

Estimating Depth from Line Drawings

P.A.C.Varley and R.R.Martin
Dept. of Computer Science, Cardiff University
Cardiff, Wales, UK
{Peter.Varley, Ralph.Martin}@cs.cf.ac.uk

ABSTRACT

Our goal is unassisted machine interpretation of a single line drawing of an engineering object (with hidden lines removed) as a B-rep model. As part of this process, we seek to deduce a *frontal geometry* of the object, a 3D geometric realisation of that part of the object visible in the drawing.

Inflation takes a drawing in which all lines have been line-labelled, and creates the frontal geometry by adding a z -coordinate to the x - and y -coordinates of each junction. This depth information comes from *compliance functions*, interpretations of drawing features expressed as equations in junction z -coordinates. We examine several compliance functions, and assessing their use in interpretation of engineering objects. We also describe a compliance function based on junction labels, and remove its previous restriction to trihedral vertices.

We give a comparative analysis of applying combinations of compliance functions to a set of test drawings. As a result, we recommend using edge parallelism in combination with either corner orthogonality or junction label pairs, the latter being more reliable in general. Additional use of face planarity compliance is often beneficial and even necessary.

Categories and Subject Descriptors

J.6 [Computer-Aided Engineering]: CAD

General Terms

Algorithms, Experimentation

Keywords

sketch input, B-rep models

1. INTRODUCTION

Our goal is unassisted machine interpretation of a single line drawing (with hidden lines not shown) as a solid object, in order that we can automatically create B-rep models of

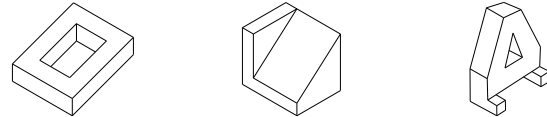


Figure 1: Natural Line Drawings

engineering objects from freehand sketches. To provide later stages of this process with full information, 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 that part of the object visible in the drawing. The depth information so produced is used in estimating the merit of various hypotheses made by later stages of processing about the drawing.

Clearly, it is impossible to uniquely calculate the depths of junctions in a single line drawing: there are, in principle, an infinite number of objects which the drawing could represent. Notwithstanding this, humans can interpret line drawings, and can usually reach a consensus about the depth implications: most readers, for example, will interpret the drawings in Fig. 1 in the same way.

Interpretation of line drawings is a complex problem requiring assumptions. We distinguish between *recognition*, where the object in the drawing is compared with a set of known objects, and *reconstruction*, where an attempt is made to realise the drawing as a 3D object even though that object may be unknown. We aim for reconstruction.

Secondly, one must decide how many drawings of the same object are required, and the viewpoints from which they are made. The common choices here are: three drawings made from orthogonal viewpoints along the principal axes of the object [17], multiple—often only two—drawings from close but non-identical viewpoints (this approach is the one currently favoured in interpretation of aerial photographs [15]), and (our own preference) a single drawing made from a *general position* viewpoint.

Thirdly, one must decide on whether to interpret *natural line drawings* (e.g. Fig. 1) or *wireframe drawings*, in which lines occluded by the material of the solid object are nevertheless shown in the drawing. Our preference, justified in [28], is for natural line drawings; others [5, 12] are investigating the problems of interpretation of wireframe drawings.

Fourthly, one must decide whether or not one is interested in the problem of *realisability*: is the line drawing realisable as a 3D object or not? Our assumption is that the user has an object in mind when creating the line drawing, so realis-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SM'02, June 17-21, 2002, Saarbrücken, Germany.

Copyright 2002 ACM 1-58113-506-8/02/0006 ...\$5.00.

ability is not an issue (the current state of the art concerning this problem is described and advanced by Ros [20]).

A key initial stage in the interpretation of natural line drawings is *line-labelling* [4, 9], in which all lines in the drawing are labelled as convex, concave or occluding. We discuss this non-trivial problem in another paper [27], where we show that although there is no theoretically satisfactory algorithm, in practice most natural line drawings can be labelled in the way a human would expect. In this paper, we shall assume that all line drawings can be, and have been, labelled correctly.

In our application, we wish to assign preliminary depth coordinates to each vertex in a line drawing. This depth information will be used to estimate the merit of various hypotheses we make about the drawing. Ideally, we wish to produce depth information: *quickly*: depth estimation is one of several stages of an interactive system; *robustly*: in a way insensitive to inaccuracies in the drawing; *without prejudice*: using as little prior processing of the drawing as possible (we wish to use the depth information to test our hypotheses, so the depth information should not incorporate these hypotheses if this can be avoided); as *accurately* as possible, while achieving our other objectives.

We assume that the 2D coordinates x_v, y_v of each vertex v are known (these coordinates come from the original drawing and may or may not be accurate), as are the vertices terminating each edge, and the loops of edges forming each face. Also, all pairs of edges presumed to be parallel in 3D may be known. *Inflation* uses any of this information which may be relevant, translates it into *compliance functions* expressed as equations, and finds the best solution to the resulting system of equations. The outputs are depth (z -)coordinates for each visible vertex and for each point at which a partially-occluded edge disappears from view.

This paper examines several known compliance functions, assessing their relevance to interpretation of engineering objects. It also gives the first rigorous presentation of a compliance function based on adjacent pairs of junction labels, and removes the previous restriction to trihedral vertices. This paper also presents the first comparative analysis of results obtained by applying combinations of compliance functions to a set of test drawings. In most cases, for inflation we recommend using a combination of edge parallelism with either corner orthogonality or junction label pairs, with the latter being more reliable in general. Use of the face planarity compliance function in combination with these is often beneficial and sometimes necessary.

This paper does not describe “beautification”, in which an existing 3D model is improved by adjustment of face equations or vertex coordinates. Thus, we do not describe optimisation methods which rely on a preprocessing stage to identify approximate depth coordinates (thereby ensuring that downhill methods start in the right valley); nor do we cover methods which iteratively detect and remove incompatible compliance functions, nor methods which change the x and y coordinates of visible junctions. Many past systems contain only one stage which determines z -coordinates and therefore do not make the distinction between inflation and beautification; this paper discusses those which emphasise analytical rather than optimisation-based methods.

Section 2 summarises prior work. Section 3 considers compliance functions in more detail, including the new Junction Label Pairs (JLP; Section 3.15) approach. It is preferred be-

cause it can be used directly in a system of linear equations in which the unknowns are the values sought, and because the results are intuitively plausible.

Section 4 considers a recent depth estimation method in more detail, identifying weaknesses which make it inappropriate for initial inflation. Section 5 describes a method better-suited to the specific requirements of preliminary inflation. Use of JLP as the primary compliance function in any inflation system is new (as is a z -coordinate linear least-squares approach using corner orthogonality (Section 3.6) and line parallelism (Section 3.3)).

Section 6 demonstrates that JLP gives acceptable results and compares its effectiveness with corner orthogonality. The effects of some secondary compliance functions on the quality of output are also analysed.

2. OVERVIEW AND HISTORY

We wish to infer depth coordinates for each vertex in a line drawing. All we know are the 2D coordinates x_i, y_i of each vertex i , and the vertex pairs joined by each edge. From the latter, it is usually possible to deduced which vertices lie on which face. We wish to collect any information which may be relevant, translate it into equations, and find the best solution to the resulting system of equations. Methods which have been tried fall into two basic categories. Firstly, we may choose to limit our information to that which can be translated into a linear system of equations, and find the optimum solution by linear algebra. Alternatively, we may choose to include non-linear equations, and find a solution using an iterative optimisation process. There are also two possible targets. Here, an approximate geometry is preferred, being the “best” (least squares) fit to all compliance functions. A fully-correct geometry in which, e.g., all faces are exactly planar may require adjustment of the x_v, y_v coordinates to enforce this and may require analysis to detect and eliminate incompatible compliance functions from the solution; neither of these is desirable when estimating frontal geometry.

The simplest linear method, used in our earlier work [25], is to use a system of equations linear in just one set of variables z_v , where z_v is the depth coordinate of vertex v .

A more complex method [8] is to include coefficients in the face equation $P_f x_v + Q_f y_v + z_v + C_f = 0$ for any combination of vertex v and face f where the vertex lies on the face; the output variables are P_f, Q_f and C_f for each face and z_v for each vertex.

Optimisation aims to minimise an *objective function* (a weighted sum of compliance functions) describing the drawn object. It is often the case that objective functions are non-linear; when this is so, optimisation methods can be subdivided into those which adjust all vertex depths simultaneously (usually using a black-box optimisation algorithm) [5, 6, 11, 14] and those which adjust the vertex depths one by one [1, 12, 13].

A further refinement, due to Leclerc [11], is to introduce into the objective function a parameter λ which increases from 0 to 1 as the optimisation process progresses; the overall objective function becomes $F = F_A + (1 - \lambda)F_B + \lambda F_C$, where F_A are those parts of the objective function which must *always* be satisfied, F_B are those parts of the objective function used to inflate the (originally flat) line drawing into 3D, drawing the solution towards the global minimum, and F_C are those parts of the objective function used to fine-

tune the solution once it is reasonably close to the global minimum. Although successful in practice, this refinement blurs the distinction between the problems of inflation (F_B) and beautification (F_C).

As noted in the introduction, methods using black-box optimisation are considered more suitable for beautification than initial inflation and are not considered in this paper. As an example of the one-by-one approach, Barrow [1] uses an iterative optimisation scheme which slides the z coordinates of vertices in and out along the z -axis in order to maximise or minimise a single compliance function. He prefers one of three global measures of regularity:

- sum of squares of angles between faces,
- sum of squares of cosines of angles between faces,
- sum of squares of $(2\pi - \text{sum of angles at a vertex})$.

He reports that all three measures produce similar results.

Lipson [12] provides a list of compliance functions which he used: face planarity, line parallelism, line verticality, isometry, corner orthogonality, skewed facial orthogonality, skewed facial symmetry, line orthogonality, minimum standard deviation of angles, face perpendicularity, “prismatic face”, line collinearity, planarity of skewed chains. These are described in the following section. The compliance functions are weighted according to their degrees of accuracy in free-hand drawings, and depth values are found by performing a least-squares fit. He examined several optimisation methods and prefers cyclic application of Brent’s method [2] to each vertex in turn. Reasonable preliminary estimates are needed in order to limit optimisation time.

Choice of method may depend on the application: the relative importance of speed and accuracy varies from one application to another, e.g. whether depth estimation is the final objective [5, 6] or merely a stage in a larger system [25]. Another point to be considered is the sensitivity of the process to errors in the input: this may be unimportant if the input can be assumed (or preprocessed) to be error-free, but may be crucial if (as in our application) it comes from a freehand line drawing. There may also be differences in requirements between systems which process wireframe drawings of entire objects [11, 12, 13, 5, 6] and those which process line drawings of the front (visible) parts [8, 25].

3. COMPLIANCE FUNCTIONS

This section describes various compliance functions. Mathematical detail is included only for those which are self-contained or which could form part of a linear system.

3.1 Approach: Use of Mirror Symmetry

Vetter [29] notes that if a polyhedral object is known to have an axis of bilateral symmetry, the entire object can be *recognised* from one 2D drawing, given the general position assumption, and providing that at least four pairs of bilaterally symmetric points can be identified. This is not true for *reconstruction*: hidden topology bisected by the mirror plane cannot be deduced from mirror symmetry alone. E.g., although it is clear that there should be an edge descending behind the object from vertex A in Fig. 2, neither the original drawing nor its reflected equivalent contains this edge. Other methods must be used to deduce its presence.

On implementing this approach we found that it is not an improvement on the more general methods described elsewhere in this paper. Errors in the generated reflected geometry magnify inaccuracies in the original drawing. It is

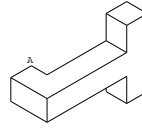


Figure 2:
J-Block

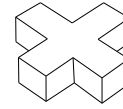


Figure 3:
X Block

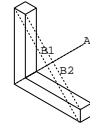


Figure 4:
L Block

possible to include the z -coordinates predicted by this approach in a linear system, but in view of the poor performance of mirror planes in predicting geometry, this idea is not promising and has not been investigated in detail.

3.2 Facial Planarity

Let each face f lie in the plane $P_f x + Q_f y + R_f z + C_f = 0$, and each vertex v have coordinates (x_v, y_v, z_v) . Since we assume no face is parallel to the projection direction, $R_f \neq 0$, so it is possible to set $R_f = 1$ to normalise the equation: $P_f x_v + Q_f y_v + z_v + C_f = 0$. This equation is the basis of Grimstead’s linear system method [8] (see Section 4).

Alternatively, for any four vertices A, B, C and D on a face, an equation can be generated in z_A, z_B, z_C and z_D to make A coplanar with the other three. Providing BC and BD are non-collinear, BA can be expressed as a linear combination of the two, i.e. so $(A - B) = m(C - B) + n(D - B)$, where m and n can be calculated from the known x and y coordinates of the vertices; rearranging gives

$$z_A + (m + n - 1)z_B - m z_C - n z_D = 0$$

which could be used in any linear system in which the unknowns include depth coordinates.

This does not extend uniquely to non-quadrilateral faces, for which more than one equation must be generated. As Lipson [12] points out, enforcing coplanarity of any four points on a face does not enforce global face planarity. Furthermore, it must be ensured that BC and BD are non-collinear, which is not always straightforward if line drawings may include non-trihedral vertices.

An alternative approach is to enforce face planarity after inflation by taking face equations as the input data and placing vertices at the intersections of the appropriate faces. Face equations could be obtained directly (using Grimstead’s linear system) or by finding the best fit to vertices lying on the face (following use of a z -coordinate linear system).

Evidently, any use of facial planarity requires knowledge of which vertices lie on which faces. For natural line drawings of trihedral polyhedra, this is straightforward since all T-junctions in the drawing are occluding. For non-trihedral polyhedra, a labelling is required in order to distinguish occluding from non-occluding T-junctions. The labelling problem is itself non-trivial, and is discussed in [27]. If the drawing is also permitted to contain hole loops, further prior processing is required in order to ensure that cofacial loops are identified (all loops belonging to a face must be coplanar).

This compliance function will not inflate a drawing into 3D by itself—clearly, $P_f = Q_f = C_f = z_v = 0$ for all (f, v) solves one linear system and $z_A = z_B = z_C = z_D$ for all (A, B, C, D) solves the other—so, if used at all, it must be combined with an inflationary compliance function.

3.3 Parallel Lines

If it is believed that lines AB and CD are parallel in 3D,

it is straightforward to generate equations to encourage this. The lengths m of line AB and n of line CD can be calculated from the x and y coordinates, giving the equation

$$nz_A - nz_B - mz_C + mz_D = 0$$

which is linear in z -coordinates.

This function requires either knowledge of which pairs of lines in 2D correspond to parallel edges in 3D, or a weighting which should be applied to the equation reflecting confidence in the assumption of parallelism. We give a prototype scheme for deducing this knowledge in [24]; it has more recently been extended to allow for non-trihedral vertices [28]. This scheme does not require any prior processing of the drawing, although line labelling and common feature identification can be used to make it more robust.

Again, this function will not inflate a drawing by itself. It is useful as a secondary component of an inflation system, both for tidying the output and for ensuring that the system includes equations in occluded line coordinates (see e.g. Fig. 10).

3.4 Vertical Lines

Lipson [12] suggests that a line which is vertical in the drawing should correspond to an edge which is vertical in 3D space. This suggestion is rejected. The distinction must be made between the viewer's 2D (x, y) coordinate system, the viewer's 3D (x', y', z') coordinate system obtained by adding a depth axis z' , and the 3D (u, v, w) coordinate system of the main axes of an arbitrarily-oriented normalon. A vertical line in the (x, y) coordinate system does not usually correspond to a vertical line in the (x', y', z') coordinate system (distance from the viewer changes along the length of the line), although the corresponding edge often is aligned with one of the three axes of the (u, v, w) coordinate system.

To illustrate the point, the cube in Fig. 7 is the simplest of many counter-examples in this paper which emphasise the difference between "vertical in 3D space" (i.e. parallel with the y' -axis) and "perpendicular to a base plane" (i.e. parallel with the w -axis). The assumption of general position requires that these two axes cannot be parallel.

3.5 Isometry

Lipson [12] observes that lines which are the same length in the drawing should correspond to edges which are the same length in 3D space. This is not useful in initial depth estimation, where it is the qualitative issue of which vertices are nearer than their neighbours, rather than the quantitative issue of by how much, which is important.

3.6 Corner Orthogonality

A *cubic corner* [18] is a trihedral vertex at which the three faces are aligned with the three coordinate axes. Criteria for determining whether a W-junction or Y-junction can be an accurate drawing of a cubic corner were established by Perkins [18]: for a W-junction, the two smaller angles must both be acute; for a Y-junction, all three angles must be obtuse. The proof of this result involved equations for the ratio of depth change along a line to the 2D length of any line VA at a cubic corner V which is a W-junction or Y-junction and which is linked by edges to vertices A , B and C :

$$\frac{|z_A - z_V|}{m} = \sqrt{\frac{-\cos \beta \cos \gamma}{\cos \alpha}},$$

where m is the 2D length of line VA , and α , β and γ are the 2D angles BVC , AVC and AVB .

Rearranging and simplifying provides an equation which could be used in a linear system:

$$|z_A - z_V| = m\sqrt{(\tan \beta \tan \gamma) - 1}.$$

In order to use this method, there must be a separate mechanism for determining whether A is in front of or behind V . The method fails for junctions which do not meet the Perkins criteria, e.g. those found in oblique projections. Experimental results in Section 6 show that the output can be poor if the object drawn is not a normalon.

This compliance function has also been used successfully for normalons by fixing a single vertex and propagating depth knowledge along edges [7].

3.7 Skewed Facial Orthogonality

Kanade's method [10] calculates the normal (P, Q, R) of a face given two axes on it believed to be perpendicular in the 3D world, applied to those corners of faces which could be right-angles. The result is general for any two lines at angles α and β to the horizontal (the angles α and β of the two axes chosen such that the angle δ between them is obtuse) providing that the lines are in the plane of the face and perpendicular in 3D.

Kanade [10] notes that the vectors \mathbf{a} and \mathbf{b} , corresponding to the actual 3D directions of lines α and β respectively, must obey $\mathbf{a} \cdot (P/R, Q/R, 1) = \mathbf{b} \cdot (P/R, Q/R, 1) = 0$, and the belief that they are perpendicular translates to $\mathbf{a} \cdot \mathbf{b} = 0$. From this, he obtains the result:

$$\begin{aligned} & (\cos \alpha \cos \beta + \sin \alpha \sin \beta) + \\ & \left(\frac{P}{R} \cos \alpha + \frac{Q}{R} \sin \alpha \right) \left(\frac{P}{R} \cos \beta + \frac{Q}{R} \sin \beta \right) = 0 \end{aligned}$$

and it follows that

$$\frac{P}{R} = \rho \cos \frac{\alpha + \beta}{2}, \quad \frac{Q}{R} = \rho \sin \frac{\alpha + \beta}{2}, \quad \rho = \frac{\sqrt{-\cos(\alpha - \beta)}}{\cos \frac{\alpha - \beta}{2}}.$$

The derivation assumes a correct orthographic projection (e.g. as in a photograph of a real object). However, inaccurate or non-orthogonal projections, such as are found in line drawings, can cause problems.

Skewed facial orthogonality, and deskewing methods in general, produce two possible face normals for each face. Choice between these must be based on other reasoning.

This method fits easily into an extended linear system incorporating P_j and Q_j , (although since separate equations are produced for P_j and Q_j there is the danger that the two might be decoupled), but does not fit naturally into the minimal linear system in z_i . Skewed facial orthogonality thus has no advantages over corner orthogonality, and is both more complex and less flexible, so is not recommended.

3.8 Skewed Facial Symmetry

This method uses Kanade's original idea [10], i.e. to apply the equations given in the previous section to a face believed to show a skewed version of mirror symmetry, as in Fig. 4, where lines $B1$ and $B2$ are perpendicular to A . In addition to the problems noted above, in order to use this method for skewed mirror symmetry, candidate mirror symmetries (and the weightings attached to them) must be identified before depth estimation. This would require that potential

symmetry be identified before initial inflation, which contravenes one of the purposes of inflation (depth information will be used to assess candidate symmetries).

3.9 Line Orthogonality

Lipson [12] suggests that consecutive lines in the same face, other than those evidently collinear, should be made perpendicular in 3D space. The assumption is questionable, its implementation using skewed facial orthogonality would suffer from all of the disadvantages noted in the previous two sections, and a more restricted version would reduce to the corner orthogonality equations described earlier.

As “minimum sum of dot products” at a vertex, this compliance function has been used with success to provide initial inflation in a λ -style optimisation, but that approach has already been rejected in Section 1.

3.10 Minimum standard deviations of angles

Marill [14] noted that the natural interpretations of convex polyhedra tend to be those with the minimum standard deviations of angles (MSDA) at corners on faces. This method is relatively successful for drawings which meet his assumptions, but the assumption of convexity is too limiting for the method to be of general use.

Since MSDA is a property of the object as a whole, not a local property, it cannot be incorporated in a linear system approach. It is not ideal even for the optimisation approach which adjusts a single vertex at a time, since the MSDA for the entire object must be recalculated after each adjustment.

3.11 Face Perpendicularity

Lipson [12] suggests that all dihedral angles should initially be made 90° . This could be regarded as an improvement on cubic corners in that, as it uses edges rather than vertices, it is unaffected by the presence of non-trihedral vertices. This compliance function would be a useful way of providing initial inflation in a λ -style optimisation, an approach already been rejected here.

3.12 Prismatic Face

Under this title, Lipson [12] includes the various factors which contribute to a right extrusion, in particular planar end caps and rectangular side faces. The geometric implications of these are not qualitatively different from face planarity and line parallelism. Weightings for these compliance functions could be increased for potential extrusions, but this contradicts the requirements by prejudging that the object drawn is indeed an extrusion rather than a frustum.

3.13 Line Collinearity

Lipson [12] suggests that lines which are collinear in the drawing should correspond to collinear edges in 3D space. This is intuitively plausible (a stronger definition of general viewpoint would make it necessary), but it is unclear whether inflation should enforce this. E.g., in Fig. 49, line collinearity is a consequence of the object’s mirror symmetry, and the presence or otherwise of such symmetry is one of the things depth information will be used to assess; it could be argued that in such cases including line collinearity equations is premature. It is clear that line collinearity equations could potentially improve inflation of Fig. 5, but to no practical benefit: neither identification of mirror symmetry nor classification of the object as a normalon extru-

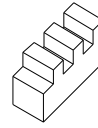


Figure 5:
Collinear Lines

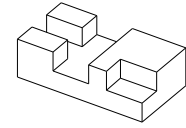


Figure 6:
Normalon [16]



Figure 7:
Standard
Isometric



Figure 8:
Slanted
Isometric

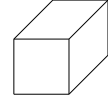


Figure 9:
Diagonal
Isometric

sion will be affected. A good case can be made for inclusion of line collinearity equations where the lines are in different subgraphs, as in Fig. 6, as a means of relating depth coordinates of the two subgraphs.

3.14 Planarity of Skewed Chains

As we described in a previous paper [24], adjacent faces sharing an edge with mirror symmetry can be chained. This imposes additional constraints on the geometry of the object. Since these constraints are non-linear in depth coordinates, require identification of symmetry before they can be generated, and depend on the topology of the object as a whole, they are more suited to optimisation-based approaches and are not discussed in this paper.

3.15 Junction Label Pairs

Although geometrically sound, existing compliance functions take little account of human perception. The junction label pair (JLP) function [24] is an attempt to remedy this. The original implementation suffered from a number of drawbacks, on which this paper improves.

As noted earlier, there is little disagreement about what the drawings in Fig. 1 represent. We believe that this is because of shared processes in human interpretation of line drawings, and that one of these resembles junction labelling (e.g. in the left-hand drawing of Fig. 1, the lower Y-junction is interpreted as all-convex, leading to the inference that “therefore this vertex is nearest the eye”). Such interpretations form the basis of the JLP compliance function.

In Fig. 7, a drawing of a cube in isometric projection, all lines are either Lba to $Wbca$ or $Wbca$ to $Yccc$. (Junction labels are defined in [27]: briefly, the suffices mean (a) occluding, with occluded region clockwise from the line, (b) occluding, with occluded region anticlockwise, (c) convex, (d) concave. Suffices are ordered clockwise.) It is clear that the $Yccc$ junction is nearer than the $Wbca$ junctions, which in turn are nearer than the Lba junctions [24]. In either case, the ratio of change of depth to 2D line length is $1/\sqrt{2}$. Depth information for the visible part of a cube, or any other axis-aligned drawing in exact isometric projection which uses only these three junction types, can be recovered precisely from this knowledge.

In [24], only these two JLP ratios were derived geometrically; all others were obtained by statistical analysis of

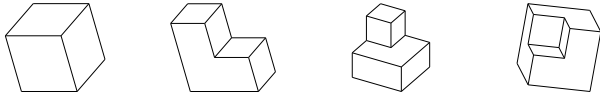


Figure 10: Figure 11: Figure 12: Figure 13:
Huffman-Clowes Objects [4, 9]

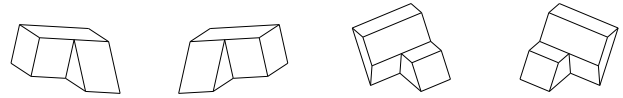


Figure 25: Figure 26: Figure 27: Figure 28:
K-type Objects

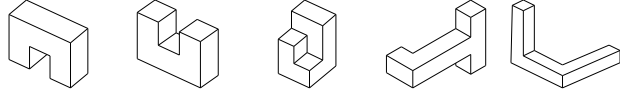


Figure 14: Figure 15: Figure 16: Figure 17: Figure 18:
N-Block U-Block Normalon J-Block Z-Block



Figure 29: Octahedron Figure 30:
Non-Trihedral

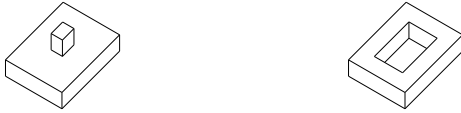


Figure 19: Boss Figure 20: Hole

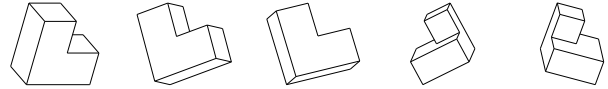


Figure 31: Figure 32: Figure 33: Figure 34: Figure 35:
Huffman-Clowes Objects [4, 9]

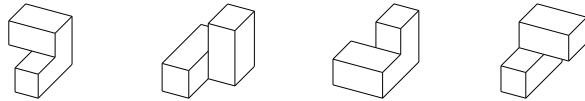


Figure 21: Figure 22: Figure 23: Figure 24:
Extended Trihedral Objects

drawings. However, the logic above can be extended for the JLPs in Figures 10–13 and other views of the same objects. Further JLPs can be found in drawings of objects built from a small number of cubes, e.g. Figures 14–18, and of the simplest trihedral objects with hole loops, Figures 19 and 20. Handling of JLPs including extended trihedral junction labels is straightforward, from Figures 21–24.

It can be seen in Fig. 31 that the *Wbca-Lac* junction label pair has different implications depending on whether the line being followed is convex and occluding. Failure to make this distinction was the most serious deficiency in the original presentation of JLPs [24].

It is not clear whether or not lines terminating in occluding T-junctions should be included in this analysis. It is certainly true that in normalons all lines leaving *Lba* junctions approach the viewer, even those terminating at T-junctions; this could be used as an argument in favour of including the junction label pairs *Lba-Tbaa* and *Lba-Tbab*. Against this, it can be argued that since nothing is known about what lies at the occluded end of an edge when the line terminates at a T-junction, no use should be made of “knowledge” about this occluded vertex. The argument generalises to other frequently-occurring combinations involving T-junctions. This issue is discussed in Section 6.

Not all drawings are in perfect isometric projection, or any mathematically-correct projection. It is not obvious whether JLP is more or less sensitive to different projections or drawing inaccuracies than (e.g.) corner orthogonality, particularly when used in combination with other compliance functions; a comparison is given in Section 6.

The JLP approach can be extended to non-normalons. Although some extra JLPs can appear in drawings of non-normalons compared to drawings of normalons, this can often be handled by ignoring the unknown JLP (one JLP per vertex is sufficient). A more robust solution is to generate a low-weight equation making the depth coordinates equal for the two vertices of an unrecognised JLP.

Extension to K-type tetrahedral vertices is straightforward, but it is at this point that the law of diminishing returns sets in: only those from Figures 25–28 have been included in our test system, RIBALD [25] (the earlier implementation [24] was even more restrictive, not allowing non-trihedral vertices). Similar logic can also be used for the occluding M-L pairs in Figures 29 and 30. Since these values are of less universal validity, RIBALD includes an arbitrary weighting factor W (range 0..1) in the equations in order to give priority to the more trustworthy values. The equations thus become (L_i is the length of edge i ; K is the depth/length ratio for the JLP)

$$W \times (z_{i1} - z_{i2}) = W \times L_i \times K.$$

The complete set of junction label pairs as implemented in RIBALD and used to obtain the results below are tabulated in Tables 1–3. As well as entries for JLPs deduced from geometric reasoning, some were derived experimentally, either from pairs which invariably produced the same depth ordering (marked *I* in the tables), or from pairs whose implications were unclear (e.g. it seems in Fig. 36 that the *Yccc* junction is in front of the *Mbcca* junction, but that a change of viewpoint could alter this), but which occurred sufficiently often to make doing something explicit worthwhile (marked *V* in the tables). For the latter, the depth ratio is $K = C/\sqrt{2}$ and the confidence $W = C/2$, where $C = (c_A - c_B)/(c_A + c_B + c_E)$ and c_A is the frequency with which vertex *A* is clearly closer than vertex *B*, c_B is the frequency with which vertex *B* is clearly closer than vertex *A*, and c_E is the frequency of them being about the same.

Line Type	Nearer J.	Further J.	K	W	Fig.
Convex	Yccc	Wbca	0.7071	1.0000	10
Convex	Yccc	Wdcd	0.7071	1.0000	11
Convex	Lac	Wbca	0.7071	1.0000	31
Convex	Lcb	Wbca	0.7071	1.0000	32
Convex	Yccc	Wdcd	0.7071	1.0000	12
Convex*	Wbca	Tbaa	0.7071	1.0000	14
Convex*	Wbca	Tbab	0.7071	1.0000	14
Convex	Lac	Wdcd	0.7071	1.0000	15
Convex	Lcb	Wdcd	0.7071	1.0000	15
Convex	Yccc	Tbdc	0.7071	0.75	22
Convex	Yccc	Tdac	0.7071	0.75	24
Convex	Yccc	Kcccd	0.7071	0.50	25
Convex	Kcccd	Wbca	0.7071	0.50	25
Convex	Yccc	Kccdc	0.7071	0.50	26
Convex	Kccdc	Wbca	0.7071	0.50	26
Convex	Yccc	Kccdc	0.7071	0.50	27
Convex	Yccc	Kdcdc	0.7071	0.50	28
Convex	Mbcca	Wbca	0.7071	0.50	I
Convex	Yccc	Kabcd	0.7071	0.50	I
Convex	Yccc	Kabdc	0.7071	0.50	I
Convex	Yccc	Mccdc	0.7071	0.50	I
Convex	Yccc	Mcdcc	0.7071	0.50	I
Convex	Yccc	Mbdca	0.7071	0.50	I
Convex	Yccc	Mbdca	0.7071	0.50	I
Convex	Xcccd	Wbca	0.4030	0.28	V
Convex	Xcccd	Mbcca	0.3928	0.28	V
Convex	Yccc	Mbcca	0.1414	0.10	V
Convex	Wdcd	Wbca	0.0345	0.02	V
Convex	(any)	(same)	0.0	0.4000	n/a
Convex	(any)	(other)	0.0	0.0100	n/a

Table 1: Constants and Weights for Depth Estimation

Line Type	Nearer J.	Further J.	K	W	Fig.
Concave	Wdcd	Yabd	0.7071	1.0000	11
Concave	Wdcd	Yabd	0.7071	1.0000	12
Concave	Wdcd	Lbd	0.7071	1.0000	34
Concave	Wdcd	Lda	0.7071	1.0000	35
Concave	Wdcd	Yadd	0.7071	1.0000	13
Concave	Wdcd	Yadd	0.7071	1.0000	16
Concave	Wdcd	Lbd	0.7071	1.0000	19
Concave	Wdcd	Lda	0.7071	1.0000	19
Concave	Wdcd	Tbda	0.7071	0.75	21
Concave	Tbdc	Yabd	0.7071	0.75	22
Concave	Wdcd	Tdab	0.7071	0.75	23
Concave	Tdac	Yabd	0.7071	0.75	24
Concave	Kccdc	Yabd	0.7071	0.50	27
Concave	Kdcdc	Yabd	0.7071	0.50	28
Concave	Kcccd	Yabd	0.7071	0.50	I
Concave	Kccdc	Yabd	0.7071	0.50	I
Concave	Wdcd	Kabcd	0.7071	0.50	I
Concave	Wdcd	Kabcd	0.7071	0.50	I
Concave	Mbdca	Mdcd	0.7071	0.50	I
Concave	Mbdca	Mdcd	0.7071	0.50	I
Concave	Mbdca	Mdcd	0.7071	0.50	I
Concave	Mbdca	Mdcd	0.7071	0.50	I
Concave*	Tbad	Wdcd	0.0304	0.02	V
Concave	(any)	(same)	0.0	0.4000	n/a
Concave	(any)	(other)	0.0	0.0100	n/a

Table 2: Constants and Weights for Depth Estimation

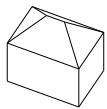


Figure 36: House

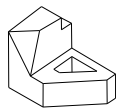


Figure 37: Solid [30]

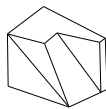


Figure 38: Solid [30]

(Frequencies were measured from 479 test drawings which can be labelled correctly.) JLPs using lines which terminate in occluding T-junctions are marked with asterisks.

This compliance function fits easily into either linear system approach and could also be useful either as a part of an optimisation approach or as a preprocessing stage to avoid the “local minimum trap”. Entries in the JLP tables can be used directly to predict (with very few exceptions) the depth ordering of neighbouring vertices, so this combina-

Line Type	Nearer J.	Further J.	K	W	Fig.
Occluding	Wbca	Lba	0.7071	1.0000	10
Occluding	Wbca	Yabd	0.7071	1.0000	11
Occluding	Wbca	Lac	0.7071	1.0000	31
Occluding	Wbca	Lcb	0.7071	1.0000	32
Occluding*	Tbab	Lba	0.7071	1.0000	31
Occluding*	Tbaa	Lba	0.7071	1.0000	32
Occluding	Lab	Lba	0.7071	1.0000	33
Occluding	Wbca	Lbd	0.7071	1.0000	34
Occluding	Wbca	Lda	0.7071	1.0000	35
Occluding	Lab	Lac	0.7071	1.0000	14
Occluding	Lab	Lcb	0.7071	1.0000	14
Occluding	Lab	Yabd	0.7071	1.0000	17
Occluding*	Tbab	Yabd	0.7071	1.0000	20
Occluding*	Tbaa	Yabd	0.7071	1.0000	20
Occluding	Wbca	Tbda	0.7071	0.75	21
Occluding	Tbda	Lba	0.7071	0.75	21
Occluding	Wbca	Tbdc	0.7071	0.75	22
Occluding	Wbca	Tdab	0.7071	0.75	23
Occluding	Tdab	Lba	0.7071	0.75	23
Occluding	Wbca	Tdac	0.7071	0.75	24
Occluding	Mbdca	Lba	1.4142	0.50	29
Occluding	Mbdca	Lba	0.7071	0.50	30
Occluding	Mbdca	Lba	0.7071	0.50	30
Occluding	Kabcd	Lba	0.7071	0.50	I
Occluding	Kabcd	Lba	0.7071	0.50	I
Occluding	Wbca	Kabcd	0.7071	0.50	I
Occluding	Wbca	Kabcd	0.7071	0.50	I
Occluding	Mbdca	Wbca	0.1872	0.13	V
Occluding	Lba	Yabd	0.0848	0.06	V
Occluding	(any)	(same)	0.0	0.4000	n/a
Occluding	(any)	(other)	0.0	0.0100	n/a

Table 3: Constants and Weights for Depth Estimation

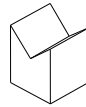


Figure 39: Extrusion

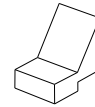


Figure 40: Extrusion [30]

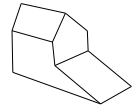


Figure 41: Object [30]

tion could therefore be used in a preprocessing stage for the corner orthogonality method described above.

JLP requires junction and line labels.

4. GRIMSTEAD’S APPROACH

Using equations of the form $P_f x_v + Q_f y_v + z_v + C_f = 0$, Grimstead generated the frontal geometry of the object using a system of linear equations with the unknowns P_f , Q_f , C_f and z_v .

Face planarity equations were generated for each vertex-face pair. Line parallelism equations were generated for each pair of 2D parallel lines identified. Equations for P and Q were generated for each skewed symmetry artifact detected.

The system was then solved to give equations for each face using a weighted least-squares algorithm. The solution process was iterative, with the equation with the largest residual error being dropped at each subsequent iteration. Weightings of equations with large residuals were reduced on the next iteration; the original weightings of each equation were arbitrary. Eventually, a self-consistent set of equations remained. The performance was assessed using right-angle fit and minimum standard deviation of angles in order to determine whether the process was converging towards a 3D solid or towards the redundant (flat) solution. In the former case, the output of the linear system gave P , Q and C values for each face; vertex coordinates were obtained by intersecting faces.

Grimstead’s approach suffered from several disadvantages: It is an iterative process, $O(n^4)$ or worse. This could be

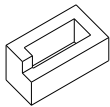


Figure 42:
Sugihara's Box [23]

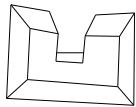


Figure 43: No Visible Vertices on Back Line

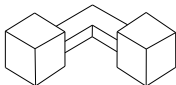


Figure 44: Cannot Reach Back From Front

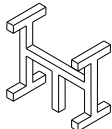


Figure 45: Too Many Parallel Line Pairs

overcome: the output of the first iteration could be used for preliminary depth coordinates.

It changes vertex x - and y -coordinates, an idea which has already been rejected. This too is not insurmountable: the linear system gives vertex z -coordinates as part of its output, and these could be used directly instead of recalculating vertex coordinates from face equations.

The chosen weightings are arbitrary: those which were empirically found to work best for the test drawings used.

The variables being optimised are of different kinds, thus creating doubt as to whether the fit really is “best”.

The skewed symmetry equations for P and Q are decoupled: a single geometric constraint is represented by multiple constraint equations (iterative least-squares fitting might drop one and retain the other when rejecting inconsistent equations; iterative weighting adjustment might make one more important than the other).

Also, a hidden weighting is also given to skewed symmetry estimates of ‘front-on’, rather than ‘side-on’, faces, through the representation of the face normal as $[P, Q, 1]$ rather than $[P, Q, R]$. This is justifiable: it is easier for the user to draw “front-on” faces, so they will be more accurate.

5. DEPTH FROM LABELLING

RIBALD uses a linear system where the only unknowns are the vertex z -coordinates. The number of unknowns is the number of junctions in the drawing. In the simplest form of this approach, the number of equations is one more than the number of lines in the drawing, these equations being generated using the JLP compliance function. The additional equation fixes an arbitrary vertex at an arbitrary depth in order to locate the object in space. Where a JLP is not explicitly tabulated, the depth difference is assumed to be zero and the weighting very low. This guarantees that each unknown will be represented in the system. A linear system solver is used to find the least-squares fit to the overconstrained system of equations.

For a simplistic method, this approach works well in practice for most junction types (it is even possible to obtain a correct depth ordering for Sugihara's Box, Fig. 42)! The method fails entirely for drawings which are not graph-complete, e.g. Figures 43 and 44: it is not possible to find a path between every pair of edges without making use of occluding T-junctions, and the resulting linear system cannot be solved.

Various refinements have been investigated. Equations from line parallelism (linear in z -coordinates) are easily incorporated into the linear system and in many cases improve the frontal geometry significantly, so are recommended. It is found in practice that the computational overhead is acceptable for most drawings of engineering objects but becomes unacceptable if *all* possible parallel line pairs in drawings e.g. Fig. 45 are included in the linear system. A threshold is applied; if the number of lines n in a parallel bundle is below the threshold, all $n(n-1)$ possible line pairs generate equations; if n is above the threshold, $(n-1)$ equations are generated to make the longest line in the bundle parallel to all of the others. Setting the threshold to 20 is a satisfactory compromise.

RIBALD sets weightings for parallel line pair equations proportional to the product of the lengths of the two lines and to the figure of merit for parallelism. The relative weightings of parallel line pairs and JLPs are such that if the two longest lines are exactly parallel then the weighting for the equation making them parallel is the same as the weighting for the equation giving the direction of an $Lba-Wbca$ line.

If line parallelism is not included, some other mechanism must be used to ensure that edges resulting from interrupted lines at K-type vertices and extended trihedral T-junctions are collinear in 3D.

Face planarity is more difficult. Enforcing it as a postprocessing stage can be rejected: it sometimes improves well-drawn sketches, but can also make poorly-drawn sketches significantly worse, and conflicts with the preference for leaving the x and y coordinates for each vertex unchanged from the original drawing.

For drawings including hole loops, e.g. Fig. 46, some mechanism is needed to make cofacial loops coplanar. This is best achieved by incorporating the four-vertex version of the face planarity function into the linear system.

Use of four-vertex coplanarity for the general case of four or more vertices lying on a face has been examined, with results shown in Section 6.

Various solutions are possible to the problems posed by Figures 43 and 44, but none are ideal. An equation placing the point at which the line vanishes from view at a fixed arbitrary depth behind the corresponding point on the line which occludes it is linear in z -coordinates, and could be incorporated. These are sometimes needed for drawings where the subgraph count is greater than 1, but if included unnecessarily the quality of the output deteriorates. As implemented in RIBALD, such equations are, by default, not included, but if the linear system cannot be solved and the subgraph count is greater than 1, such equations are added to the system, and the linear system solver invoked a second time.

It can be seen in Figures 46 and 49 that lines terminating at occluding T-junctions are often parallel to other lines in the drawing. If this is so, use of entries in the JLP tables referring to occluding T-junctions is not required.

Since corner orthogonality can be used in linear systems, and is preferable for isometric projections of normalons, RIBALD includes an auto-selection process which uses corner orthogonality equations in place of the default JLP for objects with exactly three bundles of parallel lines and where all W-junctions and Y-junctions meet the Perkins criteria. Alternatively, either compliance function can be invoked

α	β	JLP			CO		
		worst	mean	best	worst	mean	best
30	30	89.999	90.000	90.000	89.999	90.000	90.000
25	40	84.369	85.951	86.743	86.340	87.600	88.801
20	50	79.014	81.486	82.722	82.873	84.803	86.357
15	60	73.917	76.598	77.938	79.474	81.688	83.684
10	70	69.060	71.322	72.453	76.063	78.308	81.409
5	80	64.427	65.745	66.404	72.587	74.721	79.053

Table 4: Sensitivity of Inflation to Projection

α	β	JLP			CO		
		worst	mean	best	worst	mean	best
30	30	89.999	90.000	90.000	89.999	90.000	90.000
25	35	87.040	88.026	89.855	88.014	88.974	89.886
20	40	83.927	85.951	89.420	85.746	87.866	89.422
15	45	80.658	83.772	88.699	83.146	86.719	88.886
10	50	77.228	81.486	87.698	80.192	85.417	89.270
5	55	73.640	79.093	86.427	77.193	82.926	85.214

Table 5: Sensitivity of Inflation to Projection

specifically. Section 6 analyses the output produced.

6. RESULTS AND CONCLUSIONS

The derivation of the JLP approach assumed a drawing in isometric projection. Its performance in non-isometric projections is compared here with corner orthogonality, which does not make this assumption. Furthermore, the derivation of both compliance functions assumed a drawing in a correct orthogonal projection. The performance of both in incorrect projections is also investigated.

Choice between JLP and corner orthogonality, and choice concerning which secondary compliance functions should be used in combination with them, is investigated with reference to six test drawings selected partly to be a representative sample of test drawings and partly to investigate specific issues.

6.1 JLP vs Corner Orthogonality

A preliminary investigation [24] was sufficient to reject skewed facial symmetry. With parallelism of lines incorporated, RIBALD produced dihedral angles for the three cubes shown in Figures 7–9 of 90.0° , 90.0° , 90.0° for Fig. 7, 87.3° , 80.8° , 80.4° for Fig. 8, and 84.3° , 84.3° , 71.3° for Fig. 9. By way of comparison, skewed symmetry produces angles of 90° for Figures 7 and 8, both of which are correct projections, but fails entirely for Fig. 9. Distorting the projection by moving the right-hand lines 10° closer to the horizontal gives inter-facial angles of 89.4° , 84.5° , 83.9° for Fig. 7 (skewed symmetry gives 99.1° , 89.4° , 81.7°) and 87.9° , 82.0° , 74.5° for Fig. 8 (skewed symmetry gives 101.5° , 71.4° , 69.5°). Although better in some circumstances, skewed facial symmetry is inappropriate for initial inflation because there are valid drawings for which it fails entirely, and because it is roughly twice as sensitive as JLP to typical free-hand drawing errors.

Comparison of depth ratios derived from JLP with those derived from corner orthogonality gives less consistent results. The tests in Tables 4–6 were run on variants of Fig. 46 in which all lines were drawn parallel with one of three axes, one of the axes being vertical, and the other two axes at angles of α and β with the horizontal. Distortions in Table 4 approximate correct non-isometric projections; distortions in Tables 5 and 6 represent two typical ways of deviating from isometric projection. The RIBALD options used were

α	β	JLP			CO		
		worst	mean	best	worst	mean	best
30	30	89.999	90.000	90.000	89.999	90.000	90.000
30	35	87.040	88.026	89.855	87.610	88.608	89.918
30	40	83.927	85.951	89.420	85.154	87.196	89.858
30	45	80.658	83.772	88.699	82.786	85.554	89.311
30	50	77.228	81.486	87.698	80.424	83.654	87.648
30	55	73.640	79.093	86.426	76.576	81.563	86.025

Table 6: Sensitivity of Inflation to Projection

\perp	methods				dihedral angles			Δ planarity	
	ll	4vp	T \perp		worst	mean	best	mean	worst
JLP	N	N	N		80.657	84.575	88.999	5.621	36.3523
CO	N	N	N		83.390	86.029	89.619	9.968	27.4684
JLP	Y	N	N		80.657	83.772	88.699	0.000	0.0021
CO	Y	N	N		82.728	85.543	89.225	2.832	5.6158
JLP	N	Y	N		80.657	84.575	88.999	5.621	36.3523
CO	N	Y	N		83.036	85.887	88.881	6.602	27.2313
JLP	Y	Y	N		80.657	83.772	88.699	0.000	0.0021
CO	Y	Y	N		82.924	85.433	88.606	1.660	5.3446
JLP	N	N	Y		80.657	83.772	88.699	0.000	0.0000
CO	N	N	Y		81.867	85.511	89.619	5.691	13.9337
JLP	Y	N	Y		80.657	83.772	88.699	0.000	0.0000
CO	Y	N	Y		82.591	85.501	89.186	2.745	5.6111
JLP	N	Y	Y		80.657	83.772	88.699	0.000	0.0000
CO	N	Y	Y		82.427	85.371	88.881	2.473	7.7982
JLP	Y	Y	Y		80.657	83.772	88.699	0.000	0.0000
CO	Y	Y	Y		82.788	85.392	88.567	1.631	5.3402

Table 7: Results for the O-Block

to generate depth equations from all vertex pairs, including the two Y–T pairs, and from parallel lines bundled together. The JLP columns show the worst, mean and best perpendicular dihedral angles obtained using a fixed depth ratio of $1/\sqrt{2}$, and the CO columns show the equivalent results obtained using variable depth ratios calculated using corner orthogonality. These results show no consistent reason for preferring either method, except in Table 4, where corner orthogonality is clearly preferable.

6.2 Illustrative Results

Further testing is needed to establish whether JLP or corner orthogonality is to be preferred more generally, and which (if any) secondary compliance functions should be used in combination with them. Sample results are illustrated here with reference to six test drawings (see Figures 46–50). In the Tables, column \perp indicates the method used for creating perpendicularity, column ll shows whether or not line parallelism was used, column 4vp shows whether four-vertex planarity was used, and column T \perp indicates whether lines terminating at T-junctions produced perpendicularity equations. The remaining columns vary depending on what is examined.

Table 7 shows the worst, mean and best perpendicular dihedral angles after inflation of Fig. 46, and the mean and worst deviations from planarity. Dihedral angles for each edge were estimated by calculating a best-fit face equation for the faces meeting the edge. Deviations from planarity were estimated by calculating a best-fit face equation for each face and the distance of the vertex coordinates from this plane (the absolute values, in pixels, are arbitrary but the relative values significant). Several points can be noted. Firstly, because of the presence of a hole loop, some face planarity equations were present in the system regardless of the options selected in order to make the inner and outer loops of the O-face coplanar. Secondly, none of the methods give dihedral angles of 90° : although the drawing looks reasonable, it is not a perfect projection. Thirdly, although best

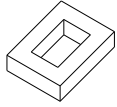


Figure 46:
O Block

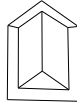


Figure 47:
Bracket [21]

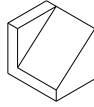


Figure 48:
Wedge [19]

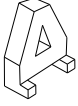


Figure 49:
A Block [30]

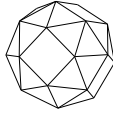


Figure 50:
Snub Cube

methods			dihedral angles			Δ planarity	
\perp	ll	4vp	worst 90	best 90	roof	mean	worst
JLP	N	N	67.623	87.121	76.570	22.097	53.5437
CO	N	N	73.418	85.196	72.811	22.715	71.8669
JLP	Y	N	77.060	87.256	71.505	15.635	43.9723
CO	Y	N	78.283	86.798	66.865	14.253	51.5327
JLP	N	Y	70.907	87.903	60.294	6.828	17.5546
CO	N	Y	79.500	88.506	55.368	3.129	10.9773
JLP	Y	Y	78.094	88.539	59.345	3.695	5.7795
CO	Y	Y	82.980	87.886	54.814	2.492	10.5396

Table 9: Dihedral Angles for the Wedge

dihedral angle results are obtained using just corner orthogonality, the differences are slight: all variants of the method give reasonable results. Fourthly, line parallelism equations are as effective as face planarity equations in enforcing face planarity: one or other should be included, but using both is superfluous.

Fig. 47 given in various textbooks, e.g. Shirai’s [21], illustrates the difference between freehand drawings and mathematically correct drawings. It is not mathematically-correct, since the two edges which divide the top face from the front face must be collinear, and the corresponding lines are not. Ideally, after inflation, all dihedral angles should be either 90° or 45° (making the convex hull of the top face an equilateral triangle might also be a reasonable interpretation). The final four columns of Table 8 show the worst and best right-angles, the worst and best 45° angles, and the mean and worst deviations from planarity. It is clear from the right-angle columns that JLP is preferable to corner orthogonality, both in terms of dihedral angles and face planarity, confirming the initial impression that corner orthogonality does not produce good results for non-normalons. Otherwise, there is little to choose between the variants. Analysis of dihedral angles suggests that it is preferable to use the simplest version of the method, since this is the one in which the two dihedral angles which should be 45° are most nearly equal. Face planarity equations have some effect in achieving their objective.

The wedge in Fig. 48 shows what happens for objects with non-axis-aligned faces and no helpful mirror symmetry. If the idea of line isometry is accepted, the “sloping roof” should make an angle with the L-shaped front face of $\arctan \frac{4}{3}$, 53.13° . The rightmost columns in Table 9 show the worst and best perpendicular dihedral angles and the angle between the sloping roof and the L-shaped front face, and the mean and worst deviations from planarity. Surprisingly, corner orthogonality is consistently preferable to

methods			dihedral angles			Δ planarity	
\perp	ll	4vp	small	mean	large	mean	worst
JLP	N	N	0.028	17.737	71.918	2.342	69.5815
JLP	Y	N	2.257	17.636	67.669	1.501	28.5971
JLP	N	Y	1.203	21.189	54.495	0.359	4.9473
JLP	Y	Y	3.978	19.343	62.795	0.424	5.7120

Table 11: Dihedral Angles for the Snub Cube

JLP in achieving the correct dihedral angle, although the difference is not always large. Four-vertex planarity is almost essential for a good interpretation. The vertex which is most seriously misplaced if face planarity equations are not included is the one where the concave line meets the L-shaped front face. This gives few clues concerning which feature present in Fig. 48 necessitates the use of face planarity equations.

In Fig. 49, the dihedral angle between the square top and the sloping side should be 60° if line isometry is accepted; the dihedral angle between the sloping side and the vertical side should be 30° . If line parallelism is accepted, the internal dihedral angle in the triangular through hole should be 60° . The rightmost columns of Table 10 list, respectively, these three angles, the worst and best perpendicular dihedral angles, and the mean and worst deviations from planarity. This drawing was included because it illustrates a common type of problem junction (an occluding T-junction completing a triangular partial face) for which only line parallelism generates a reasonable equation, and as expected, line parallelism is almost essential for good dihedral angles. Four-vertex planarity does more harm than good to dihedral angle values, but its incorporation can still be justified for its significant effect in making faces planar. Note that here the improvement of JLP over corner orthogonality on deviations from planarity is almost as large as that achieved using the face planarity function.

Also note that in neither this drawing nor any of the preceding ones does presence or absence of occluding T-junction entries in the JLP tables make any significant difference.

Fig. 50 is a snub cube, an Archimedean solid in which all dihedral angles are either 26.8° or 37.0° . The rightmost columns in Table 11 show the smallest, mean and largest dihedral angles and the mean and worst deviations from planarity. As all vertices of this object are pentahedral, corner orthogonality cannot be used.

None of the variants tested produce good frontal geometry for this drawing, and did not necessarily achieve correct depth ordering of adjacent vertices. In view of the global regularity of the object, a global approach such as Marill’s MSDA would be more appropriate here than any approach based on accumulation of local data.

6.3 Overall Results

In this Section, all 479 drawings which can be labelled correctly using any of the labelling algorithms described in [27] were used as test data. Table 12 lists the number of drawings for which all neighbouring vertices were ordered correctly in depth by the variants listed above. Since different methods may be preferred for normalons and non-normalons, results are also presented separately for those drawings where bundling identified three bundles and those where it identified four or more bundles.

The results confirm that use of parallel line information in inflation is almost essential.

methods			dihedral angles				Δ planarity	
\perp	ll	4vp	worst 90	best 90	worst 45	best 45	mean	worst
JLP	N	N	84.032	87.945	49.316	46.025	17.440	40.6833
CO	N	N	73.132	88.562	58.292	49.494	23.341	156.3602
JLP	Y	N	84.643	88.204	52.905	42.458	12.802	43.5912
CO	Y	N	74.347	89.809	55.720	50.002	12.633	55.8313
JLP	N	Y	84.166	89.993	52.134	42.793	6.105	34.0688
CO	N	Y	77.057	89.297	59.107	43.848	9.004	39.2854
JLP	Y	Y	84.338	88.819	53.263	41.456	6.642	35.9234
CO	Y	Y	77.312	89.046	59.443	43.247	8.575	39.9671

Table 8: Dihedral Angles for the Angle Bracket

methods				dihedral angles					Δ planarity	
\perp	ll	4vp	$T\perp$	$T(60^\circ)$	$F(30^\circ)$	$I(60^\circ)$	Worst \perp	Best \perp	mean	worst
JLP	N	N	N	59.248	36.660	86.411	60.674	89.505	7.234	47.373
CO	N	N	N	61.523	37.440	86.411	59.849	89.795	9.135	81.887
JLP	Y	N	N	60.310	35.722	77.466	84.125	89.644	6.066	44.208
CO	Y	N	N	62.089	37.408	78.139	78.761	89.432	7.475	82.506
JLP	N	Y	N	65.221	27.754	86.019	56.057	89.402	3.613	18.771
CO	N	Y	N	65.307	28.083	82.644	51.349	89.534	5.390	51.428
JLP	Y	Y	N	65.723	27.197	78.024	79.780	89.854	2.904	15.759
CO	Y	Y	N	65.844	27.574	80.403	70.211	89.965	5.485	52.259
JLP	N	N	Y	59.248	36.660	84.815	59.184	89.505	6.327	47.373
CO	N	N	Y	61.523	37.440	84.815	58.326	89.795	9.135	81.887
JLP	Y	N	Y	60.310	35.722	77.470	84.125	89.644	6.066	44.208
CO	Y	N	Y	62.089	37.409	78.144	78.760	89.433	7.475	82.506
JLP	N	Y	Y	65.221	27.754	84.431	54.532	89.402	2.709	13.886
CO	N	Y	Y	65.307	28.083	80.961	49.822	89.534	5.390	51.428
JLP	Y	Y	Y	65.723	27.197	78.027	79.781	89.853	2.902	15.759
CO	Y	Y	Y	65.844	27.574	80.409	70.210	89.966	5.485	52.259

Table 10: Dihedral Angles for the A-Block

methods			overall		3-bundle		4+bundle	
\perp	ll	4vp	\checkmark	\times	\checkmark	\times	\checkmark	\times
JLP	Y	Y	293	186	128	7	165	179
JLP	Y	N	305	174	131	4	174	170
CO	Y	Y	255	224	103	32	152	192
CO	Y	N	254	225	100	35	154	190
JLP	N	Y	144	335	55	80	89	255
JLP	N	N	138	341	62	73	76	268
CO	N	Y	155	324	51	84	104	240
CO	N	N	144	335	54	81	90	254



Figure 52: House



Figure 53: Setting Piece [19]

Table 12: Correct and Incorrect Neighbour Ordering



Figure 51: Extrusions [3]



Figure 54: Archimedean Solid



Figure 55: Normalon

The apparent advantage of JLP over corner orthogonality comes from the inability of corner orthogonality to handle correctly drawings using the projection in Fig. 51.

The tabulated results show that it is slightly preferable to omit four-vertex planarity equations. This could be an artifact of the assessment criterion chosen, that of correct depth ordering of adjacent vertices. Since JLP is specifically designed to produce correct depth ordering of adjacent vertices, it is plausible that a combination with other compliance functions will make it less effective at meeting this specific criterion even if it improves results in a more general but less easily-quantified way. In drawings with more than one subgraph, e.g. Fig. 6, four-vertex planarity equations are the best way of ensuring that the depth coordinates of the two subgraphs are related.

In many of the cases for which the best variant (JLP, using parallel lines but not 4-vertex planarity) produced “incorrect” results, depth ordering produced by the algorithm is tolerable but is not the one we expect. Figures 52 and 53 show two such cases.

The Archimedean solids defeat RIBALD’s inflation approach: even the best variant produced some incorrect depth ordering for all but one of them.

6.4 Timings

Only limiting cases were investigated. With Fig. 54, the drawing with most lines, depth estimation using JLPs only takes approximately 0.10 seconds (on a Sun Ultra 10 running Solaris). Adding parallel line pair equations increases the time to 0.26 seconds. Adding face planarity equations increases the time to 0.18 seconds. Adding both increases the time to 0.34 seconds.

Fig. 55 has the largest number of parallel lines. Depth estimation using the Perkins equations only takes approximately 0.05 seconds. Adding parallel line pair equations increases the time to 0.19 seconds. Adding face planarity equations increases the time to 0.07 seconds. Adding both increases the time to 0.22 seconds.

It can be concluded that the ideas in this paper lead to an implementation which runs acceptably quickly.

7. CONCLUSIONS

Experimental results justify the use of linear systems in which the only unknowns are the depth coordinates as a fast and relatively robust method of producing initial estimates of depth coordinates of junctions in a line drawing. For typical engineering objects, correct depth ordering and reasonable geometry are obtained in acceptable time. Where a later beautification stage will follow initial inflation, the method outlined here is usually adequate to ensuring that any optimisation process will start close enough to the global minimum to avoid being trapped in a local minimum.

Selection between JLP and corner orthogonality is less clear. A sophisticated system should choose between them using criteria based on the particular drawing being processed. It was expected that corner orthogonality would produce better results for normalons and JLP for non-normalons; this is often true, but there are exceptions e.g. Fig. 48. If a system must use one or the other, choice of JLP can be justified in that it is more robust and versatile: corner orthogonality can only be used for trihedral vertices which meet the Perkins criteria.

Of the complementary options investigated, use of line parallelism is strongly recommended: there are drawings for which good output can only be achieved if line parallelism is used. Four-vertex planarity equations are required for drawings with hole loops; their use in other circumstances can result in minor benefits but is not essential. Line collinearity should be used for drawings such as Fig. 6 but its use in drawings such as Fig. 5 produces no benefit. The presence or absence of T-junction entries in the JLP table is irrelevant.

In a few cases, e.g. the Platonic and Archimedean solids, the linear system approach fails to achieve its objectives. These do not correspond to common engineering objects, and can be ignored. In applications where they are needed, they should be treated as special cases (they are easily identified: all internal edges are convex) and processed using Marill's MSDA [14].

8. ACKNOWLEDGEMENTS

The authors are grateful to Unigraphics Solutions Inc. for providing Parasolid to the authors for use in this research.

9. REFERENCES

- [1] H.G.Barrow, J.M.Tenenbaum, *Interpreting Line Drawings as Three-Dimensional Surfaces*, Artificial Intelligence **17**, 75–116, 1981.
- [2] R.P.Brent, *Algorithms for Minimization without Derivatives*, Prentice-Hall, 1973.
- [3] Capital Aluminium Extrusions Ltd, *Catalogue of Standard Profiles*, 2000.
- [4] M.B.Clowes, *On Seeing Things*, Artificial Intelligence **2** 79–116, 1970.
- [5] P.Company, J.M.Gomis, M.Contero, *Geometrical Reconstruction from Single Line Drawings using Optimization-based Approaches*, in ed. V.Skala, Proc. WSCG99 7th Int. Conf. in Central Europe on Computer Graphics, Visualization and Interactive Digital Media'99, University of West Bohemia, Plzen, **II** 361–368, 1999.
- [6] P.Company, J.M.Gomis, M.Contero, *An Optimization-Based Algorithm to Reconstruct 3D Models from Single Line Drawings*, II Seminario Italo-Spagnolo. **2** 952–958, 1998.
- [7] J.Conesa Pastor, P.Company Calleja, J.M.Gomis Marti, *Initial Modeling Strategies for Geometrical Reconstruction—Optimization-Based Approaches*, Proc. 11th Int. Conf. on Design Tools and Methods in Industrial Engineering, 161–171, 1999.
- [8] I.J.Grimstead, *Interactive Sketch Input of Boundary Representation Solid Models*, PhD Thesis, Cardiff University, October 1997.
- [9] D.A.Huffman, *Impossible Objects as Nonsense Sentences*, Machine Intelligence **6** 295–323, 1971.
- [10] T.Kanade, *Recovery of the Three-Dimensional Shape of an Object from a Single View*, Artificial Intelligence **17**, 409–460, 1981.
- [11] Y.G.Leclerc, M.A.Fischler, *An Optimization-based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames*, Int. J. Computer Vision, **9**(2) 113–136, 1992.
- [12] H.Lipson, M.Shpitalni, *Optimization-based Reconstruction of a 3D Object from a Single Freehand Line Drawing*, Computer-Aided Design **28**, 651–663, 1996.
- [13] H.Lipson, M.Shpitalni, *An Interface for 3D Conceptual Design Based on Freehand Sketching*, in ed M.J.Pratt, R.D.Sriram, M.J.Wozny, Product Modeling for Computer Integrated Design and Manufacture, Chapman and Hall, 1997.
- [14] T.Marill, *Emulating the Human Interpretation of Line-Drawings as Three-Dimensional Objects*, Int. J. Computer Vision **6**(2) 147–161, 1991.
- [15] H Mayer, *Automatic Object Extraction from Aerial Imagery—A Survey Focusing on Buildings*, Computer Vision and Image Understanding **74**(2), 138–149, 1999.
- [16] S.Meeran, J.M.Taib, *A Generic Approach to Recognising Isolated, Nested and Interacting Features from 2D Drawings*, Computer-Aided Design **13**(14), 891–910, 1999.
- [17] I.V.Nagendra, U.G.Gujar, *3D Objects From 2D Orthographic Views*, Computers & Graphics **12**(1) 111–114, 1988.
- [18] D.N.Perkins, *Cubic Corners*, Quarterly Progress Report 89, 207–214, MIT Research Laboratory of Electronics, 1968.
- [19] F.Pickup, M.A.Parker, *Engineering Drawing with Worked Examples*, **1**, 3rd Edition, Hutchison and Co, 1979.
- [20] L.Ros, *A Kinematic-Geometric Approach to Spatial Interpretation of Line Drawings*, PhD Thesis, Universitat Politècnica de Catalunya, 2000.
- [21] Y. Shirai, *Three-Dimensional Computer Vision*, Springer-Verlag, 1987.
- [22] M.Shpitalni, H.Lipson, *Identification of Faces in a 2D Line Drawing Projection of a Wireframe Object*, IEEE Trans. Pattern Analysis and Machine Intelligence, **18**(10) 1000–1012, 1996.
- [23] K.Sugihara, *Machine Interpretation of Line Drawings*, MIT Press, 1986.
- [24] P.A.C.Varley, R.R.Martin, *Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch—Sketch Categorisation and Frontal Geometry*, 1st Korea-UK Joint Workshop on Geometric Modeling and Computer Graphics, 2000.
- [25] P.A.C.Varley, R.R.Martin, *Interpretation of Single Sketch Input for Mesh and Solid Models*, Int. J. Shape Modelling **6**(2), 207–241, 2000.
- [26] P.A.C.Varley, R.R.Martin, *The Junction Catalogue for Labelling Line Drawings of Polyhedra with Tetrahedral Vertices*, Int. J. Shape Modelling **7**(1), 23–44, 2001.
- [27] P.A.C.Varley, R.R.Martin, *Alternatives for Labelling Line Drawings*, submitted to Int. J. Shape Modelling, 2000.
- [28] P.A.C.Varley, *Automatic Creation of Boundary-Representation Models from Single Line Drawings*, PhD Thesis, Cardiff University, 2002.
- [29] T.Vetter, T.Poggio, *Symmetric 3D objects are an easy case for 2D object recognition*, Human Symmetry Perception, ed C.W.Tyler, 349–359, 1996.
- [30] H.W.Yankee, *Engineering Graphics*, Prindle, Weber and Schmidt, 1985.