 for T-junctions, 1 for other junctions).

- If, at a line mid-point, one region is clearly further from the viewer (using the plane equations) than the other, the line is (most probably) occluding. Otherwise, determine whether the line is (most probably) convex or concave from the two region normal vectors.

- Use the probabilities so determined to bias the initial probabilities in a relaxation algorithm for producing line labels [24, 25, 26].

Rather than directly use the labels provided by our new approach, we have found it preferable to use them to bias the initial probability values for a probabilistic relaxation labeller [24, 25, 26]. The relaxation labeller acts not only as a labelling approach in its own right, but also as a way of collating the predictions made by other labelling approaches. For each prediction made by one of the support functions, we add to the existing probability value the product of (i) a measure of the confidence the support function has that the prediction is right and (ii) a measure of the confidence the collation function has in the support function. This approach is similar to that in [9] of combining evidence from *support functions*.

One benefit of this information collation structure is that it avoids the need for each information source to always provide sensible information: information sources are allowed to fail. For example, we can use both the new approach described here, and "traditional" Clowes-Huffman trihedral line labelling, as information sources whose output is collated by the relaxation labeller. Clowes-Huffman line labelling fails if the object is not trihedral; the linear systems in Section 5 may also have no solutions. As long as they provide *no* information, rather than *incorrect* information, we are no worse off than the current state of the art (i.e. the relaxation labeller using its default values).

Another advantage is that it allows the junction-catalogue labelling method to perform its traditional function of rejecting impossible interpretations. Typically, in those cases where the provisional frontal geometry is nearly, but not exactly, correct, the confident predictions will be accepted but inconsistent weaker predictions will be overruled.

At present, we have only examined combinations using a limited set of information providers. In view of its success with a limited set of drawings, and the fact that it makes no predictions for drawings outside its scope, Clowes-Huffman trihedral labelling is an obvious candidate: if it gives a unique solution, we increase the merits of the junction and line labels it predicts. We have included this as a possible information provider. Similarly, if Clowes-Huffman trihedral labelling fails but extended trihedral labelling results in a unique solution, we increase the merits of its predicted junction and line labels.

The only other information providers we include at present are the hypotheses of *cofacial configurations* [25], configurations of junctions which imply hole loops corresponding to bosses, pockets or through holes, and those of simple *slot feature configurations* [25]. Labellings matching the predictions of these hypothesis are to be favoured, and thus the initial probabilities of junction and line labels matching these predictions are increased as before.

4. Choice of Axes

To identify from a 2D drawing which lines are parallel to the main axes of an object, we must choose how we believe these 3D axes are represented in 2D. The simplest choice of axes is to assume a standard isometric projection, with $+k$ being vertically downwards, corresponding to the y-axis in 2D, and $+i$ and $+j$ being $120°$ from it in either direction, exactly as shown in Figure 22.
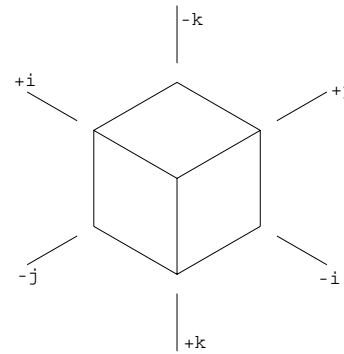
Figure 22: *Three Perpendicular Axes in 2D*

This is simple, as well as being robust and comprehensible—even if it goes wrong, it is usually obvious *why*. Our initial investigations [26] used this assumption with some success.

However, standard isometric projection often fails to meet the requirements of *general position*: vertices and edges may accidentally coincide. To avoid such coincidences, it is often necessary deliberately to deviate from this projection—as is the case for all drawings in this paper. Thus, we next consider a more sophisticated choice of axes based on analysis of lines in the drawing.

We can immediately reject using the longest lines in the drawing to determine the main axes. In practice, it is often diagonal lines which are longest. For similar reasons, using the line with the longest vertical range to determine $+k$ can also be rejected. (Figure 13 illustrates both points).

Assuming for now that we can identify which lines in the drawing correspond to parallel edges in 3D (this is not easy: see Section 4.1), we could also base our 3D axes on the three groups of parallel lines in the drawing which contain either (i) most lines or (ii) the largest sums of line lengths.

If we use this method for the i- and j-axes, it is worth considering whether or not we should use this approach for the k-axis too. Lipson and Shpitalni [12] have suggested that a line which is vertical in the drawing (y-axis) should correspond to an edge which is vertical in 3D space (k-axis), and it may be preferable to keep this idea even if we use more sophisticated methods for i and j.

Another starting point which leads to similar conclusions is to note that that objects are frequently drawn as if resting on a planar surface such as a table. With the exception of Figure 15, all of the test drawings can be imagined to be viewed "at rest". This planar surface contains our i- and j-axes, and leads to the conclusion that when choosing groups of lines for these two axes, we should not consider groups of lines which are "obviously" vertical. Taking

this idea further leads to another way of choosing a group of parallel lines which represent the k-axis: after finding the i- and j-axis directions, find the two directions bisecting them, decide which is nearer vertical, and then choose the line group closest to this direction to give the k direction.

This gives seven methods for comparison:

1. simple method for i, j, k;
2. three most populous groups of lines for i, j and k;
3. simple method for k, two of the three most populous groups for i and j (discarding the one closest to vertical);
4. three geometrically longest groups for i, j and k.
5. simple method for k, two of the three geometrically longest groups for i and j (discarding the one closest to vertical);
6. two of the three most populous groups for i and j (discarding the one closest to vertical), group nearest their bisector for k
7. two of the three geometrically longest groups for i and j (discarding the one closest to vertical), group nearest their bisector for k

In comparing the "simple" variants with the "more intelligent" variants, it is obvious that if the "more intelligent" variants identify the three main axes correctly, they will produce better realisations of the frontal geometry. The question at issue is whether the "more intelligent" variants can be confused and choose the wrong group of parallel lines. The results of using variants 1, 2, 3 and 6 on the 20 test drawings are shown in Table 1. Variants 4, 5 and 7 are not listed, as for all 20 drawings the results using variants 4, 5 and 7 were the same as for variants 2, 3 and 6. The former variants were thus not investigated further.

It can be seen that (with one exception) the diagonal lines in Figure 11 have indeed confused the "more intelligent" variants, and here the variants which fix the k-axis to the y-axis are to be preferred; the exception is the "bisector" variant, which finds the correct group of lines for the k-axis. Naturally, only the "more intelligent" variants note that the k-axis is not vertical in Figures 5, 10, 14, 15 and 19. No variant handles Figure 15 correctly—there is an obvious major axis of the object to which no edges are parallel—but the results produced, while incorrect, are tolerable.

Of the variants tested, our results suggest variant 6 is best, but (as noted) there are also arguments favouring variant 3. In view of this ambiguity, we include a tuning parameter allowing us to interpolate between the two approaches, and subsequent sections assume use of this interpolation.

4.1. Parallel Lines

Parallel lines are possibly the most important as well as the most common regularity visible in 2D drawings—all 20 test drawings contain them—and they provide an important clue to the frontal geometry. It is usually obvious to a human which lines in a drawing are intended to be parallel, so to replicate the user's intentions, it is important to have a method of inferring which lines in the drawing correspond to parallel lines in 3D. If one is to allow for freehand drawing inaccuracies, or uses a relatively weak interpretation of the general viewpoint assumption, this process is tricky.

The first problem is that it is difficult, and sometimes impossible, to set a numerical threshold for parallelism. See, for example,

Figure 15. Clearly, the non-axially-aligned lines should not be parallel to one another. This figure is comparatively well-drawn: the largest angle between pairs of lines which should be parallel to one another is $0.17°$, while the smallest angle between pairs of lines which should not be parallel to one another is $0.78°$. It is easy to see that in a less-well-drawn drawing, some line pairs which "obviously" should not be parallel to one another could be closer in angle than some which "obviously" should be parallel.

A second problem is that there are circumstances where lines may appear to be parallel to one another, within a threshold, but which geometric reasoning tells us cannot be parallel. See, for example, Figure 11. It is clear that the central sloping face cannot be parallel to either of the other two sloping faces. Assuming that the lines at the top of the front U-shaped face are collinear, it also follows that the two end sloping faces cannot be parallel to one another either.

Reasoning of this form is particularly problematic in Figure 15, as conditional reasoning is required. Pairs of the non-axially-aligned lines *could* be parallel to one another, on the assumption that this is a (rather poor) drawing of a cube joined to a quadrilateral frustum.

We previously [21] presented a method for grouping lines based partly on existing 2D parallelism and partly on heuristics concerning which pairs of lines might be expected to be parallel. It required line labels as input in order to deduce line pairs which could not be parallel. Even with labelling information, cases exist where the reasoning required is beyond the capabilities of the method presented. It is unrealistic to expect any method which has less input information to match, let alone exceed, that performance.

Accepting that there is no perfect solution to this problem, we instead use a simple approach, as follows:

- Determine, from the junction shapes, any pairs of lines which either *must* or *cannot* be parallel: "through" lines at T- and K-junctions must be parallel; any other pair of lines meeting at a junction cannot be parallel.
- For every other pair of lines, make a prediction that they *might* be parallel, and determine a merit figure (see [21]) for this prediction based on (a) how close the lines are to being parallel and (b) any clues which suggest that they might be parallel (at present, the only clue we use is that lines which are on opposite sides of a quadrilateral are more likely to be parallel so have a higher merit figure).
- Sort the predictions into descending order of merit
- For each prediction, in descending order of merit,
  - if the prediction that the lines are parallel contradicts an established belief, exit the algorithm, as we have finished
  - if the merit of the prediction is below a threshold, exit the algorithm, as we have finished
  - note that the lines are parallel to one another, and note also that any lines parallel to either are parallel to the other, and that any lines known not to be parallel to either cannot be parallel to the other
- note any remaining lines, not parallel to any other line

This algorithm is not foolproof, and indeed produces incorrect results for Figure 15: the 6 non-axially aligned lines, none of which should be parallel to one another, are grouped into 3 parallel line groups. It also produces incorrect results for Figure 11.

Fig.	True Angles			Variant 1			Variant 2			Variant 3			Variant 6		
	I	J	K	I	J	K	I	J	K	I	J	K	I	J	K
1	288	66	180	300	60	180	287.86	66.00	180.00	287.86	66.00	180	287.86	66.00	180.00
2	288	66	180	300	60	180	288.00	66.00	180.00	288.00	66.00	180	288.00	66.00	180.00
3	316	68	180	300	60	180	316.16	68.00	180.00	316.16	68.00	180	316.16	68.00	180.00
4	294	72	180	300	60	180	294.00	72.00	180.00	294.00	72.00	180	294.00	72.00	180.00
5	304	62	175	300	60	180	303.99	62.00	174.98	303.99	62.00	180	303.99	62.00	174.98
6	288	66	180	300	60	180	288.00	66.00	180.00	288.00	66.00	180	288.00	66.00	180.00
7	288	66	180	300	60	180	288.00	66.00	180.00	288.00	66.00	180	288.00	66.00	180.00
8	310	67	180	300	60	180	310.03	66.99	180.00	310.03	66.99	180	310.03	66.99	180.00
9	310	70	180	300	60	180	310.03	70.57	180.00	310.03	70.57	180	310.03	70.57	180.00
10	285	65	170	300	60	180	284.97	64.99	169.99	284.97	64.99	180	284.97	64.99	169.99
11	295	50	180	300	60	180	295.01	50.00	208.81	295.01	50.00	180	295.01	50.00	180.00
12	292	57	180	300	60	180	292.00	57.00	180.00	292.00	57.00	180	292.00	57.00	180.00
13	292	57	180	300	60	180	292.00	57.00	180.00	292.00	57.00	180	292.00	57.00	180.00
14	295	55	190	300	60	180	295.03	54.92	190.01	295.03	54.92	180	295.03	54.92	190.01
15	285	65	190	300	60	180	272.45	64.98	189.97	272.45	64.98	180	272.45	64.98	189.97
16	285	70	180	300	60	180	285.03	69.99	180.00	285.03	69.99	180	285.03	69.99	180.00
17	320	70	180	300	60	180	320.02	69.98	180.00	320.02	69.98	180	320.02	69.98	180.00
18	320	70	180	300	60	180	320.02	69.98	180.00	320.02	69.98	180	320.02	69.98	180.00
19	315	70	170	300	60	180	314.56	70.00	168.39	314.56	70.00	180	314.56	70.00	168.39
20	325	65	180	300	60	180	325.00	65.01	180.00	325.00	65.01	180	325.00	65.01	180.00

Table 1: *Choice of 2D Axis Angles*

Although the 3 non-axially aligned faces cannot be parallel, the 6 non-axially-aligned edges are grouped together. Attempting to make these edges parallel in 3D would inevitably result in distortions when inflating the object. Worse, as shown in Table 1, this erroneous "group" of 6 edges has been chosen to represent one of the major axes of the object instead of a correct group containing only 5 edges. Far more subtle reasoning than the current state of the art is required to process these two drawings correctly.

It can be noted that all variants we test label Figure 15 correctly, notwithstanding the incorrect grouping of parallel lines. However, with some variants, the incorrect grouping causes problems when processing Figure 11 as it produces a more populous grouping than one of those corresponding to a major axis.

5. Linear Systems

We want an interactive system, and inflation is just one part of that system. Thus, we base the core of our inflation method, the determination of vertex z-coordinates, on direct solution of a weighted linear least-squares problem [1], rather than on iterative non-linear optimisation. Weighting the equations allows us the freedom to experiment by altering the relative importance of different heuristics;

solving by least-squares fit allows us the freedom to add contradictory equations suggested by different heuristics.

We first describe our approach in its simplest form; later, we describe how we generate the additional equations required when the drawing contains more than one distinct subgraph (see Section 5.1), and other additional equations based on deductions about alignment of the visible faces (see Section 6).

The vertex z-coordinates are the variables in the linear system. Most junctions involve just one variable, the z-coordinate of the corresponding vertex. T-junctions may be occluding or non-occluding, so we need two variables, one for the z-coordinate of the (possibly-)occluded line as it passes from view, and the other for the z-coordinate of the occluding line at the same xy-coordinates. If the two z-coordinates come out as similar, this gives a strong hint that the T-junction is non-occluding.

Although parallel line information has been successfully used in inflation [4, 11], we do not use it. Firstly, the parallel line grouping process described in Section 4.1 occasionally makes incorrect groupings, distorting the object, but more importantly, the basic method implicitly makes 2D parallel lines parallel in 3D, and doing so explicitly is redundant.
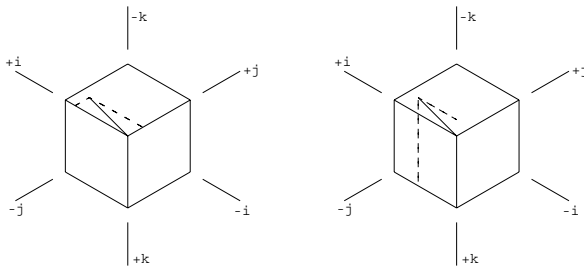
Figure 23: *Axes and Other Lines in 3D*

Figure 24: *Cubic Corners*

The linear systems in their simplest form use one equation for each of the *i*-, *j*- and *k* coordinates for each line, relating the coordinates of the two ends of the line, using the line angle with respect to the axes determined above. This results in three uncoupled linear systems.

For lines which are not close to one of the three axis directions, we must make assumptions as to what they represent. Figure 23 illustrates the two most plausible interpretations of a non-axially-aligned line, the interpolated prediction (left-hand), which in this example would predict an edge in the *ij*-plane, and the extrapolated prediction (right-hand), which in this example would predict an edge in the *ik*-plane. For simplicity, we always use the interpolated prediction. The relative magnitude of the two vector components (in the example, the *i*- and *j*-components) is obtained by forming an axis-aligned skewed rectangle.

Clearly, the implications of lines aligned with the three main axes are more certain than the implications of lines interpreted by interpolation. We account for this by weighting lines differently as follows:

- Where a line is very close to parallel with one of the main axes, the weight of the equation in each of three the linear systems is 1
- Where a line is not close to parallel with one of the main axes, the weights of the equations in the three linear systems are reduced in proportion to the difference in 2D angle between the line and the axis
- An equation with a weight of 0 or less is dropped (note that dropping so many equations that one of the linear systems cannot be solved is not in itself a problem: the consequence is simply that this approach cannot make recommendations; the relaxation labeller still runs, using initial junction and label probabilities provided by defaults and other labelling approaches).

As well as the direction of each edge vector, we must also determine its length. For the fixed-2D-axis variant which corresponds to isometric projection, equal-length lines correspond to equal-length edges.

For the variants from Section 4 in which the 2D alignment of the major axes is variable rather than fixed, the mathematically-correct method of determining 3D lengths of lines is by using the *cubic corner* method [17]. See Figure 24. For any line *VA* meeting a trihedral junction *V* which meets the requirements of a cubic corner, the depth change is given by:
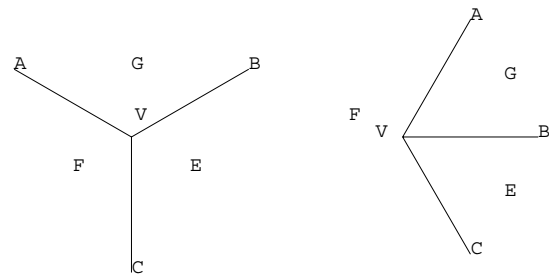
$$|z_A - z_V| = m/\sqrt{(\tan F \tan G) - 1}.$$

where *m* is the 2D length of line *VA*, and *F* and *G* are the 2D angles *AVC* and *AVB*. The requirement which a trihedral junction must meet to be a cubic corner is simply that $(\tan F \tan G) > 1$ [17]. It can be noted that for the special case corresponding to the "fixed angle" variant in Section 4, this reduces to $|z_A - z_V| = m/\sqrt{2}$, the isometricity assumption which we used in [26].

Lacking firm evidence about whether engineers tend to sketch using isometricity (easier to draw but mathematically incorrect) or mathematically-correct cubic corners, we use a tuning parameter to interpolate between the two predictions.

5.1. Subgraphs

In some drawings, the vertex-edge graph divides into disjoint subgraphs, resulting in separation of the systems of equations into discrete subsystems, rendering a unique solution impossible. However, while the relative *z*-coordinates of the two or more distinct groups of vertices can have any possible value, there is usually only one *good* way of relating the depths of the groups.

T-junctions act as subgraph boundaries: the *z*-coordinate of the possibly-occluded line need not be the same as the *z*-coordinate of the occluding line. To try to ensure there are enough equations for a unique solution to the linear system in the absence of any other information, we add very-low-weighting equations equating the two *z*-coordinates of the *T*-junctions; however, this is a last resort, and other, better methods, corresponding more closely to human perception, are preferred whenever possible.

Two distinct categories of drawings have multiple subgraphs. If Figures 8 and 9 were drawn as wireframes rather than natural line drawings, it would be seen that the wireframe is graph-connected; the rear corners of the objects only appear to be isolated because their connections to the rest of the object are occluded. By contrast, the hole loops corresponding to the bosses and pocket in Figures 5, 16 and 19 would still be isolated. The methods for dealing with these two categories differ.

As well as using hypothesised cofacial configurations as a separate information provider, as described above, we can also use them to tie subgraphs together. We use these hypotheses to generate equations in each of the three linear systems which, using 2D (*xy*) geometry, determine the *ijk* coordinates of a vertex on the inner loop relative to those of three neighbouring vertices on the cofacial outer loop. Lamb and Bandopadhay [10] used a similar approach for cases where they deduce that a vertex from one subgraph is in the plane of a region of the other subgraph. However, their method

relies on lines not being unexpectedly occluding, and so does not extend to the non-trihedral case.

For any four coplanar vertices A, B, C and D, an equation can be generated in z_A, z_B, z_C and z_D. Providing BC and BD are non-collinear, BA can be expressed as a linear combination of them: $(A - B) = m(C - B) + n(D - B)$, where m and n can be calculated from the known x and y coordinates of the junctions; rearranging this gives

$$z_A + (m + n - 1)z_B - mz_C - nz_D = 0.$$

In practice, many drawings with viewpoint-specific subgraphs contain lines in the separate subgraphs which are collinear in 2D and which should remain collinear in 3D, e.g. Figures 8 and 9. We add equations to the linear systems to enforce this collinearity.

These methods, while not a complete solution to the problems of separate subgraphs, cover most situations encountered in practice, including all of this paper's test drawings. The practical problems posed by separate subgraphs are not as great as theoretical considerations might suggest.

6. Face Normal Alignment

In many drawings, it is possible to determine not only that some edges are aligned with one of the three major axes, but also that some face normals are similarly aligned. We now consider how to make and use such deductions.

It is simple to add extra equations to our linear systems if we know that a face normal is aligned with the i-, j- or k-axis. For example, if we know that the face normal is aligned with the i-axis, we can set $i_P = i_Q$ for every pair of vertices P and Q on the face. Such extra equations should both make it more certain that the linear systems have a unique solution and that the solution is a good one.

The major problem is determining which vertices are actually in the plane of the face. A junction, in a loop of junctions and lines bounding a region, does not necessarily imply a corresponding vertex in the plane of the face—the face may be only partially-visible, with the vertex being one which occludes the face. Currently, we start by attempting to determine those regions corresponding to fully-visible faces: junctions bounding such regions must correspond to vertices in the plane of the face.

a Any region bounded by lines from *all* groups of parallel lines is not a fully-visible face—e.g., for normalons, it is clear that if the region is bounded by lines in all three main axes, the corresponding edges cannot all be in the same plane.
b Any face which includes the occluded line at an occluding T-junction is only partially-visible. As at this stage we do not know which T-junctions are occluding and which are not, we assume that *any* face including the "tail" of a T-junction is not a fully-visible face.
c At least one of the edges at an L-junction always occludes one or other of the faces it meets (some relatively rare non-trihedral vertices produce L-junctions which occlude both, the two edges occluding different faces). Without a labelling, it is not possible to tell which face is the occluded one (unless the L-junction occludes the boundary). If the lines meeting at an L-junction

separate two non-background regions, we assume that if one region is already known not to be a fully-visible face, that is the occluded one, but failing that, *neither* region is a fully-visible face.

Even this step is not entirely reliable: stage (c) in particular is an uneasy compromise between theoretical soundness and practical utility.

Any fully-visible face including edges aligned with two of the major axis has its normal aligned with the remaining major axis, so we can set the corresponding coordinates in that axis of all vertices in the face equal.

Having identified the fully-visible faces, we try to predict face normal alignment for the partially-visible faces. This is less certain, so we include a merit figure for our predictions.

For each region which is not a fully-visible face,

d count the numbers of pairs of consecutive lines bounding it which are aligned with 2 of the 3 main axes
e count the numbers of lines leaving the face at trihedral (W- or Y-) junctions which are aligned with the 3 main axes
f from this, obtain a count of the data which suggest that the face normal might be aligned with 1 of the 1 major axes, and normalise the 3 counts to obtain likelihood estimates
g if the region shares a line with a face which is already thought to have a normal aligned with one of the major axes, this region cannot have the same normal, so set that likelihood estimate to zero
h predict that the face normal is aligned with the axis corresponding to the highest likelihood estimate, with a merit figure obtained by subtracting the second-highest from the highest likelihood estimate

We note that stage (g) is clearly incorrect: two faces might be parallel if the corresponding regions share an occluding line. This stage too is an uneasy compromise: while noting that there will be cases where a partially-visible face should be parallel to a fully-visible face despite the two regions sharing a line, there will also be cases where we should wish to ensure that two regions, *both* corresponding to partially-visible faces, sharing a line are not parallel, and nothing in our current method prevents this. Figure 2 is an example where the two problems combine to produce incorrect output; Figure 3 is an example where both problems occur without any erroneous effect.

Despite these shortfalls, the predictions of face normals made by this method are correct far more often than they are wrong, and the most serious failures are not a consequence of the deficiencies we have already noted:

- 123 face normals were correctly identified as being certainly aligned to one of the main axes
- 23 face normals of partially-visible faces were correctly identified as being probably aligned to one of the main axes
- 10 face normals were correctly identified as being unaligned
- 20 face normals of aligned faces were not identified as being aligned
- 8 face normals of partially-visible faces were incorrectly identified as being probably aligned to one of the main axes
- 8 face normals were incorrectly identified as being certainly aligned to one of the main axes
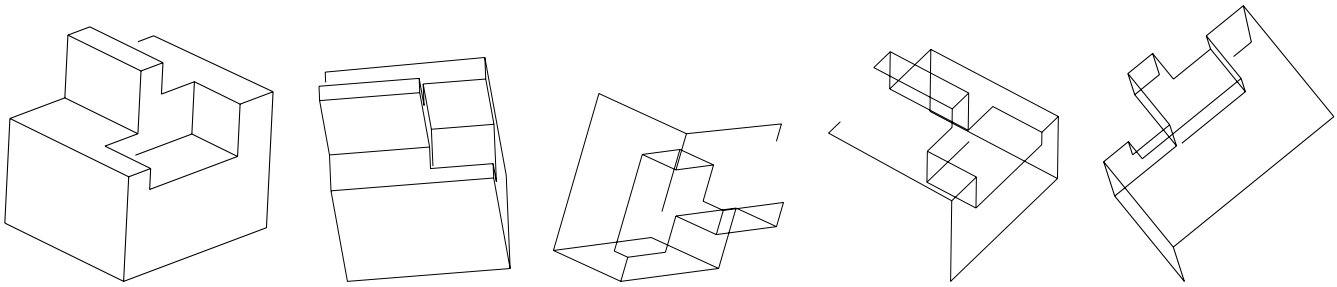
Figure 25: *Inflation of Figure 1*

The drawings in Figures 11 and 15 defeat this method, partly because the inputs from grouping of parallel lines are incorrect. Figures 17 and 18 also defeat this method: incorrect deductions are made from the region corresponding to the boundary of the boss in each of the two drawings.

In conclusion, we note that the major problem with the idea is not how well it works, but that it contravenes one of our earlier assumptions: it is more important for an information provider to give correct recommendations when it gives any recommendation than it is for it to always produce some recommendation. Thus, despite the results in Section 7.2, we cannot recommend its inclusion in any system until the uneasy compromises in stages d and h are resolved.

7. Test Data and Results

7.1. Test Data

Our test data (see Figures 1–20) are taken from two papers by Sashikumar et al [19, 20] concerned with the use of CAD packages. To ensure that they represent a reasonable selection of engineering objects, the drawings were taken without any selection other than to exclude: trivial drawings such as extrusions, repetitions, incomplete drawings, and non-polyhedral objects. Some have been redrawn to ensure a general position viewpoint.

Of the 20 drawings, 12 are normalons (1 of these is extended trihedral, the rest trihedral), 7 can be decomposed into cuboids and axially-aligned wedges (2 of these are trihedral, 5 are tetrahedral with one or more K-vertices, and 1 has a 5-hedral vertex), and 1 (Figure 15), although having a cuboid as its convex hull, is neither a normalon nor built from cuboids and axially-aligned wedges, and has tetrahedral vertices which are not K-vertices. These proportions are reasonably close to those reported in a part survey [18].

6 of the drawings have one or more hole loops; 5 have one or more bosses, 4 have a pocket, and 1 has a through hole; these proportions broadly agree with another survey [14] albeit with more bosses. Considering these matches with the surveys, we believe these 20 drawings are more representative of real engineering objects than the test cases used in many earlier line labelling papers.

7.2. Results

In order to compare the reliability of the approaches described here, we have determined the number of mislabelled lines in each of the twenty test drawings; the results are shown in Table 2. Columns
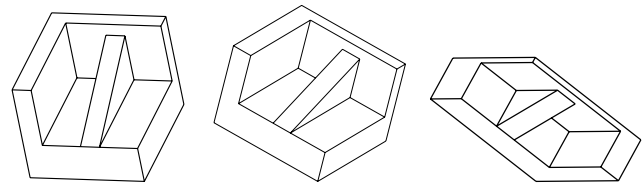

Figure 26: *Inflation of Figure 14*

C-H and Rel are previous methods: the original Clowes-Huffman method for trihedral drawings (or the extended method [16] for extended trihedral drawings), and a probabilistic relaxation approach [25]. Column Fix is the naïve implementation of our new approach described in [26]. Column Var is the preferred implementation of varying the 2D angles of the major axes, as described in Section 4. Column Ded adds to this the deductions concerning face alignment from Section 6. Columns C+F, C+V and C+D are the results of combining Clowes-Huffman trihedral labelling or extended trihedral labelling [16] as additional information providers with the methods of columns Fix, Var and Ded respectively.

Various tuning parameters were optimised separately for each of the variants considered. The data used for optimisation comprised nearly 600 drawings, combining the test data for [24] and [26]. No drawing is in both the optimisation set and the set of 20 test drawings used in this paper. However, some of the drawings in this paper, particularly the simpler ones, are similar to drawings in the larger set.

Figures 25 and 26 show the output of inflating Figures 1 and 14 respectively using the preferred implementation of varying the 2D angles of the major axes. It can be seen that the inflated frontal geometry, while not perfect and not as good as could be achieved if line labels were known in advance, is adequate for the purposes for which we use it here.

The results shown depend both on the variant used and on the settings of the tuning parameters. Tuning a given variant does not always produce a clear optimal setting for each parameter. In some cases, the fact that different results were produced by the different variants may be more to do with the specific optimal values used for the parameters for each variant than with the intrinsic differences in approach between the variants.

There are 380 non-boundary edges in the test set. The best of our new variants labels over 90% of these correctly (unassisted re-

Drawing	C-H	Rel	Fix	Var	Ded	C+F	C+V	C+D
Fig. 1	0	6	1	1	0	0	0	0
Fig. 2	2	8	2	4	2	2	2	0
Fig. 3	2	7	0	2	5	0	2	2
Fig. 4	0	2	2	1	3	1	3	4
Fig. 5	2	3	2	2	2	0	0	0
Fig. 6	n/a	8	6	4	2	0	2	6
Fig. 7	n/a	7	4	2	2	2	2	2
Fig. 8	2	8	6	6	5	0	0	2
Fig. 9	0	4	2	0	4	1	1	2
Fig. 10	0	0	0	0	0	0	0	0
Fig. 11	n/a	8	5	6	8	6	4	13
Fig. 12	0	9	3	1	0	5	1	0
Fig. 13	n/a	6	6	1	0	0	0	0
Fig. 14	n/a	8	5	3	0	0	2	2
Fig. 15	n/a	0	0	0	0	0	0	0
Fig. 16	0	0	1	0	0	0	2	2
Fig. 17	n/a	3	4	1	7	7	3	6
Fig. 18	1	3	4	1	7	7	3	6
Fig. 19	2	1	2	1	2	2	4	4
Fig. 20	0	0	0	0	0	0	2	2
Totals	n/a	91	55	36	49	33	33	51

Table 2: *Test Results: Numbers of Incorrect Edge Labels*

laxation labelling, by far the worst option listed, labels about 75% correctly).

Table 2 does not distinguish clearly *wrong* labellings from plausible but suboptimal labellings e.g. labelling the pocket in Figure 2 as a hole. Such minor mishaps can outweigh real performance differences between variants, so the totals should be taken as being indicative rather than as proof that one variant is superior to another.

Overall, the approach described in this paper is clearly a significant improvement on previous approaches; it is reasonably clear that the variable-axes variants introduced here are to be preferred to the fixed-axis variant (3rd column) we introduced in [26], and that using Clowes-Huffman labelling as an additional support function is worthwhile.

In all cases, the labellings were produced in a fraction of a second (using an Intel Pentium 4 GHz CPU).

8. Conclusions and Recommendations

Line-labelling is useful; without line labels, it is much more difficult to interpret line drawings. However, labelling is a non-trivial problem , especially when non-trihedral vertices are allowed, and no perfect solution is known.

For drawings of trihedral and extended trihedral objects, Kanatani's algorithm [8] for the Clowes-Huffman method, and Parodi's extension, achieve as good results as any other approach. However, the limitation to trihedral and extended trihedral objects is unacceptably restrictive.

It is clear that any of the variants presented in this paper is a significant improvement on the unassisted relaxation approach for general objects. However, selection between the variants is less clear-cut in view of the suboptimal labelling issue noted above. In general, from the variants discussed in Section 4, moveable-axis variants are to be preferred to fixed-axis variants. It is unclear as to whether it is beneficial to incorporate the ideas of Section 6.

The failure modes of our approach are not well-understood at present. Ideally, we should wish to be able to determine, by analysis of the original line drawing, how much confidence can be placed in the outputs of the various stages of processing.

We will continue to look at a variety ideas for combining the requirements of geometric realisability with those of discrete con-

straint satisfaction problems as alternative solutions to the problems of line-labelling, using the variants presented in this paper as a benchmark representing the current state of the art.

9. Acknowledgements

References

[1]   L.Bauer, *Elimination with Weighted Row Combinations for Solving Linear Equations and Least Squares Problems,* Handbook for Automatic Computation **II**, Linear Algebra, eds. J.H.Wilkinson and C.Reinsch, Springer-Verlag 1971.

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

[3]   P.Company, M.Contero, J.Conesa and A.Piquer, *An Optimisation-Based Reconstruction Engine for 3D Modelling by Sketching*, accepted for publication in Computers & Graphics.

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

[5]   A.Guzman, *Decomposition of a Visual Scene into Three-Dimensional Bodies,* AFIPS Proc. Fall Joint Computer Conference, **33**, 291–304, 1968.

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

[7]   D.L.Jenkins, *The Automatic Interpretation of Two-Dimensional Freehand Sketches,* PhD Thesis, University of Wales College of Cardiff, 1992.

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

[9]   J.Kittler and E.R.Hancock, *Combining Evidence in Probabilistic Relaxation,* International Journal of Pattern Recognition and Artificial Intelligence **3**, 29–52, 1989.

[10]   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.

[11]   H.Lipson, Computer Aided 3D Sketching for Conceptual Design, PhD Thesis, Technion-Israel Institute for Technology, Haifa, 1998.

[12]   H.Lipson and M.Shpitalni, *Optimization-based Reconstruction of a 3D Object from a Single Freehand Line Drawing,* Computer-Aided Design **28**(8), 651-663, 1996.

[13]   J. Malik, *Interpreting Line Drawings of Curved Objects,* International Journal of Computer Vision **1**, 73–103, 1987.

[14]   B.I. Mills, F.C. Langbein, A.D. Marshall and R.R. Martin, *Estimate of Frequencies of Geometric Regularities for use in Reverse Engineering of Simple Mechanical Components*, Technical Report GVG 2001-1, Geometry and Vision Group, Department of Computer Science, Cardiff University, 2001.

[15]   S.E. Palmer, *Vision Science. Photons to Phenomenology*, MIT Press, 1999.

[16]   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.

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

[18]   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.

[19]   V.Sashikumar and M.Sohoni, *Reconstruction of Feature Volumes and Feature Suppression*, in ed. K.Lee and N.Patrikalakis, Proceedings, Seventh ACM Symposium on Solid Modelling and Applications SM'02, 60–71, ACM Press, 2002.

[20]   V.Sashikumar, M.Sohoni and R.Rajadhyaksha, *Removal of Blends from Boundary Representation Models*, in ed. K.Lee and N.Patrikalakis, Proceedings, Seventh ACM Symposium on Solid Modelling and Applications SM'02, 83–94, ACM Press, 2002.

[21]   P.A.C. Varley and R.R. Martin, *Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch— Sketch Categorisation and Frontal Geometry,* in ed. H.I.Choi, M.S.Kim, K.W.Lee and R.R.Martin, 1st Korea-UK Joint Workshop on Geometric Modeling and Computer Graphics, 113–128, Kyung Moon Publishers, 2000.

[22]   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.

[23]   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.

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

[25]   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.

[26]   P.A.C. Varley, H. Suzuki and R.R. Martin, *Interpreting Line Drawings of Objects with K-Junctions*, Submitted to GMP 2004.

[27]   P.A.C. Varley, R.R. Martin and H. Suzuki, *Frontal Geometry of Engineering Objects: Is Line Labelling Necessary?* Submitted to SMI 2004.