A System for Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch

P. A. C. Varley, R. R. Martin

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

Abstract

This paper describes a system which converts a two-dimensional sketch with hidden lines removed of a single polyhedral object into a boundary representation solid model.

The system improves on a previous system by Grimstead by analysing the sketch for symmetry elements and regularities. It uses the results of this analysis to produce a more plausible topological completion of the hidden parts of the object, and to enforce exact constraints on the geometry of the boundary-representation model.

1. Introduction

This paper describes a system which converts a two-dimensional sketch with hidden lines removed of a single polyhedral object with no cracks or shadows into a boundary representation solid model. It reports on the current capabilities of the system, which is still under development. Until we have established a set of methods which cover our input domain, it would be premature to attempt to abstract any general principles; in places, our approach is based on analysing and reconstructing various special cases.

Textbooks on engineering drawing (e.g. [6]) emphasise the importance of freehand sketching in the design process. Studies such as Jenkins [14] have shown that such a system would be of considerable interest to engineers and architects, whose current working practices involve freehand sketching on paper and subsequent manual input to a CAD package.

Although the theoretical impossibility of perfect conversion of 2D to 3D is both obvious and well-known, there exists an expanding set of objects for which conversion can be achieved in practice. Some assumptions concerning the sketch are required — removing some of the restrictions stated below will be the subject of future research. We assume that the sketch is of a single manifold polyhedral object. Topologically, we currently assume that all vertices in the object are trihedral and that no face contains more than one loop of edges. Geometrically, we assume that all faces are planar, and that the object is in a general position, with no faces lying perpendicular to the view direction. Lastly, we assume that the object has been sketched from the "most interesting viewpoint", i.e. that there is nothing at the rear of the object which could not reasonably be inferred from the visible part of the object.

In terminology, we use *region*, *line* and *junction* to denote *atoms* of a two-dimensional *sketch*, and *face*, *edge* and *vertex* to denote atoms of the three-dimensional *object* which has been sketched. Most, but not all, junctions are also vertices; visible vertices are junctions. All lines are edges or parts of edges; visible edges and visible parts of partially-obscured edges are lines. Regions are visible faces or the visible parts of partially-obscured faces. A junction where two lines meet is *biconnected*; a junction of three lines is *triconnected*.

We also distinguish *recognition*, where the computer chooses one of a finite set of candidate objects, from *interpretation*, where the computer creates a new object from an infinite set of constructible objects. We are interested only in the latter.

The problem is to convert the sketch into a boundary representation solid model of the most plausible 3D interpretation of the sketch, and to do so in a reasonable time (a second or less on a powerful personal computer is "reasonable").

Several variations on the type of input are possible: the sketch could show shaded faces or simply lines representing edges, and in the latter case the sketch could have hidden lines drawn, or only the visible lines drawn as in the case of

our system.

A method requiring a complete wireframe object has been developed by Lipson and Shpitalni [22, 23]. It is tolerant of freehand sketching errors, and allows flat and cylindrical faces. It assumes the object is in a general position. Many of the ideas contained in it are of interest. In particular, they provide a list of useful compliance functions: face planarity; line parallelism; line verticality; isometry; corner orthogonality; skewed facial orthogonality; skewed facial symmetry; line orthogonality; minimum standard deviation of angles; face perpendicularity; regularity of end faces of prisms; line collinearity; planarity of skewed chains. These compliance functions are weighted according to their degrees of accuracy in freehand sketches. The system produces depth values by performing a least-squares fit against these. The authors have examined several optimisation methods and chose cyclic application of Brent's method [3] to each vertex in turn. They stress the need for reasonable preliminary estimates in order to minimise optimisation time.

The disadvantage of using sketches with hidden lines visible, that of ambiguity (Necker reversal), is in principle insoluble, although frequently the symmetry of the drawn object will make the two potential interpretations the same. There are other disadvantages of using wire-frame sketches rather than sketches without hidden lines: it is more natural to draw only that which can be seen, and it is quicker and easier to draw a smaller rather than a larger number of lines.

Grimstead [8] provides a prototype system based on input of sketches made with hidden lines removed. This comprises five stages: incremental line labelling; two-dimensional drawing tidying; conversion to 3D using a linear system; recovery of hidden parts; and three-dimensional drawing tidying. It is described in more detail below. Grimstead's system makes the same assumptions as those listed above. In order to make the system interactive, further limitations were also introduced; these are discussed below.

A further advantage of interpreting sketches with only visible lines is that this makes it easier to incorporate work deriving from attempts to recognise real-world objects. Naturally, if the two-dimensional data is obtained from a photograph of a real object rather than from a sketch, hidden lines will not be visible. This process is inevitably more complex, as it cannot enforce assumptions on the real world in the way that a sketching interface can enforce assumptions on its user. It must be able to handle multiple objects, only some of which are of interest; it must be able to convert shading indicating differently-oriented faces into lines representing edges; it must be able to cope with noisy input. Nevertheless, it has one advantage which analysis of sketch input does not: it can assume that the input is valid, that it genuinely represents a real object, and that the projection is "drawn" correctly.

One such recognition system by Barrow and Tenenbaum [1] obtains line drawings by preprocessing greyscale pictures into region boundaries. A line may represent either a true edge between adjoining faces, or a depth discontinuity where one face obscures another. Their objective is to produce depth information on a per-pixel basis. The system can handle semicircles, cylinders and spheres, using Chakravarty's junction catalogue [4]. However, if the input sketch contains these, the iterative algorithms used converge too slowly to be useful for an interactive system. More interestingly for us are the special cases used for polyhedra. They use an iterative optimisation scheme which adjusts the Z coordinates of vertices in order to optimise a single global measure of regularity, preferring one of three measures: sum of squares of angles between faces, sum of squares of cosines of angles between faces; and sum of squares of ($2\pi-$ sum of angles at a vertex). All produce similar results.

Similar methods are also used in analysis of aerial photographs, although as Mayer's survey [28] shows, this field has subsequently become more specialised. Early work analysed single pictures by modelling buildings as firstly rectangular prisms [11] and later by grouping rectangular regions and parallel lines [29]. Artefacts such as skewed symmetry [15] are equally applicable to such pictures and to interpretation of sketches. This is a method for calculating the normal of a face given two axes on the face which are believed to be perpendicular in the 3D world. Given α and β, the angles of the two axes chosen such that the angle between them is obtuse, and expressed in the convention that anticlockwise is positive and zero is to the right along the horizontal axis, it can be shown that the face normal $[P, Q, R]$ is given by $P/R = \rho\cos\lambda$ and $Q/R = \rho\sin\lambda$ where $\lambda = (\alpha+\beta)/2$ and $\rho = \sqrt{-\cos(\alpha-\beta)}/\cos((\alpha-\beta)/2)$. The derivation assumes a correct orthographic projection. We know of no sensitivity analysis of the effect of inaccuracies in the projection, as would be present in a sketch, on the results.

More distant fields also produce results of interest. One such is the detection of motion of an object in an image, or of egomotion (movement of the viewpoint with respect to the image). Useful mathematical results can be found in Kanatani [17].

Interpretations of sketches generally involve some method of labelling lines. The most common line labelling method is that using the Clowes-Huffman [5, 13] catalogue. This assumes trihedral vertices, but can in principle be extended to curved surfaces using Chakravarty's catalogue [4] or Turner's [35]. The terminology used in this paper follows Chakravarty: junctions are L-junctions, T-junctions, W-junctions or Y-junctions, and lines are arrows (occluding), concave or convex. When it is necessary to indicate particular types of junction, these are extended by adding

the suffices a for an incoming arrow, b for an outgoing arrow, c for a convex line, and d for a concave line, reading clockwise; thus the legal trihedral junction labels are *Yccc, Yddd, Yabd, Wbca, Wdcd, Wcdc, Lac, Lab, Lcb, Lbd, Lba, Lda, Tbaa, Tbab, Tbac, Tbad*.

Various algorithms for line labelling have been proposed. In general, these are based on constraint propagation [17, 40]; an attempt to improve on these using genetic algorithms was unsuccessful [33]. An alternative method of labelling lines proposed by Kanatani [17] assumes not only that vertices are trihedral but that the object is orthogonal. It not only labels lines according to their concavity/convexity but also uses their orientation to classify them by axis as i, j or k lines. It is more successful in rejecting impossible objects, and more capable of providing useful information for later stages of the recognition process. The assumption of object orthogonality is however a major limitation.

A method for constructing the most plausible interpretation of a sketch, as determined by human perception, must of necessity be heuristic rather than mathematical in some places. Draper[7], in advocating sidedness reasoning, makes the interesting statement that such an approach is "more intuitively correct" than gradient and dual space algorithms. We are of the opinion that "intuitive correctness", although difficult to make precise, should also be an important criterion for assessing our method.

We consider that in a quick, natural sketch input system the advantages of drawing only the visible lines outweigh the disadvantages, so we take Grimstead's method as our starting point. We extend his approach by using symmetry and regularity relationships inferred from the sketch (a) to produce topologically more plausible completions of the hidden parts of objects, and (b) to enforce exact constraints on the geometry of the boundary-representation model. This paper gives an overview of our approach, and gives some examples for which it improves on Grimstead's approach; more technical detail is given in related papers [37, 38, 39].

Section 2 describes Grimstead's system. Section 3 outlines our extended system. Some results are shown in section 4, and section 5 lists possible future work.

The Figures are selected from our wide repertoire of test sketches as the ones which best illustrate particular points (rather than being the only ones we can solve!). These test sketches, currently numbering over 130, are culled from various sources, including [8, 34], all of the solid objects (but not the paper objects or wire-frames) from [21] and all of the planar objects from [43]. We continue looking for additional sketches of solid objects with planar faces and would welcome input from readers.

2. Grimstead's System

In order to infer depth information for the Y-, L- and W-junctions in a 2D sketch (we call this the *frontal geometry*), the following stages are common to all approaches summarised above: labelling of junctions and lines in the two-dimensional sketch; identifying loops or regions in the two-dimensional sketch; identifying artefacts of interest in the two-dimensional sketch; deducing constraints from the artefacts; and producing depth information such that the constraints are satisfied while the resulting shape fits the original sketch as well as possible (the order of these stages may differ). In addition, to produce a complete and valid solid model, both the topology and the geometrical positions of any hidden vertices, edges and faces must be deduced. Grimstead's system included a rudimentary method for doing this.

2.1. Incremental Line Labelling

In Grimstead's system, this stage incorporated both line labelling and region identification. Line labelling used Waltz's algorithm [40], and the Clowes-Huffman catalogue for trihedral vertices. Line-labelling both validated the sketch, and identified the faces and the edges which bound them. Little use was made of the line or junction labels in subsequent stages of the system.

2.2. Two-Dimensional Drawing Tidying

In Grimstead's system, this stage identified only one regularity, 2D line parallelism, and adjusted the sketch to improve this. The purpose was to enhance this regularity in order to make other artefacts which depend on parallelism easier to detect.

The algorithm used for this process allocated angles of 2D lines to buckets, and merged nearby buckets until it was evident which lines should be parallel. This bucketing process [8] is not very robust. A simplistic interpretation produces an unreliable implementation which can introduce problems with some types of sketch.

To illustrate this, consider the sketch in Figure 1. It is evident that edges A, B and C should be parallel, and any reasonable bucketing process will detect this. It is also evident to an intelligent observer that edges D, E and F should be parallel; however, depending on the quality of the sketch, D and E may be closer to G than to F, and an algorithmic process may therefore decide to make D, E and G parallel instead. It is also quite possible that F and G may be within the allowed angle below which buckets are merged, and made parallel; this is clearly nonsense geometrically. It is not clear how these problems should be resolved.

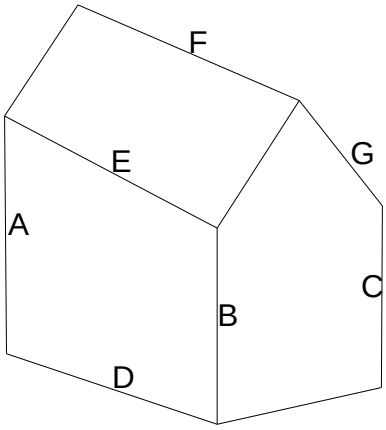

Figure 1. House

Even if the system correctly determines that edges D, E and F should be parallel, problems can occur. Suppose that lines A, B and C are well-drawn, being both parallel and of matching lengths, but that line F is misdrawn. A least-squares fit, requiring D, E and F to be parallel while trying to keep A, B and C parallel too and to preserve existing vertex positions, will change the orientations of lines D and E, and therefore change the lengths and possibly the orientations of A and B, while C remains unaffected.

Regularity may be lost, not gained, by two-dimensional tidying.

2.3. Generation of Frontal Geometry

Grimstead generated the frontal geometry of the object using a linear system. Let each face be of the form $Px + Qy + Rz + C = 0$, and each vertex (X, Y, Z). Constraining a vertex to lie on a face, and assuming $R = 1$ (no face is parallel to the projection direction), leads to a linear equation with the unknowns P, Q, C and Z. Such equations were generated for each vertex-face pairing.

Equations $m(Z_{n_1} - Z_{n_2}) = n(Z_{m_1} - Z_{m_2})$ (where m and n are the lengths of the two lines M and N, and $Z_{m_1}, Z_{m_2}, Z_{n_1}, Z_{n_2}$ are the depths of the start and end points of the lines) were generated for each pair of 2D parallel lines identified.

Equations $P = \rho \cos \lambda$ and $Q = \rho \sin \lambda$ were generated for each skewed symmetry artefact detected, using a variant of the equations described previously.

The system was then solved to give equations for each face using a weighted least-squares algorithm [2]. The original weighting of each equation was somewhat arbitrary. The solution process was iterative, with the equation with the largest residual error (the one which disagreed most with the consensus) being dropped at each subsequent iteration. The performance was assessed, once per iteration, using right-angle fit and minimum standard deviation of angles. Eventually, a self-consistent set of equations remained, and the output of the linear system gave P, Q and C values for each face.

The linear system approach as a whole suffered from several disadvantages:

- It was analytically opaque, in that if it did not work as well as was desired, modifications were made to weightings of equations, rather than to algorithms. The resultant weightings thus appear arbitrary, being the values which worked best for the test sketches used.
- It did not distinguish between constraints which *must* be met and equations for which a best-fit approach is adequate. Even the constraint that ensures that vertices lie on faces was only approximately satisfied, and had to be enforced at a later stage. Others, such as making two faces parallel or orthogonal, were also only approximately enforced.
- The variables being optimised were of different kinds, thus creating doubt as to whether the fit really was 'best'.
- The approach influenced unduly the choice of artefacts which were used in creating the fit, in that only those constraints which could be written as equations linear in P, Q, Z and C were admitted. For example, the assumption of corner orthogonality, first suggested by Mackworth [25] and used successfully in several systems, does not translate into a linear equation and so was not used.
- The skewed symmetry equations for P and Q were decoupled. The iterative least-squares fit might drop one and retain the other when rejecting inconsistent equations. A value of P or Q on its own is meaningless; both are needed for either to make sense.
- A hidden weighting was 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 $[P, Q, R]$. However, this is justifiable on the grounds that it is easier for the user to draw 'front-on' faces, so they will be more accurate.

Despite these disadvantages, the approach also had one major advantage: it ran sufficiently quickly to be interactive for sketches such as Figure 2.

2.4. Recovery of Hidden Parts

Grimstead's system assumed that every hidden face in the object met an occluding edge. The topological reconstruction of the hidden part of the object thus produced the simplest possible object, not necessarily the most plausible.

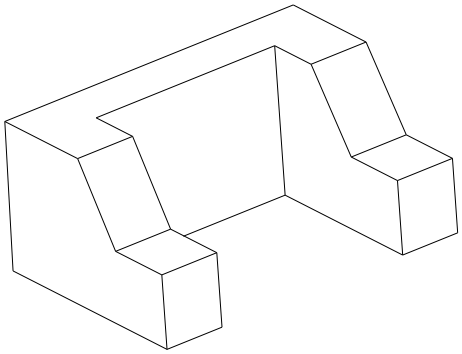

Figure 2. Grimstead's Bracket

Where three or more points on a hidden face existed in the 3D conversion of the visible sketch, the equation of the hidden face could be obtained directly. Where only two points existed, the equation was deduced on the basis of topological information.

Edges were extended through T-junctions, and new edges created behind the visible object from visible L-junctions. Groups of three such edges were then tested for consistency — if the three edges were allocated to three faces, an additional hidden vertex was created where the three faces intersected, and the three edges were removed from the set of edges requiring completion.

Grimstead's algorithm for recovery of hidden parts was successful in the test cases he used. However, Grimstead acknowledges that knowledge of the algorithm makes it simple to find sketches for which the algorithm does not work correctly. Figure 3 is one such — T-junction A would be joined to junctions B (correctly) and C (incorrectly).

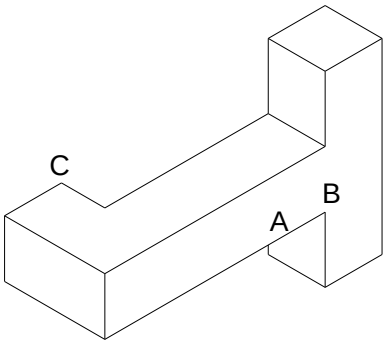

Figure 3. J Block

2.5. Three-Dimensional Drawing Tidying

Grimstead's final three-dimensional tidying process used the face equations generated above to give the 3D positions of each vertex by intersecting the appropriate faces, the values of X and Y from the sketch and Z from the linear system being discarded. This ensured a self-consistent boundary representation model.

One disadvantage with this approach was that the equations of hidden faces were derived from positions of visible vertices, themselves derived from a least-squares fit of various constraints, not all of which were fully-satisfied. Numerical errors were propagated at each stage, and using these error-prone values to calculate the positions of hidden vertices further compounded the error.

As already noted, the approach to geometrical completion of the object was that of a best fit, frequently producing edges which were nearly, but not quite, parallel, and corners which were nearly, but not quite, perpendicular. Regularity constraints such as parallelism of faces were not strictly enforced on the final solid.

The assumption that all hidden faces have at least two visible vertices was not always plausible — it is simple to construct objects where a more complex but symmetrical reconstruction is psychologically preferable to the minimal reconstruction given by Grimstead's system. However, the minimal approach may enable the system to produce an interpretation, albeit an undesired one, of a sketch where a more complex system may fail altogether (for example, the dodecahedron in Figure 4).

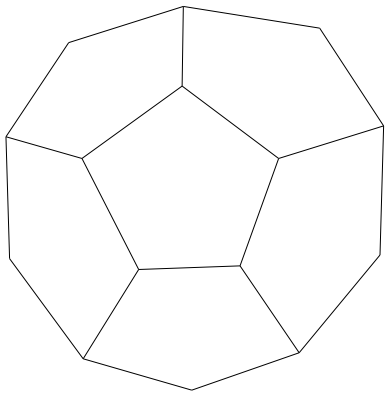

Figure 4. A Platonic Solid (Dodecahedron)

3. A System using Symmetries and Regularities

We consider the most serious defects in Grimstead's system to be that the topological completion of hidden parts can be implausible and that regularities implied by the sketch are not strictly enforced. Lockwood [24] describes the aesthetic appeal of symmetry, and Martin and Dutta [26] give various technical reasons why the possession of symmetry may be beneficial in designing shapes, and consider various approaches for making almost-symmetric shapes

symmetric. It is therefore plausible that sketches are intended to be symmetrical in some way, so we use implied symmetry and regularities in the sketch as the means to overcome these defects.

In many cases, there may be several plausible alternatives for reconstruction. We attempt to assign a *figure of merit* to each. This is a numerical value intended to represent the plausibility of a particular choice; higher values represent more plausible choices, and of a set of alternatives, the one with the highest figure of merit will be considered first by our algorithm.

Following Kettle [18], *symmetry* is characterised by the fact that it is possible to carry out operations which, whilst interchanging the positions of some or all of the atoms, give arrangements of atoms which are indistinguishable from the initial arrangement. Rotation, reflection and inversion are thus symmetry operations. Rotation axes and mirror planes are termed *symmetry elements*. Other artefacts which give clues to the structure of an object, such as parallelism or extrusion, which do not meet the strict definition of symmetry are here termed *regularities*. Identification of a symmetry element or regularity in an object produces one or more *constraints*, which limit the possible positions of the vertices.

Grimstead's system does not use mirror or rotational symmetry in object completion. When mirror and rotational symmetry artefacts are used, these affect the whole object, not just the visible part. For this reason, we advocate that the topological structure of the model including the hidden parts should be deduced *before* finalising the geometry of faces and vertices.

For convenience of implementation, our sketch interpretation system is therefore subdivided into four stages: labelling, symmetry identification, topological 3D completion and geometric finishing.

3.1. Labelling

This stage comprises a Clowes-Huffman line-labeller and identification of the regions of the sketch, including the 'background region', and follows the algorithm outlined by Kanatani [17]. The output is a list of vertices, edges and faces. For each vertex, the x and y coordinates, the junction type, and the incident edges and faces are listed. For each edge, the edge type, vertices and faces are listed. For each face, the vertices and edges are listed.

3.2. Symmetry Identification

We concluded above that two-dimensional tidying provides little benefit and some potential harm. It is the identification of symmetry artefacts which is of importance, not the small movements of junction x and y coordinates to improve them. Therefore, the two-dimensional tidying stage has been replaced by a stage where various symmetry artefacts and other global and local regularities in the sketch are identified.

Symmetry and regularity are used in two ways in later stages: in assisting the process of recovery of hidden parts, and in producing constraints to be met by the face normals and positions and vertex positions.

Figures of merit are assessed for potential symmetries and regularities, and also for hypotheses drawn from these artefacts at various stages of processing. The numerical values used are still the subject of experimentation [37]. They are intended to reflect the probability that an apparent artefact is a genuine feature of the object the user intended to sketch. When the system has a choice of actions, the one with the highest figure of merit will be tried first.

In general, figures of merit take into account the following four factors:

- the probability of a particular artefact occurring — some artefacts, such as face parallelism, are common, whereas others, such as C_5 rotational symmetry, are comparatively rare.

- a measure of how well a particular artefact fits in with what is already known about the object — regular pentagons may be quite rare, but become more plausible if several pentagons are already visible in the sketch.

- the probability that the artefact corresponds to the sketcher's intentions — where two lines are 1° apart, they are probably intended to be parallel, whereas if they are 10° apart, the intention is less certain.

- the complexity of the artefact — if an artefact is hard to draw (see, for example, the sixty-sided semi-regular solid in Figure 5), then anything approximating to it is probably intended to be that artefact.

Note that in some cases, such as partially-occluded faces, it will not be possible to determine whether or not the completed face has a particular symmetry or regularity. It is convenient to assume in such cases that the artefact is present, but to assign it a low figure of merit. An alternative approach might consider artefacts to have three states — completely present and thus likely; partially present and thus possible; and demonstrably absent on the basis conflicting evidence. We have not pursued this, as it is not clear that this is a sufficiently-detailed classification. For example, an artefact present locally, but contradicted by evidence elsewhere in the sketch, might still be of use in local reconstruction of hidden parts, and should neither be rejected immediately nor accepted unequivocally.

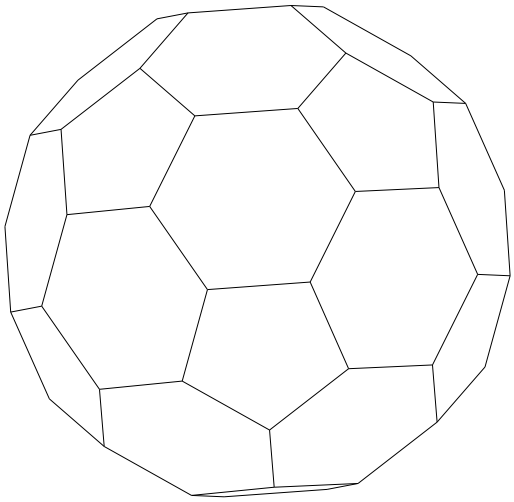

Figure 5. An Archimedean Solid (Truncated Icosahedron)

3.2.1 Bucketing

A bucketing process similar to Grimstead's is used to identify groups of parallel lines. In our system, bucketing requires lines which are to be grouped as parallel to be no more than 15° apart, and each group to be at least 15° from any other group. The angles are arbitrary, but sufficient to permit at least six groups of parallel lines in any sketch (Figure 6 has six lines, none of them parallel to one another), and up to twelve in well-drawn sketches.

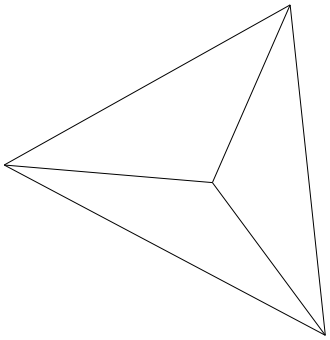

Figure 6. Tetrahedron

There are various possible outcomes of this:

- It may not be possible to allocate lines to distinct buckets in this manner (Figure 5 cannot be drawn in a way which meets the requirements above), in which case the sketch will be treated as having no parallel lines, but the system does not know whether or not this is intended.
- The bucketing process may produce an unacceptable result — two lines meeting at a junction are identified as being parallel. In this case also, bucketing is considered to have failed.
- Bucketing may succeed but each bucket contains one line (Figure 6 is a sketch where this occurs), in which case it is known that the lines are intentionally not parallel.
- Bucketing may succeed and identify candidate parallel lines.

An adaptive bucketing process could be developed to interpret badly-drawn sketches, as discussed in [37].

3.2.2 Frontal Geometry

Grimstead's system does not use mirror or rotational symmetry in object completion. If these are added, it becomes necessary to take account of the entire object when calculating spatial positions of vertices, not just of the visible part. For this reason, precise calculation of spatial positions is deferred until after the topology of the hidden part of the object has been deduced.

It is still necessary to have a topologically-correct completion of the frontal part of the object with added depth information. Accurate geometry is not required, but it is desirable that the depth-ordering of the visible vertices is correct, and necessary that adjacent vertices are ordered correctly (for example, skewed symmetry produces multiple solutions and selection of the correct one is based on knowledge of this ordering).

It can be observed in Figure 7 that lines from a $Wbca$ junction (A or C) to an Lba junction (B or D) are travelling away from the viewer. Similarly, lines from mostly convex junctions to mostly concave junctions are away from the viewer. Frequently, the longer the line, the further away from the viewer it travels.

Thus to achieve a correct ordering of adjacent vertices, we create a linear system with the number of equations given by the number of visible lines in the sketch and the number of unknowns given by the number of visible vertices in the sketch. The unknowns are the depth values; the equations are of the form $W * (Z_a - Z_b) = W * L * K$, with Z_a and Z_b being the depths of the near and far end points of the line, L being the length of the line in the two-dimensional sketch, K being a constant which depends on the junction types of the start and end points of the line and

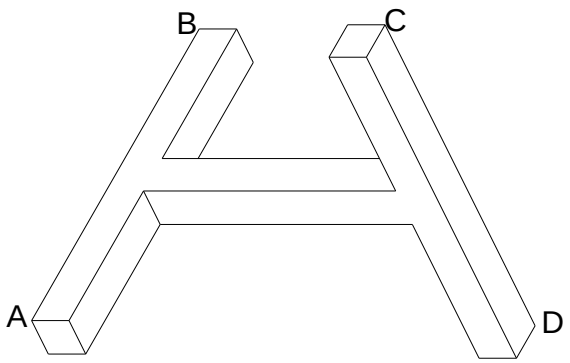

Figure 7. H Block

W (range 1..10) is a subjective measure of the confidence in the correctness of the value of K. Vertex 0 is constrained to an arbitrary fixed depth. The algorithm is described in more detail, and the K and W values tabulated, elsewhere [37].

For a simplistic method, this works surprisingly well in practice for most junction types, although it is found that Lcb junctions are frequently mispositioned. It fails entirely for sketches which are not graph-complete — if it is not possible to find a path from any edge to any other edge without making use of T-junctions, the resulting linear system cannot be solved. Various refinements have been tried, without significant improvement. One which has been adopted is to add further equations to the linear system to encourage lines in the same 2D bucket to be parallel in 3D. It has not been found desirable to follow this by forcing all visible vertices to lie on planar faces — this improves accuracy slightly for well-drawn sketches at the expense of exaggerating inaccuracies in poorly-drawn sketches.

3.2.3 Local Symmetries and Regularities

While it is not possible to determine the symmetry of an entire object from a sketch which only shows part of the object, it is possible to gain clues to the structure of the object by considering the symmetry of each region. Each region is usually a skewed view of all or part of the corresponding 3D face. Line lengths may not be preserved, and junction angles will almost certainly be distorted, but parallel lines remain parallel within sketching tolerance.

For two arrangements of atoms to be indistinguishable, the atoms occupying the corresponding positions in the two arrangements must be *compatible*. Two vertices are compatible if, in the final object, they have the same numbers of concave and convex edges. Two corners of a face are compatible if, when viewing a clockwise circumnavigation of the face from outside the object, they are both left turns or both right turns. Two edges are compatible if, in the final object, they are both convex or both concave (occluding edges will become convex); additionally, it must be possible to group their end points into two pairs of compatible vertices. Two faces are compatible in a particular orientation if they have the same number of edges, their edges when paired are compatible, their vertices when paired are compatible, and their corners when paired are compatible.

To identify rotational symmetry, it is noted that any axis of rotational symmetry must pass through either a vertex possessing C_3 symmetry (vertices are trihedral), an edge possessing C_2 symmetry about its mid-point, or a face possessing any rotational symmetry about its centre. It is therefore sufficient to identify such axes, which can be used later as "seeds" for potential global symmetry.

Triconnected junctions are candidates for local C_3 symmetry if their faces are compatible in the orientations which pair the junction with itself. Not all types of junctions need be analysed: for example, $Wcdc$ and $Yabd$ junctions cannot have C_3 symmetry as they have one concave and two convex edges in 3D. Of the biconnected junctions, Lba junctions are provisionally assumed to be candidates for C_3 symmetry as there is only one visible face.

The mid-points of non-occluding edges are candidates for local C_2 symmetry if the faces are compatible in the orientations which pair the edge with itself. At the time of writing, no artefacts are produced for the mid-points of occluding edges — this will be investigated in the future.

Candidate local rotational symmetry operations $C_2, C_3, C_4, C_5, C_6, ...$ for each region can be identified (our implementation stops at C_6). Firstly, a particular symmetry operation C_n is only a candidate if the region has $n, 2n, 3n, ...$ junctions. The candidate operation should be rejected if the face in its original orientation is not compatible with itself in its rotated orientation – for example, C_4 is only a candidate symmetry operation for an octagonal face if the junctions are either all convex or alternately convex and concave. Even-numbered candidate symmetry operations $C_2, C_4, C_6, ...$ should also be rejected if opposing lines are not parallel to within sketching tolerance. For C_5 symmetry operations, each set of five vertices $V_i, V_{n+i}, V_{2n+i}, V_{3n+i}, V_{4n+i}$ must be a regular pentagon; the undrawn line joining vertices V_{n+i} to V_{4n+i} must be parallel to the base of the pentagon, the line joining V_{2n+i} to V_{3n+i}, and so on for each edge taken as base in turn. Similar rules are used for C_6 symmetry.

Candidate mirror symmetry operations can again be localised to a seed, which in this case must be either an edge or a face with mirror symmetry. Face-based mirror planes can be further subdivided into vertex-to-vertex planes and edge-to-edge planes in faces with an even number of sides, and vertex-to-edge planes in faces with an odd number of sides.

A face is a candidate for local mirror symmetry if all vertices (other than any which are in the symmetry plane) can

be paired sequentially across the mirror axis, and lines joining these pairs are approximately parallel with one another and centred on the mirror axis. The angle of the axis, and the angle of the lines joining sequential pairs, are recorded (they are the input angles required by skewed symmetry) — see Figure 8, where line A is the candidate mirror line and

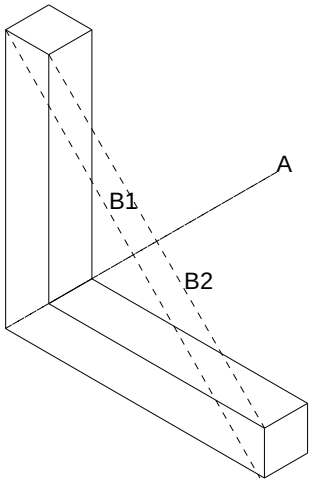

Figure 8. L Block

lines $B1$ and $B2$ join pairs of vertices. It has been found that generating "partial" mirror planes, where the face and its reflection are not compatible, can nevertheless be useful for the purposes of recovery of hidden parts — see Figure 9, where a reconstruction based on the partial mirror symmetry marked in dotted lines gives a topologically-correct completion.

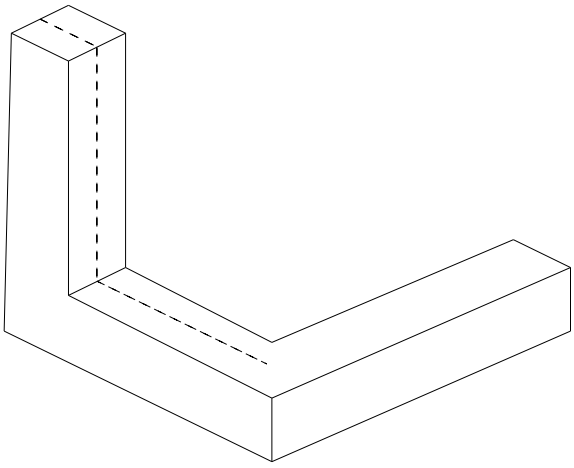

Figure 9. Z Block

Candidate mirror planes can be chained: where a face mirror plane terminates at an edge mid-point, it can be chained with another face mirror plane terminating at the same edge as in Figure 9. Where a face mirror plane terminates at a vertex, it can be chained with an edge mirror plane including that vertex, and vice versa, as in Figure 8.

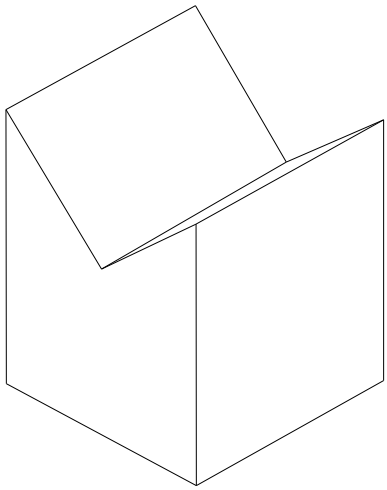

Figure 10. An Extrusion

3.2.4 Global Regularities and Symmetry Groups

An attempt is made to classify the sketch into one of several categories. A successful classification speeds up the later stages of processing considerably. Since an object may fall into more than one of the categories, a figure of merit is assigned for each classification.

Global symmetry and regularities are hypothesised on the basis of an agglomeration of local artefacts, described above, or on the basis of grouping of lines in the sketch into sets of parallel lines.

Line parallelism has been used in other systems [8, 14, 22]. We extend this idea to consider face parallelism and coplanarity. These regularities can be hypothesised either as a result of identification of a regularity which would imply them, or on the basis of having similar estimated numerical values for face constants P, Q and C. At this stage, we are concerned mostly with parallelism, and we identify various special cases.

The idea of face parallelism can further be extended to the hypothesis that the object is a right extrusion, with two identical and parallel faces joined by parallelograms. This is possible if no more than one visible face has other than four sides, all vertices are either on this special face or on edges which terminate on this face, and each junction label indicates that the vertex will have no more than one concave edge. If these requirements are met, the figure of merit for this hypothesis is based on the degree of parallelism of the edges which do not bound the special end face.

Right frusta such as Figure 11 can be identified by similar criteria, although in this case, the lines between the visi-

ble front face and the back face should meet at a single point rather than being parallel. The figure of merit for this hypothesis depends on how close to a single point these lines come, and is reduced if the object is likely to be an extrusion (this is in order to prevent the system miscategorising an extrusion as a frustum where the nearly-parallel lines meet a long way away).

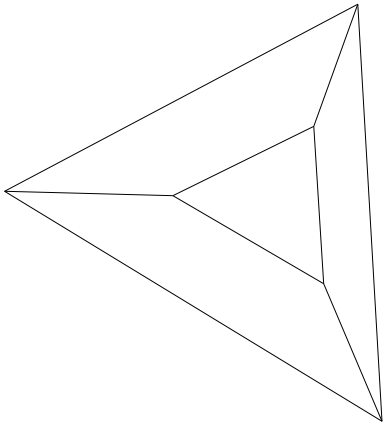

Figure 11. A Frustum

Orthogonality has been used in other systems. Lipson and Shpitalni [23] suggest that if a histogram of line angles is plotted and there are only three peaks, these probably correspond to the three perpendicular axes and the object is axis-aligned. In our system, if the bucketing process successfully identifies exactly three groups of line angles, the sketch contains no fully-visible odd-sided faces, and the fully-visible even-sided faces have four more right turns than left turns, the sketch is marked as being likely to be axis-aligned. The figure of merit of the axis-aligned hypotheses decreases as the peaks of the angle histogram become less sharp. Kanade [15] observes that the object can only be aligned to orthogonal axes if all three of the angles at Y-junctions are obtuse — since this is true of accurate projections but not necessarily of sketches, if the sketch contains Y-junctions with acute angles, the figure of merit for axis-alignment is reduced, but the hypothesis is not rejected entirely.

This concept can be extended to a category of semi-orthogonal objects such as the one shown in Figure 2. The sets of three line directions which meet at junctions are listed, and the number of occurrences (or potential occurrences for L-junctions) are counted. If one set occurs more than any other, this may indicate that these lines correspond to orthogonal axes. The figure of merit for this is calculated as for axis-aligned objects, but is reduced by a factor based on the proportion of non-occurrences of the chosen set of groups of line orientations.

In practice, many semi-orthogonal objects have a single mirror plane which is axis-aligned. Since this combination provides enough information to deduce the face normals of many hidden faces, it is detected as a special case. The figure of merit is based on that for semi-orthogonality, as above, the length of the mirror chain, and the quality of the pairing produced by propagating mirror pairings through the sketch. It would be simple to extend this further to a combination of semi-orthogonality with either a C_2 or C_4 rotation, which would also provide enough information to deduce the face normals of many hidden faces, but which appear to be less common in engineering practice.

Similar to the concept of an axis-aligned object is one where the mirror planes and rotation axes define three mutually-orthogonal axes — see for example Figure 12.

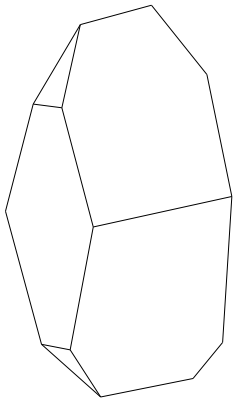

Figure 12. Tapered Wedge

Although the Platonic and Archimedean solids such as Figures 4 and 5 are useful test cases for general-case rotations, we prefer in practice to treat them as a special category. This is identified by comparing the numbers of sides (a, b, c) of the faces meeting at each vertex. If all vertices with three visible faces match (a, b, c), all vertices with two visible faces match (a, b), (a, c) or (b, c), and all vertices with one face match (a), (b) or (c), then the solid may be uniform. Additionally, all edges must be convex (or occluding).

As the system is extended, other special categories could be added — for example, if the entire sketch is consistent with C_n symmetry, $n > 4$. However, a survey of one engineering graphics textbook [43] yielded no examples in this category which could not equally well be classed as extrusions.

Identification of objects which can best be processed by splitting them into two smaller objects (for example, Figure 13) will be the subject of future work. We believe that the presence of junctions of two concave and one convex edge, such as $Wdcd$, will be a useful clue in identifying such objects.

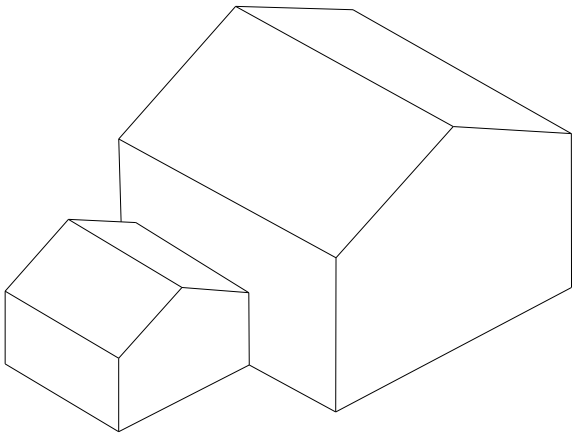

Figure 13. House with Extension

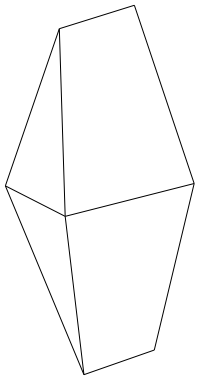

Figure 14. Wedge

3.2.5 Outputs

To the outputs of the labelling process, the artefact identification process adds: for each vertex, a provisional Z coordinate; for each edge, the mean angle of the bucket to which the line has been allocated; for each face, a list of the possible rotational symmetries, and an indication for each vertex as to whether it is convex or concave; a list of the mirror planes found in the object; and a list of the special-case configurations which the object matches. The presence of vertex Z coordinates and edge bucket angles are dependent, respectively, on estimation of frontal geometry and on bucketing.

3.3. Topological 3D Completion

We now attempt to recover the topology of the hidden parts of the object to complete the topology of a consistent and plausible 3D object. A greedy algorithm is used for this. While the back of the object remains incomplete, hypotheses are made on the basis of the current state of the object concerning possible *moves* towards completion. The various moves are assessed, and the highest-rated is chosen. Various alternative approaches using different ideas as to what constitutes a move (e.g. adding a vertex and enough edges to join it to the existing object, or adding an n-sided face with some of its vertices and edges being existing parts of the object) are described elsewhere [38] and their respective merits are assessed.

In any object which contains exactly two biconnected junctions and these junctions are not joined by an existing edge, it is reasonable to add a single hidden edge connecting them to complete the object. In trihedral objects, this can be considered to take priority over any other possible move. It is not such a good assumption for non-trihedral objects, as Figure 14 shows.

Similarly, in any object which contains exactly one biconnected junction and exactly one uncompleted T junction, and the biconnected junction is located approximately on a continuation of the line defining the T junction, it is reasonable to extend this line to the biconnected junction to complete the object.

If an object contains no remaining biconnected junctions or T junctions, it is reasonable to conclude that all vertices and edges have been found.

In any of these three cases, any additional faces required to make all vertices trifacial and all edges bifacial are generated by testing for unused loops of half-edges in the object. If this completes the object (we have no test case where it fails) the object is deemed to be topologically complete.

If none of these near-complete cases applies, moves are selected on the basis of generating candidates by the following methods and picking the one with the best figure of merit. Whenever a new hidden vertex is hypothesised, a prediction is made concerning its position. Since it is found that some hypotheses are better than others at predicting vertex positions accurately, this prediction has its own figure of merit. For trustworthy hypotheses, this may be equal to the figure of merit of the hypothesis as a whole; for hypotheses such as mirror chains, which are good at predicting topology but poor at geometry, it may be considerably lower. Whenever equivalent hypotheses are merged, the predicted positions are also merged taking account of this latter figure of merit.

Where a chain of one or more mirror planes has been identified, an attempt is made to create a list of correspondences between the current partially-complete object and the mirror-image which would be generated by reflection. Each vertex, edge or face in the mirror chain is matched to its equivalent after the reflection operation, and then an attempt is made to propagate the matches through the rest of the object (the propagation method is described in detail in [38]).

Atoms in the mirror-image which have no correspon-

dence to atoms in the original are assumed to correspond to extra atoms which will be required, and the appropriate hypotheses for generating them can be made.

Hypotheses based on chains of mirrors are allocated figures of merit based on the figure of merit of the chain — this increases with the number of mirror planes in the chain and also increases if the chain starts or ends at an object boundary. It is decreased by a factor based on the number of new vertices required by attempting to match the object with its mirror-image – the more new vertices are required, the greater the decrease.

Hypotheses derived from rotational symmetries use a similar algorithm, incorporating the same pairing mechanism, but using the rotating single face, combination of edge and two faces, or combination of vertex and three faces, as its seed.

If the object is potentially axis-aligned, lines are extended from incomplete vertices in the same manner as in Grimstead's system. Vertices are hypothesised where these lines intersect. The figure of merit of any individual hypothesised hidden vertex is increased if three of the proposed lines cross in approximately the same place. Depth information is generated by assuming that the hypothesised lines are parallel in 3D to visible lines in the same bucket.

Additionally, if two such lines are almost collinear, it is probable that they are really the same edge, and the two vertices should be connected. The figure of merit for this is high, decreasing as the angle between them increases and the distance between them increases.

Similar hypotheses, though with a lower figure of merit, are produced for semi-orthogonal objects.

If an object is potentially an extrusion, hypotheses can be made concerning a back end face of the object (it is identical to and parallel to the front end face, and includes all vertices which are not part of the front end face) and the sides (they are parallelograms, and include two vertices from the front end face and the corresponding two vertices from the back end face).

Similar hypotheses can be made if an object is potentially a frustum. It is topologically equivalent to an extrusion but with tapered sides.

No additional hypotheses are generated for other special cases such as uniform (Platonic and Archimedean) objects, as it has been found in the cases we have tested that the mirror and rotation hypotheses described above are adequate for completing the object.

Hypotheses can be made for T-junction completion. A true vertex must exist somewhere further along the line past where the T-junction intersects the occluding line. It is plausible that this vertex is directly connected to the next real vertex (vertices) obtained by following the occluded region (regions) for arrow (concave, convex) T-junctions around along the obscuring line; the figure of merit is based on the proportion of known faces in the object which would have the same number of sides as this method would predict. It is also worth generating the other hypothesis used by Grimstead: that for arrow T-junctions, the true vertex is connected to the next incomplete junction in the other direction around the object boundary; this may or may not be true in practice, so the figure of merit is low.

Both hypotheses are correct for Figure 15 — the former hypothesis correctly predicts the edge between A and B, and the latter hypothesis correctly predicts the edge between A and C. This is not the case for Figure 3, where the former hypothesis correctly predicts the edge between A and B but the latter hypothesis incorrectly predicts an edge between A and C.

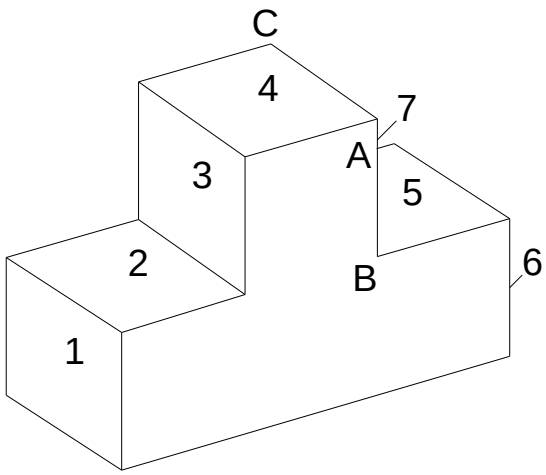

Figure 15. T Block

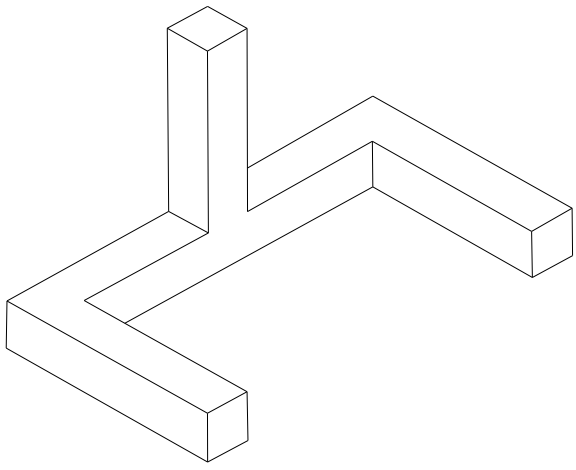

Figure 16. Double Z Block

Two local configurations of vertices lead to useful hypotheses. Firstly, given any two biconnected vertices A and B, if they are separated by a single triconnected ver-

tex C then adding a new vertex D to form a parallelogram $ACBD$ is plausible. The figure of merit is based on a fixed value for this type of move multiplied by the proportion of quadrilateral faces in the partially-completed object. It is reduced if the object is not believed to be orthogonal or semi-orthogonal. By way of illustration, this hypothesis would generate the back face of Figure 2 and the two back faces of Figure 17.

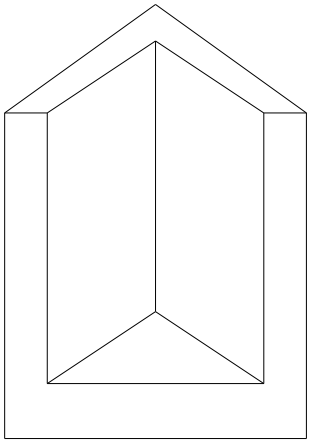

Figure 17. Angle Bracket

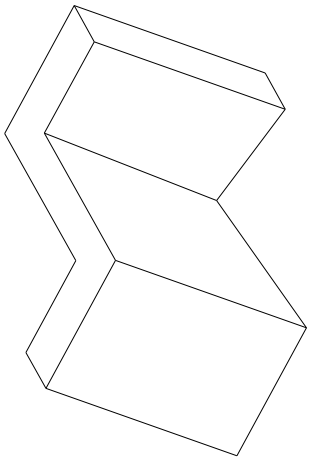

Figure 18. S Block

If two biconnected vertices A and B are separated by two triconnected vertices C and E, then adding an edge to join A and B to form the quadrilateral $ACEB$ is plausible. The figure of merit is based on a fixed value for this type of completion, the proportion of quadrilateral faces in the partially-completed object and the actual degree of parallelism of the supposedly parallel lines AC and EB. It is further reduced if the object is not believed to be orthogonal or semi-orthogonal. By way of illustration, this hypothesis

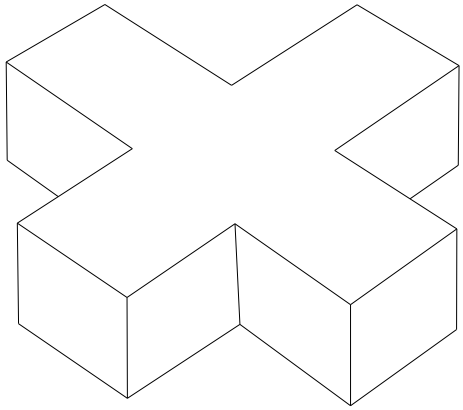

Figure 19. X Block

would complete the front, partially-obscured face of Figure 2.

Finally, if there are exactly three biconnected junctions remaining as in Figures 8 and 17, it can be hypothesised that there exists a single hidden vertex connected to all three.

Where two or more methods suggest the same move, or moves which appear to represent the same thing, the hypotheses are merged and the figure of merit reinforced. An example might be that two methods suggest creating a face by adding a new vertex, and completing a partially-occluded face by extending an existing line through a T-junction.

A hypothesis which places a new, supposedly hidden vertex in a *visible* location has its figure of merit reduced, in order to allow for the fact that the geometry of the partial object is at best provisional, and thus locations derived from it inaccurate, without rejecting better hypotheses which place the new vertex in a hidden location.

Figures of merit are also reduced if the hypothesised move conflicts with beliefs about the object as a whole, or for other reasons which reduce their plausibility [38].

A backtracking mechanism has been implemented so that the chosen move may be rejected either immediately, if after making the move the object is invalid, or after following all branches of the resulting tree of subsequent moves, if none of these results in a valid object. The former has been found useful, but the latter is only sporadically useful: in practice, a valid but poor move leads more often to a valid but unexpected object than to an invalid object.

A move is rejected entirely if, while the object remains incomplete, the resulting tree of subsequent moves is empty. It can also be rejected entirely if, after making the move, the object becomes invalid or too complex [38].

3.3.1 Outputs

To the outputs of the regularity recognition process, the topology completion process adds: a number of extra vertices; for each vertex, a depth estimate; a number of extra edges; a number of extra faces; and the connectivity information to provide a boundary-representation topological structure.

The topology completion process handles most of our test cases in interactive time, producing topologically valid solids; in a few cases, these solids are not geometrically realisable in 3D. We conjecture that those cases which do not complete in interactive time would eventually also produce topologically valid solids.

3.4. Geometric Finishing

This stage of processing takes a topologically-correct object and a group of symmetry and regularity artefacts, and aims to produce geometric information about the geometry of vertices and faces which satisfies as many of the plausible constraints as possible. Grimstead's three-dimensional tidying stage is thus extended to force the face equations of the object to conform to symmetry constraints identified in previous stages of the system, rather than merely find a best fit. The output of this stage is a complete list of geometric data for vertices, edges and faces which together with the previous topological information determines a boundary representation solid model.

The stage is subdivided into five substages: estimation of face normals; generation of constraints from symmetries and regularities; enforcing constraints on face normals; optimising face distances while enforcing constraints; and intersecting faces to obtain final vertex positions. For some special cases of objects, the face normals can be generated directly. It is still necessary to generate constraints from symmetries for use in the distance optimisation substage.

Six types of constraint are identified:

- a parallelism constraint requires two faces M and N to be parallel.
- a two-way perpendicularity constraint requires two faces M and N to be perpendicular.
- a three-way perpendicularity constraint requires three faces M, N and O to be mutually perpendicular.
- an angular constraint of angle ρ requires two faces M and N to be at a defined, non-perpendicular angle.
- a rotation constraint through angle ρ about the face normal through the centre of O requires three faces M, N and O to be oriented such that the rotation moves M to the position occupied by N.
- a mirror constraint requires two faces M and N and a mirror chain C to be related such that reflection through the mirror plane defined by C moves M to the position occupied by N and vice versa; additionally, all of the faces C_1, C_2, ... in the mirror chain must have their face normals in the mirror plane.

For the purposes of constraints, mirror planes are treated like faces: they have normals and distances, and can be constrained using the constraint types listed.

Constraints are allocated a figure of merit. The aim is to accommodate constraints in descending order of merit, with constraints being accepted if there are enough degrees of freedom left in the object to accommodate them, or if they agree with the geometry produced by previously accepted constraints.

In addition to the method described here, alternatives have been tried; these are discussed elsewhere [39].

3.4.1 Estimation of Face Normals

Vertex coordinates generated during the topological completion stage are used to provide preliminary estimates of face normals. For all but triangular faces (which can be solved explicitly) we use a linear system [2] where the variables are P, Q and C for the face (setting $R = 1$), and equations place each vertex on the face (with weightings which give priority to visible vertices). Adopting the convention that face normals are directed to look towards the face from outside the object, normals of visible (including partially-occluded) faces are provisionally assigned positive z-components, and normals of hidden faces are assigned negative z-components (this has the advantage of being quick, and thus repeatable: a more rigorous determination of the direction of face normals, deduced from convexity/concavity of edges, is performed once only at a later substage).

3.4.2 Special Categories of Object

For some of our more complex test cases, the general-case optimisation process described below was too slow to be considered interactive. There are also theoretical doubts about the general-case algorithm for face normals. We therefore use the object categorisation performed during artefact identification to take short cuts through the process for some categories of object.

If the object is believed to be a right extrusion (i.e. this category has a higher figure of merit than any other) the end caps are made parallel and the sides are made perpendicular to the end caps. If the front end face is believed to have mirror or rotational symmetry, the face normals of the sides are adjusted to preserve this symmetry. Since there are no further constraints on face distances, no further constraints

are generated for extrusions, and the general-case process is bypassed entirely. Similar methods are also used for frusta.

For axis-aligned objects, three perpendicular axes are formed, close to the average values of the appropriate face normals, and each face normal is constrained to the appropriate axis. The general-case face normal process is bypassed, but mirror and rotation constraints are still generated for use during the distance optimisation substage — this is required, for example, in order to ensure that the sides of the T-block (Figure 15, faces 2 and 5) are the same height and length.

For objects which are semi-orthogonal with an axis-aligned mirror plane, three perpendicular axes are formed, as for axis-aligned objects. The mirror plane is used to identify which of the hidden faces ought to be axis-aligned. The face normal of each face which should be axis-aligned is constrained to the appropriate axis. No part of the general-case process described below is bypassed. Optimisation of face normals takes place as for the general case, in order that relationships between non-axis-aligned faces can be established, but will be quicker by virtue of the knowledge already gained concerning parallelism and perpendicularity of axis-aligned faces and the relationships between non-axis-aligned faces and the mirror plane.

Since the uniform solids are finite in number and easily-identified, they are most easily treated as individual special cases. In practice, we adjust face normals iteratively by adding to each face normal the value which would be obtained by rotating adjacent faces by the appropriate angle (the general case is bypassed entirely). This provides a testbed for procedures also used in the general case and a template for handling non-uniform objects with multiple rotational symmetry.

Special case processing of objects with three axis-aligned sets of mirror planes, and non-uniform objects with clear higher-order rotational symmetry, is the subject of ongoing work. Similar methods to the above will be used.

3.4.3 Generation of Constraints

Constraints, firstly on the face normals and then on the face positions, are generated from the symmetry artefacts and allocated a figure of merit, both for the general case and for special categories which do not bypass the general case entirely.

Each rotational symmetry artefact produces constraints. The existing object is matched with the object which would be generated by applying the rotation about the rotating face O. For each face N in the rotated object which matches a face M in the existing object, a constraint is produced. This is normally a rotation constraint, except for the special case where $M = N$: $M = N = O$ is trivial and is ignored, and otherwise, M must be parallel to O, so a parallelism constraint is produced.

Hypotheses based on rotations are allocated figures of merit based on a fixed probability for each type of rotation, with C_4 and C_6 being given higher probabilities. The assessment is decreased by a factor based on the number of unmatched vertices left after attempting to match the object with its rotated equivalent — the more unmatched vertices which remain, the greater the decrease.

Each mirror chain generates constraints. The existing object is matched with the object generated by reflection through the mirror plane C. For each face N in the reflected object which matches a face M in the existing object, constraints are produced. One will normally be a mirror constraint, except for the special case where $M = N$, where M must be perpendicular to the mirror plane C and a two-way perpendicularity constraint is produced. Additionally, the mirror plane C is perpendicular to each face in the chain, C_1, C_2, ..., and two-way perpendicularity constraints are generated for these. As with rotations, the figures of merit are based on the quality of the match between object and reflection.

If an object is potentially axis-aligned, but this is not the most favoured interpretation, four perpendicularity constraints (three two-way and one three-way) are still generated for the faces meeting at each vertex. Their figures of merit are based on the orientation of the face normals involved and the overall figure of merit for the hypothesis that the object is axis-aligned.

In a similar manner, if the object is potentially semi-orthogonal, perpendicularity constraints are generated for those vertices at which the three presumed orthogonal axes meet. The figure of merit is calculated in a similar manner.

If the object is potentially an extrusion or a frustum, but these are not the favoured interpretations, a parallelism constraint is generated between the front and back end faces.

If bucketing is successful, a parallelism constraint is generated for any pair of faces which use edges from the same set of buckets.

If any two faces are approximately perpendicular or approximately parallel in the topological completion, it can be hypothesised that they are exactly parallel in the true object. Similarly, if the faces form an angle of $30°$, $45°$ or $60°$, or an angle the tangent of which is approximately a ratio of two small integers, it can be hypothesised that this is intentional and this angle should be retained in the true object.

3.4.4 Enforcing Constraints on Face Normals

In the general case, our aim is to find and enforce the best set of geometric constraints which can be satisfied simultaneously by the topology. This is a knapsack problem, exact solution of which is infeasible, so we use a greedy algorithm. Constraints are enforced in descending order of merit

until they specify a unique object:

- the constraint with the highest figure of merit is always accepted and enforced
- for each other constraint, in descending order of merit:
  - if logical reasoning using known information about the face normals can show that the constraint is necessarily valid, it is accepted;
  - if logical reasoning can show that the constraint is necessarily invalid, it is rejected;
  - if the constraint is already satisfied numerically by the object, it is accepted;
  - if enough degrees of freedom remain in the affected faces, the constraint is accepted;
  - if the object has enough degrees of freedom elsewhere, an attempt is made to adjust the existing face normals to accommodate the new constraint as well as all previous accepted constraints; if this succeeds, the new face normals are stored and the constraint is accepted; otherwise, the previous face normals are restored;
  - otherwise, the constraint is rejected.

Logically, the relationship between two faces is considered to be one of three: they are either parallel, perpendicular, or at some other angle. A set representing the three possible relationships is stored for each pair of faces. Initially, all three relationships are possible, except that faces sharing an edge cannot be parallel. If the object is axis-aligned no pair of faces can be other than parallel or perpendicular. When new information is added, it is propagated to all faces parallel to the faces being updated. Interrogating the status of relationships (known to be true, possible but uncertain, or known to be false) is straightforward on the whole; the most complex inference made is that two faces are necessarily parallel if that is still a possible relationship and they are both perpendicular to the same two mutually-perpendicular faces.

The logical reasoning sets an upper limit to the number of orientational degrees of freedom of a face [39]. To find the available number of degrees of freedom, this must then be reduced to allow for previously-accepted mirror, rotational and angular constraints; this is a non-trivial problem to which we know of no perfect solution.

Methods of calculating the number of degrees of freedom after satisfying a number of geometric constraints have been the subject of several studies. Sugihara [34] lists a number of these. More recently, Kramer [19] has shown the difficulty of allowing for geometrical coincidences in two dimensions. Latham [20] has methods for resolving this issue which appear to improve on existing algorithms, but these are based on symbolic reasoning and are therefore unsuitable for current interactive systems. Whitely's [42] extension to 3D of a method which is successful in 2D is unsatisfactory in that it is intolerant of sketching inaccuracies and that it is limited to triangular and quadrilateral faces.

Rather than attempt to improve further, we recognise that since perfect conversion of sketches to objects is formally impossible, and that we hope for most sketches to bypass the general case, an algorithm which works more often than not for the remainder is acceptable. Therefore, we use a practically effective algorithm which would be hard to justify theoretically [39].

Different methods of adjusting face normals have been tested [39]. Some are quicker than others, but none is quick enough to be considered interactive for objects which depend mostly on numerical, rather than logical, reasoning.

3.4.5 Optimising Face Distances

The objective of this substage is to provide values of face distances for each face which fit the faces of the object as closely as possible to the visible vertices while enforcing accepted constraints. The starting-point is an n-dimensional iterative optimisation [30, 31] where n is the number of faces in the object and the variables being optimised are the face distances. The objective function being optimised is the sum of the squares of the 2D distances between the predicted x and y coordinates of each vertex and its actual position in the original sketch.

Mirror constraints and C_2 rotation constraints affect this in two ways. Firstly, if the mirror plane or rotation axis is perpendicular to the two faces M and N, and M and N are parallel, then they must also be coplanar. The distance for N is then dropped from the optimisation (the number of variables is decremented) and set equal to that for M on each iteration (see faces 2 and 5 in Figure 15). Secondly, if faces M and N are parallel, and two other paired faces M' and N' are also parallel to one another and to M, then the distances obey the equation $C_M - C_{M'} = C_{N'} - C_N$. The distance for N' is then dropped from the optimisation (the number of variables is again decremented) and set equal to $C_M + C_N - C_{M'}$ on each iteration (see faces 1 and 3, and 6 and 7, in Figure 15).

In addition, mirror planes must bisect the faces they lie on, producing constraints on the distances of faces meeting the edges of the mirroring face. This is the subject of ongoing work.

The use of rotational symmetry in adjusting face distances is the subject of ongoing work. This is needed for

sketches such as the dodecahedron, which has a completely hidden face for which visible vertex positions provide no information.

The assumption of axis alignment, and the perpendicularity constraints it produces, have no effect on distances.

3.4.6 Intersecting Faces

This matches Grimstead's three-dimensional tidying process — the x, y and z coordinates of each vertex are finally recalculated from the equations of the three faces on which it lies.

4. Results

This section gives some examples of objects generated from sketches, and makes general observations about the merits of our approach.

4.1. Experimental Results

Table 1 collates the results obtained using all of the trihedral sketches illustrated in the Figures in this paper and a few other examples. They are grouped by category, and within a category are in order of increasing complexity. A tick $\surd$ indicates that results are as expected, and a cross $\times$ that they are not, in which case see the referenced note. Timings are in milliseconds and were obtained on a Sun Ultra 10 running Unix.

Sketch (Figure no.)	Topology	Geometry	Time (ms)
Cube (–)	$\surd$	$\surd$	30
L Block (8)	$\surd$	$\surd$	30
T Block (15)	$\surd$	$\surd$	60
J Block (3)	$\surd$	$\surd$	150
Z Block (9)	$\surd$	$\surd$ (A)	70
S Block (18)	$\surd$	$\surd$ (B)	70
X Block (19)	$\surd$	(C)	570
double Z Block (16)	$\times$ (A)	N/A	N/A
H Block (7)	$\surd$	$\surd$ (D)	560
Notched Extrusion (10)	$\surd$	$\surd$	10
Hexagonal Prism (–)	$\surd$	$\surd$	50
House (1)	$\surd$	(E)	40
Triangular Frustum (11)	$\surd$	$\surd$ (F)	10
Hexagonal Frustum (–)	$\surd$	$\surd$ (F)	20
Tetrahedron (6)	$\surd$	$\surd$	10
Dodecahedron (4)	$\surd$	$\surd$ (G)	370
Truncated Tetrahedron (–)	$\surd$	$\surd$ (H)	50
Truncated Octahedron (–)	$\surd$	$\surd$ (G)	390
Truncated Icosahedron (5)	$\surd$	$\times$ (G)	28220
Angle Bracket (17)	$\surd$	$\surd$ (B)	130
Grimstead's Bracket (2)	$\surd$	$\surd$ (I)	2530
Tapered Wedge (12)	(B)	N/A	490
House with Extension (13)	$\times$ (C)	N/A	N/A

Table 1. Results of Sketch Interpretation

4.1.1 Topology

Our system reconstructs the correct topology for the majority of the Figures.

(A) At an early stage of topological reconstruction of the double-Z-block, the hypothesis of a rectangular base to the near wing of the block has a high figure of merit, and is accepted. The expected object can be produced using different tunings for figures of merit.

(B) The tapered wedge produces a topologically-valid but unexpected interpretation based on regular triangles and hexagons.

(C) The absence of symmetry in the sketch makes the weightings of figures of merit for hypotheses arbitrary. Incorrect hypotheses are accepted.

4.1.2 Geometry

Our system reconstructs the expected geometry in over half of the Figures. In some cases the expected geometry is not ideal, since the symmetry operation which would produce the ideal geometry is not one we detect. Where unexpected geometry is added to correct topology, this is usually because a higher degree of symmetry has been imposed in preference to a lower-degree but intuitively more correct symmetry.

(A) The two ends of the Z-block may not be the same length — although the object as a whole has a C_2 rotation, this is centred on an occluding edge so is not detected.

(B) The two end sections of the S-block are the same width, but the centre section may be a different width — no symmetry relation exists between them. The same is true of the width of the U-shaped face of the Angle Bracket.

(C) The C_4 rotational symmetry is enforced in preference to the axis-aligned mirror symmetry planes, so the resulting geometry is not as expected.

(D) S_n symmetries are not detected, so the two end-pieces of the H-block may be of different lengths.

(E) The sketch of a house is converted to a pentagonal prism (C_5 rotations are considered hard to draw, so the

figure of merit for the C_5 rotation is higher than that for sides being parallel).

(F) The height of the frustum is arbitrary since the sketch contains no clues.

(G) The rear face of the dodecahedron is too large — its position is not constrained by any visible vertices. Similarly, the rear square of the truncated octahedron is too small. The problem is amplified in the truncated icosahedron, where several faces in the topological reconstruction contain no visible vertices. Treating uniform objects as special cases would enable correct interpretation of such sketches, but would not cure the problem in the general case.

(H) The triangular faces of the truncated tetrahedron are different sizes, reflecting inaccuracies in the original sketch.

(I) The angle between the axis-aligned and angled faces is to some extent arbitrary, and not necessarily the one which would be obtained by 'de-skewing' the end face.

4.2. Conclusions

The set of sketches which can be interpreted plausibly improves on Grimstead's method. In the majority of cases, the goal of producing the expected interpretation is achieved. We believe that a further improvement can be achieved by refining the figures of merit.

On the whole, the system meets the goal of interactive response times. The exceptions occur when a poor choice of move is made during the early stages of topological reconstruction. This can make the process very slow — of the order of minutes — before backtracking brings the system back to a more sensible choice.

We have added one more assumption concerning the sketch to our initial set — that it is from a viewpoint from which the visible junctions are graph-connected. An alternative method of generating provisional depth information would remove this restriction.

Objects which meet one of the special-case categories are in general categorised correctly. More commonly, objects will meet the requirements of several of the special-case categories, and the category to which they are allotted can be arbitrary. This can also vary depending on how well-drawn the sketch is — different versions of Figure 1 are categorised as a frustum, a semi-axis-aligned sketch with a mirror plane, or an extrusion.

The two non-axis-aligned Platonic solids, the tetrahedron and dodecahedron, are reconstructed correctly, although the positioning of faces containing no visible vertices is suspect. Successful conversion of the Archimedean solids has also been achieved but is not interactive for the most complex of these and the geometry is poor. Treating them as special cases throughout would lead to perfect conversion in interactive time.

We have no extrusion test case which fails. However, sketches of hexagonal prisms are categorised as extrusions (which they are) rather than as uniform solids (which is also true, and gives more information).

Intentionally axis-aligned objects can be detected (rather than assumed, as in [17]) with a reasonable degree of confidence. They are converted to exactly axis-aligned boundary representation models (rather than approximately axis-aligned, as in [8]) interactively. We have no axis-aligned test case which consistently fails, but in some cases correct topological interpretation depends on the numerical values of the figures of merit assigned to various hypotheses — the double-Z-block is one such. Poorly-drawn versions of the T-block and other similar objects intended to be axis-aligned are generally categorised as extrusions, and are reconstructed correctly via that route.

Simple semi-axis-aligned objects with mirror planes are interpreted correctly, but more slowly and less reliably than the other special categories. Where the number of non-axis-aligned faces is low, it is still quick enough to be considered interactive.

General-case objects meeting no special categories are sometimes interpreted successfully, with sketches containing ten or twelve junctions being interpreted in interactive time. The time taken rises sharply with the number of junctions. Reconstruction of topology is more stable than reconstruction of geometry. In some cases, the system can add symmetry where it was not intended, further distorting the geometry.

5. Future Work

Further investigation into the numerical values of figures of merit is planned, with the aim of correctly converting those of our test cases which are currently failing.

Thereafter, it is planned to attempt to extend the range of sketches which can be interpreted by allowing non-trihedral vertices. It is also planned to attempt to extend the range of permitted sketches by allowing sketches of objects with hole loops, and to extend the scope of reconstruction by partitioning difficult sketches into smaller, categorisable subsketches.

Further in the future, a system which can deal with simple curved objects may be attempted.

Acknowledgements

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

We also wish to thank the referees, whose observations have helped us to clarify the paper.

References

[1] H.G.Barrow and J.M.Tenenbaum, *Interpreting Line Drawings as Three-Dimensional Surfaces,* Artificial Intelligence **17**, 75–116, 1981.

[2] L.Bauer, *Elimination with Weighted Row Combinations for Solving Linear Equations and Least Squares Problems,* Contribution I/9 in ed. J.H.Wilkinson and C.Reinsch, Handbook for Automatic Computation **II**, Linear Algebra, Springer-Verlag, 1971.

[3] R.P.Brent, *Algorithms for Minimization without Derivatives,* Prentice-Hall, 1973.

[4] I.Chakravarty, *A Generalised Line and Junction Labelling Scheme with Applications to Scene Analysis,* IEEE Trans. Pattern Analysis, Machine Intelligence **1**(2), 202–205, 1979.

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

[6] H.T.Davey and R.J.Wilkins, *Engineering Drawing,* Macdonald and Co, 1952.

[7] S.W.Draper, *The Use of Gradient and Dual Space in Line-Drawing Interpretation,* Artificial Intelligence **17**, 461–508, 1981.

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

[9] I.J.Grimstead and R.R.Martin, *Incremental Line Labelling for Sketch Input of Solid Models,* Computer Graphics Forum, **15**(2), 155–166, 1996.

[10] I.J.Grimstead and R.R.Martin, *Creating Solid Models from Single 2D Sketches,* in ed. C.Hoffmann and J.Rossignac, Proceedings of Third Symposium on Solid Modelling and Applications, 323–337, ACM Press, 1995.

[11] M.Herman and T.Kanade, *The 3D MOSAIC Scene Understanding System: Incremental Reconstruction of 3D Scenes from Complex Images,* in Proceedings of the Image Understanding Workshop, 137–148, 1984.

[12] E.Horowitz and S.Sahni, *Fundamentals of Computer Algorithms,* Computer Science Press Inc, 1979.

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

[14] D.L.Jenkins, *The Automatic Interpretation of Two-Dimensional Freehand Sketches,* PhD Thesis, Cardiff University Computer Science Department, May 1992.

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

[16] K.Kanatani *Reconstruction of consistent shape from inconsistent data: Optimisation of* $2\frac{1}{2}D$ *sketches.* International Journal of Computer Vision **3**(4), 261–292, 1989.

[17] K.Kanatani, *Group-Theoretical Methods in Image Understanding,* Springer Series in Information Sciences No 20, Springer-Verlag, 1990.

[18] S.F.A.Kettle, *Symmetry and Structure,* Wiley, 1985.

[19] G.A.Kramer, *Solving Geometric Constraint Systems,* MIT Press, 1992.

[20] R.S.Latham, *Combinatorial Algorithms for the Analysis and Satisfaction of Geometric Constraints,* PhD Thesis, Department of Computer Science, Brunel University, December 1996.

[21] Y.G.Leclerc and M.A.Fischler, *An Optimization-based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames,* International Journal of Computer Vision, **9**(2) 113–136, 1992.

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

[23] H.Lipson and 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.

[24] E.H.Lockwood, *Geometric Symmetry,* Cambridge University Press, 1978.

[25] A.K.Mackworth, *Model-driven Interpretation in Intelligent Vision Systems,* Perception **5**, 349–370, 1987.

[26] R.R.Martin and D.Dutta, *Tools for Asymmetry Rectification in Shape Design,* Journal of Systems Engineering **6**, 98–112, 1996.

[27] D.Marr, *Representing Visual Information,* in A.Hanson and E.M.Riseman, Computer Vision Systems, Academic Press NY, 1978.

[28] H.Mayer, *Automatic Object Extraction from Aerial Imagery - A Survey Focusing on Buildings*, Computer Vision and Image Understanding **74**(2), 138–149, 1999.

[29] R.Mohan and R.Nevatia, *Using Perceptual Organisation to Extract 3D Structures*, IEEE Transactions on Pattern Analysis and Machine Intelligence **11**(11), 1121–1139, 1989.

[30] J.A.Nelder and R.Mead, *Computer Journal*, **7**, 308–313, 1965.

[31] W.H.Press, S.A.Teukolsky, W.T.Vettering and B.P.Flannery, *Numerical Recipes in C, The Art of Scientific Computing*, Second Edition, Cambridge University Press, 1994.

[32] Y.Shirai, *Analysing Intensity Arrays Using Knowledge About Scenes*, in The Psychology of Computer Vision, ed P.H.Winston, McGraw-Hill, New York, 1975

[33] L.A.E.Stevens, *Genetic Algorithm to Optimise Line Labelling*, MSc dissertation, Cardiff University, 1994.

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

[35] K.Turner, *Computer Perception of Curved Objects using a Television Camera*, PhD Dissertation, Edinburgh University, November 1974.

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

[37] P.A.C.Varley and R.R.Martin, *Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch — Frontal Geometry and Sketch Categorisation* , 1st Korea-UK Joint Workshop on Geometric Modeling and Computer Graphics, April 2000, to appear.

[38] P.A.C.Varley and R.R.Martin, *Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch — Topology of Hidden Parts*, 1st Korea-UK Joint Workshop on Geometric Modeling and Computer Graphics, April 2000, to appear.

[39] P.A.C.Varley and R.R.Martin, *Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch — Geometric Finishing*, 1st Korea-UK Joint Workshop on Geometric Modeling and Computer Graphics, April 2000, to appear.

[40] D.M.Waltz, *Generating Semantic Descriptions from Drawings of Scenes with Shadows*, Tech Report AI-TR-271, M.I.T., Cambridge USA, 1972.

[41] C.E.Weatherburn, *Elementary Vector Analysis*, Bell, 1921.

[42] W.Whitely, *Matroid and Rigid Structures*, in ed. N.White, *Matroid Applications*, Cambridge University Press, 1991.

[43] H.W.Yankee, *Engineering Graphics*, Prindle, Weber and Schmidt, 1985.