Artificial Intelligence 108 (1999) 31–67

Artificial Intelligence

Linear-time algorithms for testing the realisability of line drawings of curved objects

Martin C. Cooper [1]

University of Toulouse III, IRIT, 31062 Toulouse, France

Received 23 June 1997

Abstract

This paper shows that the semantic labelling of line drawings of curved objects with piecewise C^3 surfaces is solvable in linear time. This result is robust to changes in the assumptions on object shape. When all vanishing points are known, a different linear-time algorithm exists to solve the labelling problem. Furthermore, in both cases, all legally labelled line drawings of curved objects are shown to be physically realisable.

However, when some but not all of the vanishing points are known, when the drawing is an orthographic projection of a scene containing parallel lines or when we wish to minimise the number of phantom junctions, the labelling problem becomes NP-hard. The introduction of collinearity constraints also renders the labelling problem NP-complete, except in the case when all vanishing points are known. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Line drawing labelling; Constraint satisfaction problem; NP-completeness

1. Introduction

Line drawing interpretation is a classic problem in Artificial Intelligence [8]. Assigning semantic labels to the lines of a drawing is part of the major problem in computer vision of recovering the three-dimensional information lost when a 3D scene is projected into a 2D image. Semantic labels, such as “concave”, “convex” or “occluding”, assigned to lines in a drawing are valuable clues for the later complete reconstruction of the 3D scene.

The first success of the work on line drawing labelling was the ability to filter out certain drawings which were not realisable as physically possible 3D scenes, using only the information contained in the line junctions in the drawing. Early work in this area was

[1] Email: cooper@irit.fr.

PII: S0004-3702(98)00118-0

limited to line drawings of polyhedra. Sugihara elegantly brought an end to this line of research by producing necessary and sufficient conditions for the physical realisability of a line drawing of a polyhedral scene [22,23].

Malik [15] gave a complete catalogue of physically possible labelled junctions for objects with curved C^3 surfaces. We will show that this catalogue provides a necessary and sufficient condition for the realisability of a line drawing of objects with piecewise C^3 surfaces. The freedom to choose the shape of surfaces provides the key to this result.

It has been shown that testing the realisability of a line drawing of a polyhedral scene is NP-complete [13]. We will show that relaxing the assumptions on the scene to allow curved objects with piecewise C^3 surfaces renders the realisability problem solvable in linear time. In the terminology of the Constraint Satisfaction Problem, pairwise consistency [10] is a decision procedure for realisability. The major reason why the NP-complete problem for polyhedral objects becomes solvable in linear time for curved objects is that the difference between "occluding" and "convex" lines can no longer be propagated from one junction to another adjacent junction, due to the possible presence of undetected C junctions along the line.

Berkeley [1] pointed out 300 years ago that a two-dimensional projection of a three-dimensional scene is infinitely ambiguous. In order to recover depth information from an image we must make assumptions about the objects in the scene and about image formation. Strong assumptions on object shape, such as planar surfaces or surfaces of rotation, may be too restrictive and seriously limit the number of possible applications. On the other hand, weak assumptions will not provide sufficient information to provide an unambiguous interpretation. We can calculate the number of bits of information per line-end provided by the catalogue of junction labellings, and hence deduce the expected number of random incorrect interpretations which will be found for a drawing composed of a given number of lines.

This analysis, together with a study of certain pathological examples indicate that, in many cases, a catalogue of labelled junctions does not provide sufficient information to provide an unambiguous interpretation of drawings of curved objects. We therefore study other sources of information, in particular, knowledge of vanishing points and collinear points and lines.

Several versions of the line drawing interpretation problem are examined in this paper:

- different versions of the classic problem of the labelling of a line drawing of objects with C^3 surfaces;
- the line drawing interpretation problem when the orientations in space of all tangents to line-ends at all junctions are known (for example, from analysis of vanishing points);
- the general line drawing interpretation problem when some but not all orientations of line-ends are known.

For each case, the impact, on the computational complexity of the problem, of including collinearity constraints is also considered.

Throughout this paper it is assumed that scenes, the objects which compose them and the projection transformation satisfy the following conditions:

(1) Objects have C^3 surfaces separated by edges representing a discontinuity of the surface normal.

(2) In any neighbourhood centred on a vertex or an edge, the object is topologically equivalent to its interior (thus outlawing lamina, filaments and objects whose components only touch along a line or at a point).
(3) Vertices are trihedral (formed by the intersection of 3 surfaces).
(4) No edge or surface is tangential to another edge or surface.
(5) The drawing is a perfect projection of the visible edges in the scene.
(6) A general viewpoint and general object positions are assumed. A small perturbation in the position of the viewpoint or of any of the objects does not change the configuration of the drawing.

Several authors have studied non-trihedral vertices [15,25]. In a previous paper, the author gave a linear-time realisability test under the weaker assumption that edges and surfaces may be tangential [5]. Waltz [25] studied scenes which do not satisfy the general viewpoint or the general object position assumptions. Falk [8] used object models and Shapira and Freeman [20,21] used multiple views of the same object, to overcome the problem of imperfect projections. The interpretation of drawings of origami objects, which do not satisfy condition (2) above, has also been studied [12].

2. Catalogue of labelled junctions

A curved line in a drawing may be the projection of any one of an infinite family of 3D curves. Our assumptions on object shape place no constraints on the corresponding 3D curve, apart from disallowing pathological cases (such as discontinuous 3D curves projecting into continuous 2D curves, for example). The intuition of the pioneers in the domain of line drawing interpretation by machine was that the information that can be recovered from a drawing is mainly concentrated in the way lines meet at junctions and furthermore that this information concerns not the equations of 3D edges but the way the pairs of surfaces meet to form edges [3,9].

Figs. 1(a), (b) and (c) show the ways that pairs of surfaces may meet to form 3D edges. These edges are assumed to be viewed from above. The label "+" signifies a convex edge: the two surfaces subtend an angle $\theta > \pi$ on the viewer's side of the edge. The label "−" signifies a concave edge ($\theta < \pi$). The label "→" signifies an occluding edge with both surfaces projecting to the right of the line as we follow the direction of the arrow. A curved surface may also occlude itself, as shown in Fig. 1(d). The locus of points at which the line of sight is tangential to the surface is called an extremal or phantom edge, since its 3D position varies with changes in the viewpoint. We use the term "extremal edge" in this paper. Its label is a double-headed arrow, but for typographical reasons we use $\Rightarrow$ to represent this extremal label within the text of this paper. Since the occluding and extremal edges, shown in Figs. 1(c) and (d) have reflected versions in which the arrows point downwards, this makes six distinct labels in all.

Under our assumptions, there are only four ways that lines can project into junctions in the drawing: three surfaces meet in 3D (L, W and Y junctions), an edge occludes another (T junction), a surface cuts a self-occluding curved surface (C, curvature-L and 3-tangent junctions), a curved surface smooths out so that it no longer occludes itself (terminal junction). Fig. 2 shows an example of a drawing containing all eight different

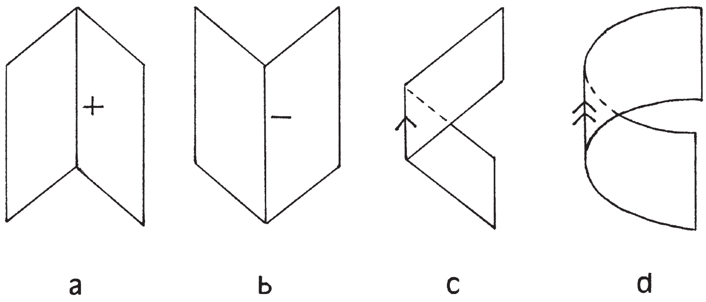


Fig. 1. The semantic labelling of lines: (a) convex edge; (b) concave edge; (c) occluding discontinuity edge; (d) extremal edge.

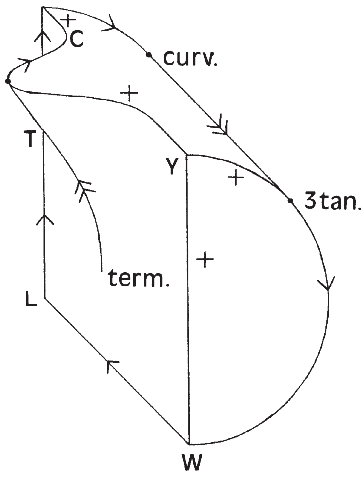


Fig. 2. An example of a labelled line drawing of a curved object containing terminal, L, C, curvature-L, 3-tangent, T, Y and W junctions.

junctions. A black dot on a line signals a discontinuity of curvature. At a 3-tangent junction there is continuity of curvature between the lines labelled $+$ and $\rightarrow$, but a discontinuity of curvature between the line labelled $\Rightarrow$ and the $\rightarrow$ line. At a C junction there is no discontinuity of curvature. This renders the junction invisible; the C junction is often known as a phantom junction. The presence of a discontinuity of curvature at curvature-L and 3-tangent junctions was proved formally by Nalwa [18]. The possibility of concave curvature-L and 3-tangent junctions was shown in a previous paper [4]. Both W and Y junctions are formed by the meeting of three lines: at Y junctions there is no angle which exceeds π; at W junctions there is one angle which exceeds π. W junctions are also known as E junctions by some authors.

Fig. 3 gives the catalogue of labelled junctions, as given by Malik [15]. The list of labellings for each junction is given on the right hand side of the figure. The label l_i

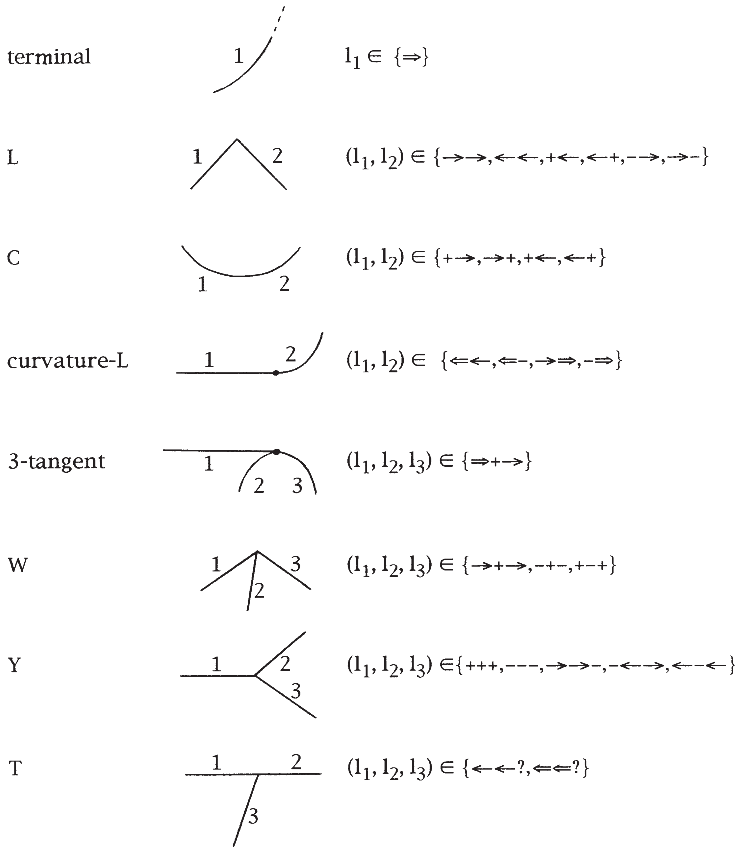


Fig. 3. Catalogue of junction labellings for objects with C^3 surfaces.

represents the label for the line i. A question mark can be any of the six labels ($+$, $-$, $\rightarrow$, $\leftarrow$, $\Rightarrow$, $\Leftarrow$).

Both curvature-L and 3-tangent junctions have reflected versions. For example, by definition of a curvature-L junction, if line 2 were to be extended to the left of the junction it would continue above line 1; in the reflected version it would continue below line 1. The set of legal labellings of a reflected curvature-L junction is

$$\{\Rightarrow\rightarrow,\ \Rightarrow-,\ \leftarrow\Leftarrow,\ -\Leftarrow\}.$$

3. Complexity of the labelling problem

Kirousis and Papadimitriou [13] have shown that the labelling problem and also the realisability problem for drawings of polyhedral scenes are both NP-complete. The reason why the labelling problem becomes tractable for curved objects is the possible presence of undetected C junctions on any line in the drawing. This means that the difference between the labels $+$, $\rightarrow$ and $\leftarrow$ cannot be propagated along a line. Note that although C junctions are usually concave, they may also occur on straight or convex lines [4]. We introduce a new label δ to represent any of the set of labels $\{+, \rightarrow, \leftarrow\}$.

The same convex edge when viewed from different viewpoints corresponds to one of $\rightarrow$, $+$ or $\leftarrow$. The label δ gives the essential information about the structure of the edge (it is convex) without specifying the position of the viewpoint in relation to the two surfaces meeting at the edge.

For the purposes of finding a legal global labelling for a drawing, we can simplify the catalogue of Fig. 3 by replacing $+$, $\rightarrow$ and $\leftarrow$ by δ. This gives the catalogue of Fig. 4. Given a legal labelling for a drawing according to the catalogue of Fig. 4, it is a simple task to transform this into at least one legal labelling according to the catalogue of Fig. 3, given the freedom to place any number of C junctions on any line.

We remind the reader that reflected versions of curvature-L and 3-tangent junctions exist. For compactness these are not shown, but are nonetheless an essential part of the catalogue.

An important point is that C junctions can be eliminated from the catalogue. Lines cannot change labels between junctions with the reduced label set $\{\Leftarrow, \Rightarrow, \delta, -\}$. Thus the labelling problem consists in assigning a unique label to each line. This can be compared with the original labelling problem, according to the catalogue of Fig. 3, where the problem

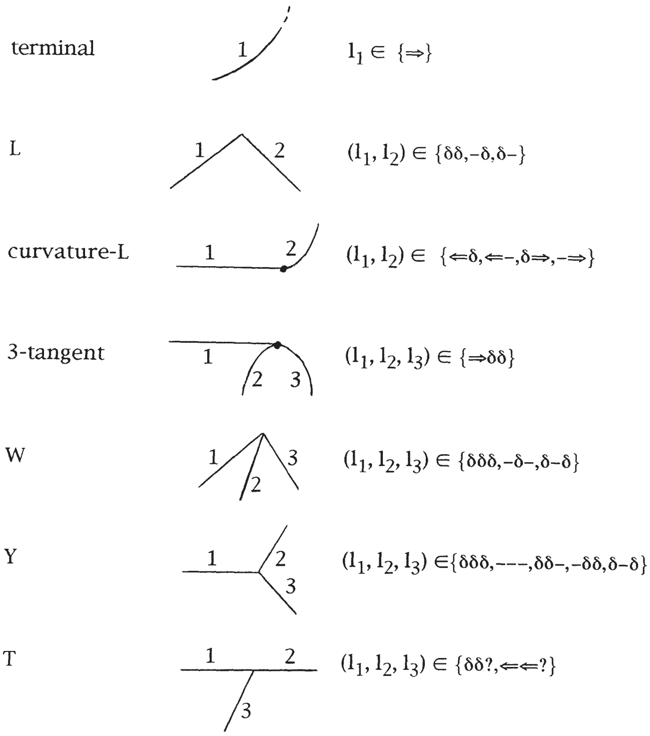


Fig. 4. Catalogue of junction labellings for objects with C^3 surfaces, in terms of the reduced label set $\{-, \delta, \Rightarrow, \Leftarrow\}$.

consisted in assigning a separate label either to each line-end (sparse version) or to every point of each line (dense version) [16].

We can now consider the labelling problem to be a constraint satisfaction problem (CSP) [24] in which the variables are the lines which must be labelled with one of the four labels ($\Leftarrow$, $\Rightarrow$, δ, $-$) and the constraints are the set of legal labellings for each junction.

Definition 3.1. A CSP is *pairwise consistent* if, for each pair of constraints which share a variable, each element of the first constraint can be extended to a labelling which satisfies both constraints.

Definition 3.2. If a total ordering exists on domains, then a constraint $C(P)$ is *max-closed* if for all pairs of tuples $(x_1, \ldots, x_r)$, $(y_1, \ldots, y_r) \in C(P)$, $(\max\{x_1, y_1\}, \ldots, \max\{x_r, y_r\}) \in C(P)$.

In the line drawing labelling problem we will impose an artificial total ordering on domains ("$\Rightarrow$" $<$ "$-$" $<$ "δ" $<$ "$\Leftarrow$"). A constraint satisfaction problem with max-closed constraints can be considered as a generalisation of HORNSAT to multi-valued logics. Pairwise consistency is a decision procedure for CSPs with max-closed constraints [11].

This result allows us to prove the following theorem.

Theorem 3.3. *Given a drawing of curved objects, composed of n lines, we can produce a legal global labelling according to the catalogue of Fig.* 4*, if it exists, or determine that no such labelling exists, in* $O(n)$ *time.*

Proof. We will prove the theorem by showing that it is possible to express the line drawing labelling problem as a CSP with max-closed constraints.

To distinguish between the two labels $\Rightarrow$ and $\Leftarrow$, we must assign a direction to each line in the drawing. The direction of each line can be arbitrary, so we choose them to be consistent for all lines in the same cycle or chain of curvature-L junctions: for example, clockwise for all lines in the same cycle. The result is that all curvature-L junctions J have one line entering and one line leaving J. Similarly we choose the directions of the two lines forming the bar of a T junction to be identical, so that one line enters and the other leaves the T junction.

Consider a line drawing labelling problem with constraints derived from the catalogue of Fig. 4. Due to the arbitrary choice of the directions assigned to the majority of lines in the drawing, the constraints which can occur in the labelling problem are those shown in Fig. 4 with the directions of any number of lines reversed. Reversing a line means interchanging the labels $\Rightarrow$ and $\Leftarrow$. The exceptions are curvature-L junctions and the bars of T junctions which, by our choice of directions assigned to lines, always have one line entering and one line leaving. The possible constraints are therefore:

$\{\Rightarrow\}$ or $\{\Leftarrow\}$	for terminal junctions
$\{\delta\delta, -\delta, \delta-\}$	for L junctions
$\{\Leftarrow\delta, \Leftarrow -, \delta \Rightarrow, - \Rightarrow\}$ or $\{\Rightarrow \delta, \Rightarrow -, \delta \Leftarrow, - \Leftarrow\}$	for curvature-L junctions
$\{\Rightarrow\delta\delta\}$ or $\{\Leftarrow\delta\delta\}$	for 3-tangent junctions

$\{\delta\delta\delta,\ -\delta-,\ \delta-\delta\}$ for W junctions
$\{\delta\delta\delta,\ ---,\ \delta\delta-,\ -\delta\delta,\ \delta-\delta\}$ for Y junctions
$\{\delta\delta\delta,\ \delta\delta-,\ \delta\delta\Rightarrow,\ \delta\delta\Leftarrow,\ \Leftarrow\Leftarrow\delta,\ \Leftarrow\Leftarrow-,\ \Leftarrow\Leftarrow\Rightarrow,\ \Leftarrow\Leftarrow\Leftarrow\}$
or $\{\delta\delta\delta,\ \delta\delta-,\ \delta\delta\Rightarrow,\ \delta\delta\Leftarrow,\ \Rightarrow\Rightarrow\delta,\ \Rightarrow\Rightarrow-,\ \Rightarrow\Rightarrow\Rightarrow,\ \Rightarrow\Rightarrow\Leftarrow\}$
for T junctions

We define the function $\max: L \times L \to L$, where L is the reduced set of line labels, according to the ordering

$$\text{“}\Rightarrow\text{”} < \text{“}-\text{”} < \text{“}\delta\text{”} < \text{“}\Leftarrow\text{”}.$$

It is easy to verify that all the constraints given above are max-closed under this ordering. For example, applying the function “max” pointwise to the two labellings $\delta\delta-$ and $-\delta\delta$ for a Y junction produces $\delta\delta\delta$, which is indeed a legal labelling for a Y junction.

It is known that pairwise consistency is a decision procedure for CSPs composed of max-closed constraints [11]. Furthermore, having established pairwise consistency, if no domain is empty then it is sufficient to select the maximum value in the domain of each variable to obtain a legal global labelling.

A drawing composed of n lines has $O(n)$ junctions. Pairwise consistency is a linear operation if we use a fast arc consistency algorithm in the dual problem [2,17]. □

Corollary 3.4. *Given a drawing composed of n lines which has N legal global labellings according to the catalogue of Fig.* 4*, we can output all N labellings in* $O(Nn^2)$ *time.*

Proof. The essential point to note is that we can determine the existence of a legal global labelling in $O(n)$ time even when some of the lines are restricted to have given fixed labels. Such restrictions are unary constraints and hence trivially max-closed (see Definition 3.2).

Imagine a backtrack search tree to find all legal global labellings of the drawing. At any node α of the search tree, some lines have their labels bound. We can determine in $O(n)$ time whether these bindings are part of at least one legal global labelling. By executing this test before the creation of each potential new node α, we can ensure that the search tree has no dead-ends.

The total number of nodes in the search tree without dead-ends is bounded above by Nn, since there are N paths from the root to a leaf, each of length n. The result follows from the fact that $O(n)$ work is required at each node to determine which son-nodes should be created. □

A useful constraint which has been employed by many workers is that the external boundary of the drawing is an occluding contour, consisting only of $\to$ and $\Rightarrow$ labels. This extra unary constraint does not destroy the max-closed property since all unary constraints are max-closed. Hence, Theorem 3.3 still holds.

Theorem 3.3 is a robust result, in the sense that it remains valid even after changes in the assumptions we made about object shape. For example, generalising the catalogue to include projections of apices of cones, to include projections of non-trihedral vertices or to include non-occlusion T junctions (see [5] or Appendix A for a definition) does not alter the validity of the above proof. Allowing discontinuities of surface curvature (smooth edges) produces another catalogue [4] whose constraints are max-closed (see Appendix A). The interpretation of line drawings of objects with possibly tangential edges and surfaces has also been shown to be solvable in linear time [5]. A similar result holds for pottery world objects [6].

These positive results are all due to the possible presence of C junctions on any line in the drawing. However, minimising the number of C junctions is an NP-hard problem (see Section 9), since labelling line drawings of polyhedra is a subproblem [13].

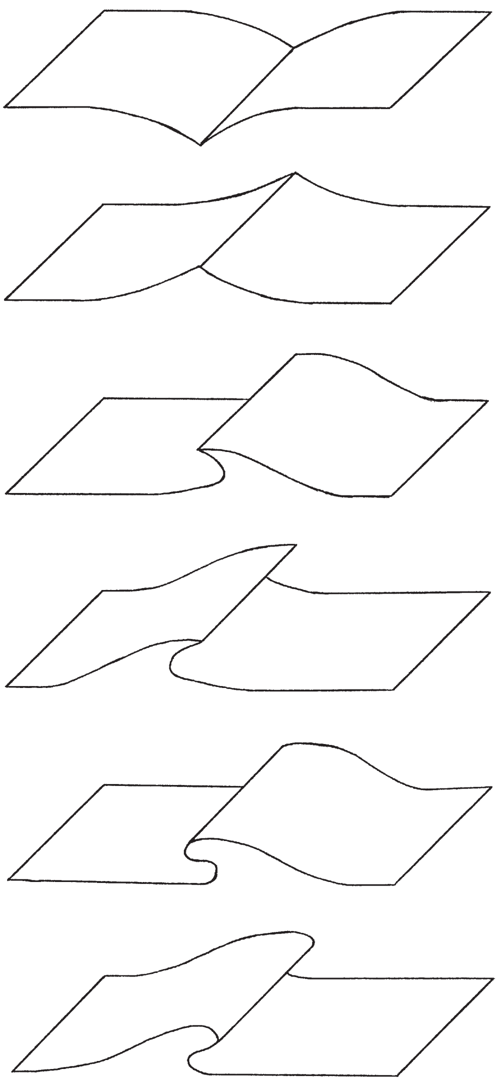


Fig. 5. The realisation of $-$, $+$, $\rightarrow$, $\leftarrow$, $\Rightarrow$, $\Leftarrow$ edges by local deformations of the rubber sheet in the vicinity of lines in the drawing.

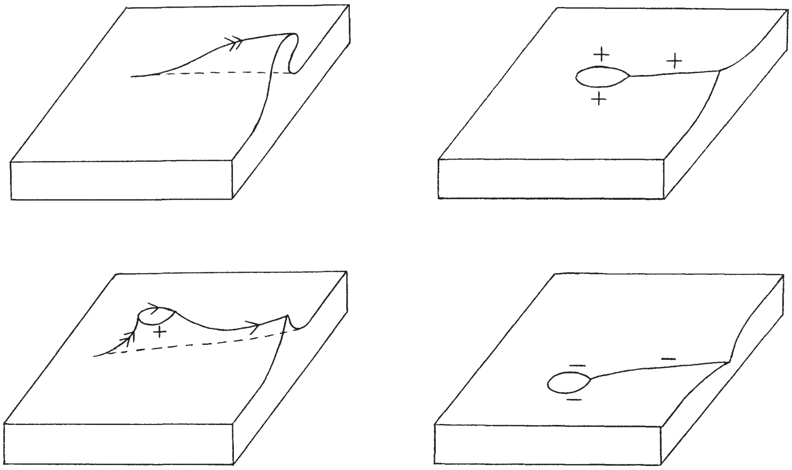


Fig. 6. Examples of local deformations in the rubber sheet in which ⇒, →, +, − edges disappear.

4. Physical realisability

The catalogue of legal junction labellings (Fig. 3) provides a necessary condition which must be satisfied by a global labelling of the drawing: each junction must be labelled by a labelling found in the catalogue. An obvious question is whether this condition is also sufficient. Are all labelled drawings which satisfy this condition physically realisable as a 3D scene? The result proved in this section answers "yes" to this question.

It is important to note that this result holds because of the freedom to choose arbitrary C^3 surfaces bounded by arbitrary C^3 edges.

Theorem 4.1. *Any legally-labelled drawing, according to the catalogue of Fig.* 3*, is physically realisable as a* 3D *scene.*

Proof. Given a labelled drawing we will construct a 3D scene which projects into this drawing.

Imagine a rubber sheet placed over the line drawing. We make small deformations in the rubber sheet along each line. These deformations create infinitesimally small edges with a shape which corresponds to the line label (−, +, →, ←, ⇒, ⇐). The deformations necessary to create +, −, → and ⇒ edges are shown in Fig. 5. The interior of each face in the drawing remains flat; deformations only occur in the neighbourhood of lines and junctions.

To complete the proof it is sufficient to give a construction of each type of labelled junction. Such constructions are fairly straightforward for Y, W and 3-tangent junctions. At other junctions it is necessary to make a hidden line disappear. Fig. 6 shows how it is possible to make ⇒, →, + and − lines disappear. Figs. 7 and 8 show a sample of the necessary constructions. Fig. 7 shows how to construct L junctions in the rubber sheet, Fig. 8(a) a C junction and Fig. 8(b) a curvature-L junction. None of the dangling edges are

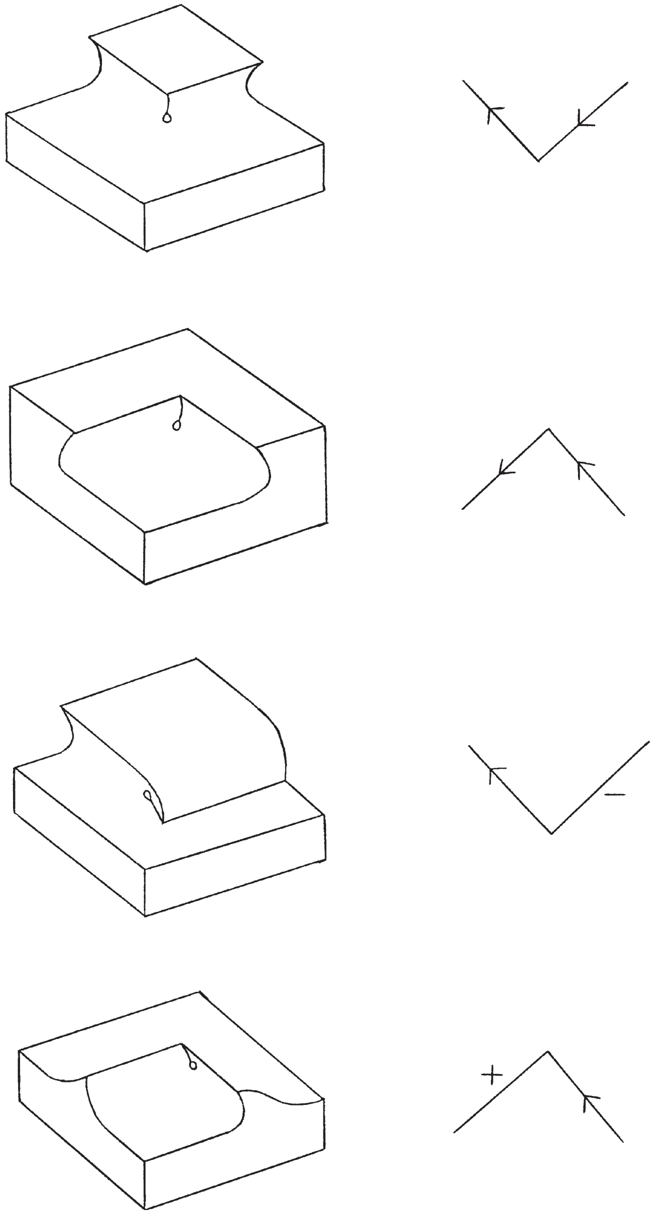


Fig. 7. Realisations of four L junction labellings by local deformations in the rubber sheet. The hidden line disappears by means of one of the constructions of Fig. 6.

actually visible in the drawing. Constructions for all other labelled junctions are analogous and even more straightforward. □

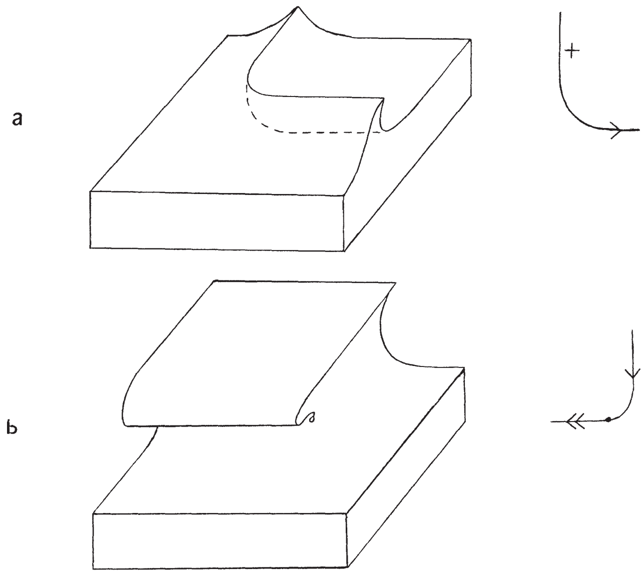


Fig. 8. Realisations of: (a) a C junction labelling; (b) a curvature-L labelling by local deformations in the rubber sheet.

5. Vanishing points

Parodi and Torre [19] extended the classic work on the labelling of line drawings of polyhedral scenes by tightening constraints, using knowledge of the positions of the vanishing points of all the lines in the drawing. The result is a linear-time labelling algorithm which generalises the linear-time algorithm of Kirousis and Papadimitriou [13] for legoland scenes. Apart from junction constraints, Parodi and Torre also made use of an L-chain constraint which is derived from the assumption of planar surfaces. However, since in this paper we refuse the assumption of planar surfaces, the L-chain constraint cannot be applied.

Even when objects may have curved surfaces, vanishing points exist and can provide tighter junction constraints. To illustrate this, Fig. 9 shows a single vanishing point P for a drawing of a non-polyhedral object. Parodi and Torre [19] derived tighter junction constraints from the positions of the vanishing points of all three lines meeting at a W or Y junction. These constraints apply equally well to line drawings of curved objects, since we can assume that, in the neighbourhood of a trihedral vertex V, the object can be approximated by a polyhedron. This approximating polyhedron is simply composed of the tangent planes to the three surfaces meeting at V. Indeed, under our assumptions of non-tangential C^3 surfaces, the three straight line edges formed by the intersection of these three planes are exactly the tangents to the three edges meeting at V.

This means that the vanishing points of tangents to curved lines at junctions can fulfill the same role as the vanishing points of straight lines. For example, in Fig. 9, *XP*, *YP* and *ZP* are tangents to curved lines in the drawing. From a practical point of view, it is clear that

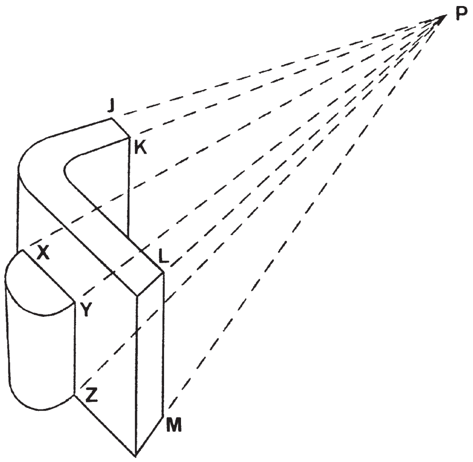


Fig. 9. Illustration of a vanishing point P in a drawing of a curved object. The tangents to line-ends at junctions J, K, L, M, X, Y, Z are all parallel in 3D space.

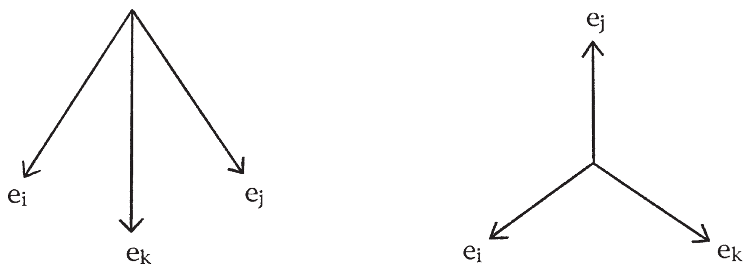


Fig. 10. The orientation vectors e_i, e_j, e_k of the three line-ends incident to a W or Y junction.

the directions of the tangents *XP*, *YP* and *ZP* will be determined with much less accuracy than the directions of the tangents *JP*, *KP*, *LP* and *MP*.

From an analysis of the positions of the vanishing points of the three lines meeting at a W junction it is possible to determine whether the middle line is convex or concave. If it is convex then the junction is known as a W(+) junction; if it is concave then the junction is known as a W(−). Similarly, it is possible to determine whether all of the lines of a Y junction are convex or whether at least one of them is concave. If all lines are convex then the junction is a Y(+), otherwise a Y(−).

We now give a simple and direct method for characterising W or Y junctions as + or −. Let f be the focal length of the imaging device. The orientation in space of the bundle of parallel lines whose projections meet at the vanishing point (x, y) is given by

$$e = \frac{(x, y, f)}{\sqrt{x^2 + y^2 + f^2}}.$$

Knowledge of all vanishing points thus allows us to determine the orientations of all three line-ends meeting at W or Y junctions: e_i, e_j, e_k (see Fig. 10). A W junction is + if and only if e_k lies in front of the plane of e_i and e_j, whereas a Y junction is + if and only if e_k lies behind the plane of e_i and e_j. In a left-handed coordinate system, the normal to the plane of e_i and e_j is given by $-(e_i \wedge e_j)$ for a W junction and by $e_i \wedge e_j$ for a Y junction. Therefore the characterisation of both W or Y junctions as + or − is given directly by the sign of

$$-(e_i \wedge e_j).e_k. \tag{1}$$

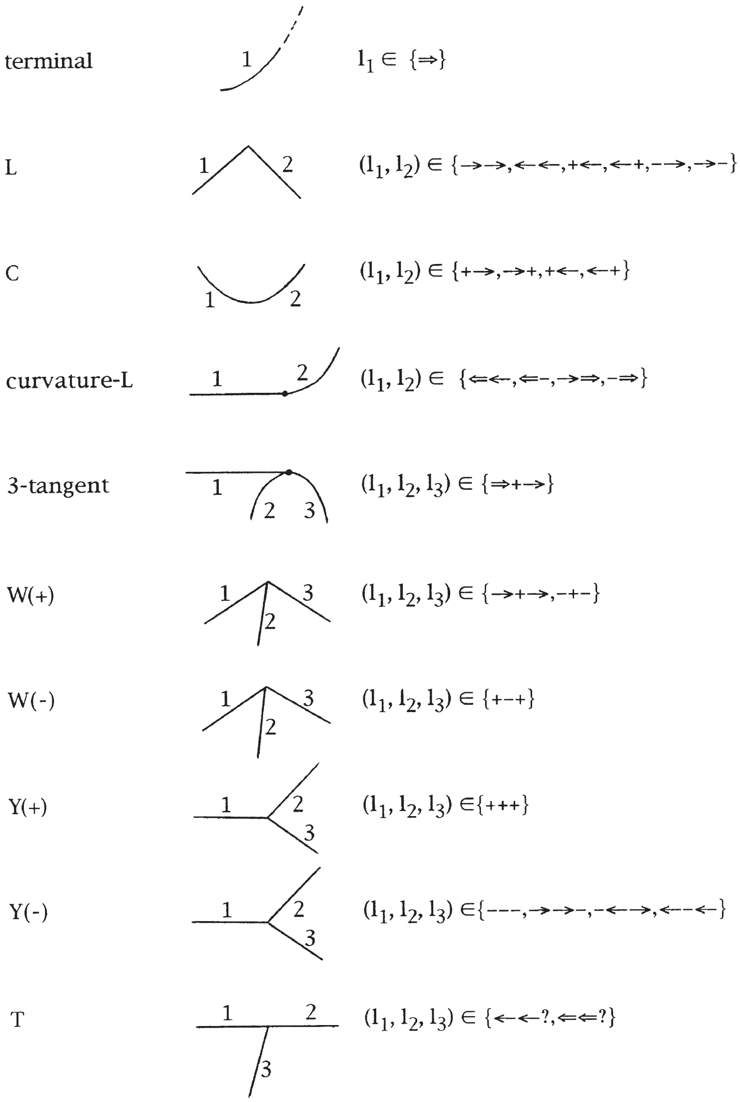


Fig. 11. Catalogue of junction labellings for objects with C^3 surfaces when the vanishing points of all line-ends are known.

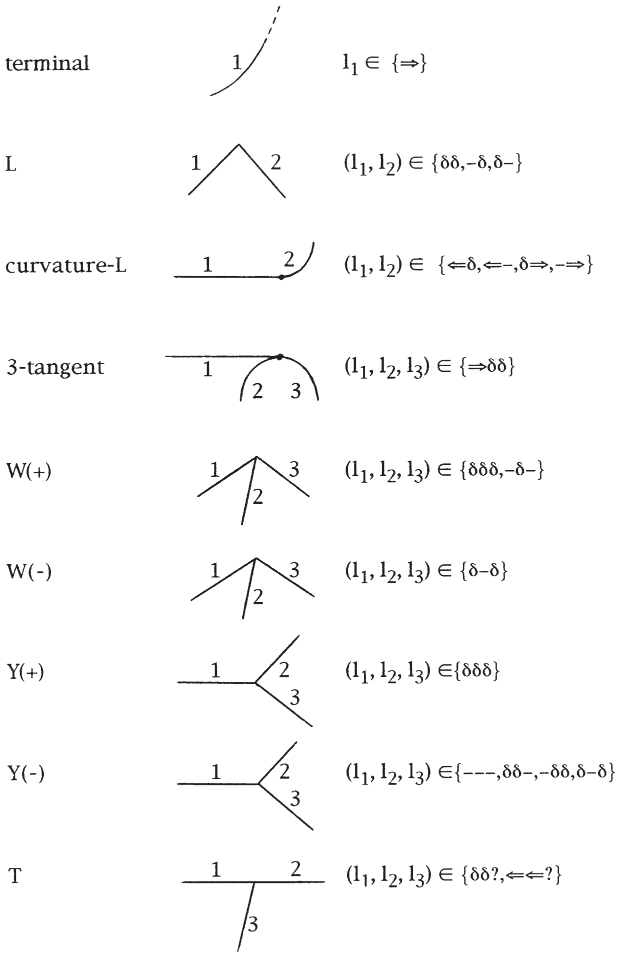


Fig. 12. The catalogue of Fig. 11 for the reduced label set $\{-, \delta, \Rightarrow, \Leftarrow\}$.

The characterisation of junctions as $+$ or $-$ is independent of the value of the focal length f (for $f > 0$) [19]. Thus knowledge of f is not essential, and we can simply set $f = 1$, for example, when f is unknown.

Fig. 11 shows the junction catalogue incorporating the classification of each W or Y junction as $+$ or $-$. Fig. 12 shows the catalogue of Fig. 11 for the reduced label set $\{-, \delta, \Rightarrow, \Leftarrow\}$ where δ represents any of the labels $+, \rightarrow, \leftarrow$.

6. Labelling constraints and backtrack-freeness

As in Section 3, we consider the labelling problem to be a constraint satisfaction problem (CSP) in which the variables are the lines which must be labelled by one of the four labels ($\Leftarrow, \Rightarrow, \delta, -$) and the constraints are the set of legal labellings for each junction.

It is easy to show that, after establishing pairwise consistency, the drawing can be partitioned into two types of connected components: those consisting exclusively of Y(−), Y(+), W(−), W(+) and L junctions and those consisting exclusively of terminal, curvature-L and 3-tangent junctions. The algorithm for labelling (terminal, curvature-L, 3-tangent) components is as for the case of line drawings without knowledge of vanishing points (Section 3). We can, thus, from now on consider drawings containing only Y, W, L and T junctions. Without knowledge of vanishing points, such drawings could always be uniformly labelled $(\delta, \delta, \ldots, \delta)$. This is no longer the case when Y(−) and W(−) junctions are present. Nevertheless, we will show that a linear-time labelling algorithm still exists.

Fig. 13 shows the initial transformations of Y(−), Y(+), W(+), W(−), and T junctions into constraints. The labels "−" and "δ" are coded as 0 and 1. This allows us to express the Y(−) constraint in a closed form as a binary linear equation. A Y(−) junction with

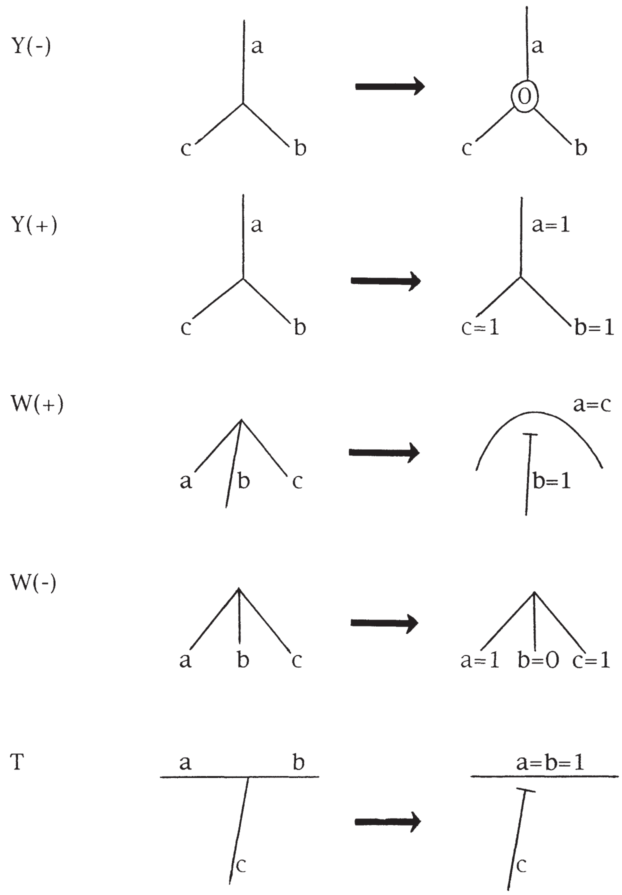


Fig. 13. The initial transformations of Y(−), Y(+), W(−), W(+) and T junctions into constraints, given the coding of labels "−" as 0 and "δ" as 1. The Y(−) junction is transformed into the binary linear equation constraint $a + b + c = 0$.

incident lines labelled a, b, c is transformed into the equation $a + b + c = 0 \pmod 2$. The constraints for L junctions are not shown in Fig. 13, since they are simply the list of legal labellings according to the catalogue of Fig. 12.

Definition 6.1. A *backtrack-free part* of a CSP is a set $B \subseteq V$, where V is the set of variables of the CSP, which satisfies the condition that any consistent labelling of $V - B$ can be extended to a consistent labelling of V.

We call *subtracting out* the act of eliminating the variables in a backtrack-free part. Subtracting out a backtrack-free part leaves a CSP on fewer variables which has the same satisfiability as the original CSP. It may involve updating those constraints whose scopes include one or more of the eliminated variables.

Definition 6.2. Two constraint satisfaction problems are *equivalent* if they have the same set of solutions.

Definition 6.3. A *backtrack-free reduction* is an operation which firstly transforms a CSP P on variables V into an equivalent CSP P' containing a backtrack-free part B and then subtracts out the variables in B to leave a CSP P'' on the variables $V - B$.

Definition 6.4. A *backtrack-free labelling rule* is a backtrack-free reduction together with a labelling algorithm. Given a consistent assignment α of values to all variables in $V - B$, this labelling algorithm extends α to a global consistent labelling of V.

A given backtrack-free reduction is an algorithm which identifies, and then eliminates by subtracting out, a certain class of backtrack-free parts. Such reductions can be applied when solving the problem of determining the existence of a global consistent labelling. A backtrack-free labelling rule not only eliminates a class of backtrack-free parts B, but also provides the algorithm to label B. It is therefore useful for solving the problem of finding a single global consistent labelling. Note that a backtrack-free labelling rule provides a single global consistent labelling; it is not a method for finding all global consistent labellings.

The backtrack-free labelling rules that we introduce below satisfy the following properties:

(1) The reduction operation as well as the labelling algorithm are linear algorithms.
(2) Every line drawing, with constraints derived from the catalogue of Fig. 12, can be reduced to an empty CSP by successive applications of this set of rules.

An important property of the following four backtrack-free labelling rules is that no new types of constraint need to be introduced when the variables in the backtrack-free part are subtracted out.

Rule 1. Two binary linear equations rule. $B = \{v\}$ where v is constrained by exactly two constraints both of which are linear equations over GF(2):

$$v + \sum_{i=1}^{s} u_i = c_1; \qquad v + \sum_{i=1}^{t} v_i = c_2.$$

The variable v is eliminated by replacing these two constraints by the single constraint

$$\sum_{i=1}^{s} u_i + \sum_{i=1}^{t} v_i = c_1 + c_2.$$

Labelling algorithm. Let $x_1, \ldots, x_s$ be the values assigned to $u_1, \ldots, u_s$. Assign to v the value

$$\sum_{i=1}^{s} x_i + c_1.$$

Rule 2. Dominating value rule. $B = \{v\}$, where there exists a value x in the domain $D(v)$ of v such that for all constraints $C(P)$ on sets $P = \{v, u_1, \ldots, u_r\}$ containing v,

$$\forall (y, y_1, \ldots, y_r) \in D(v) \times D(u_1) \times \cdots \times D(u_r)$$
$$\big((y, y_1, \ldots, y_r) \in C(P) \Rightarrow (x, y_1, \ldots, y_r) \in C(P)\big).$$

The variable v is eliminated by replacing each such constraint $C(P)$ by its projection onto the variables $P - \{v\}$:

$$C\big(P - \{v\}\big) = \prod_{P-\{v\}} C(P).$$

Labelling algorithm. Assign x to v.

Rule 3. Alternating boundary rule. $B = \{h_0, h_1, \ldots, h_{2r+1}\}$, for some $r \geqslant 0$, is a closed loop of lines forming the boundary of a face in the line drawing, such that L junction and binary linear equation constraints alternate. Fig. 14(c) illustrates such a loop.

All variables and constraints in the loop are eliminated.

Labelling algorithm. Let x_{ij} be the value assigned to v_{ij} (see Fig. 14(c)). Make the assignments

$$h_1 = h_3 = \cdots = h_{2r+1} = 1;$$
$$h_{2i} = c_i + \sum_j x_{ij} \quad (\text{for } i = 0, \ldots, r).$$

Rule 4. At most one constraint rule. $B = \{v\}$ where v is constrained by at most one constraint $C(P)$, where $P = \{v, u_1, \ldots, u_s\}$ with $s > 0$.

The variable v is eliminated by replacing $C(P)$ by its projection onto the variables $P - \{v\}$:

$$C\big(P - \{v\}\big) = \prod_{P-\{v\}} C(P).$$

Labelling algorithm. Let $x_1, \ldots, x_s$ be the values assigned to $u_1, \ldots, u_s$. Assign to v any value x such that $(x, x_1, \ldots, x_s) \in C(P)$.

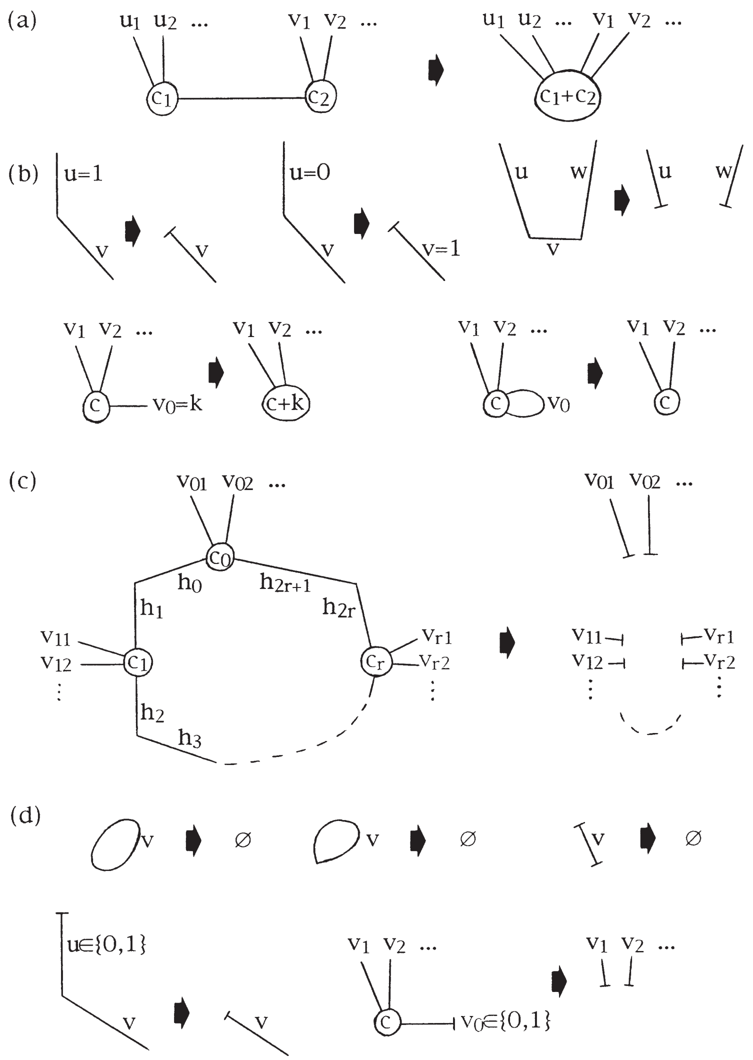


Fig. 14. The results of applying the backtrack-free reductions to the line drawing labelling problem: (a) Two binary linear equations rule; (b) Dominating value rule; (c) Alternating boundary rule; (d) At most one constraint rule.

We consider a constraint $C(\{u\})$ of order 1 to be a synonym for the domain $D(u)$. The creation of new constraints in Rules 1, 2 and 4, such as $C(P - \{v\})$, could lead to the creation of constraints $C(\{u\})$ of order 1. However, we assume that no order 1 constraints are actually created. Instead, the domain $D(u)$ is simply updated: $D(u) := D(u) \cap C(\{u\})$. If a domain becomes empty or an empty constraint is created, then the CSP has no solution.

Fig. 14 illustrates the results of applying these four backtrack-free reductions to the line drawing labelling problem. A small circle containing c represents a binary linear equation; the variables labelling the lines entering the circle sum to the value c. The only other constraints are node constraints (constraints of the form $v \in D(v)$) and L junction constraints. A line-end which is unconstrained, apart from node constraints, is represented by a small T junction.

Theorem 6.5. *Given a line drawing composed of L, Y and W junctions and for which all vanishing points of tangents to line-ends are known, coded as a CSP as illustrated in Fig.* 13*, the four backtrack-free labelling rules illustrated in Fig.* 14 *applied until convergence either demonstrate that no global consistent labelling exists or produce one.*

Proof. Suppose that the four rules have been applied until convergence without encountering a contradiction (an empty constraint). What form can the resulting line drawing have? The only possible constraints are node constraints, L junctions and binary linear equations. A binary linear equation must involve at least two variables, otherwise it is a node constraint. We can enumerate all possibilities for the pair of constraints at the two ends of a line:

(1) node constraint—node constraint,
(2) binary linear equation—binary linear equation,
(3) L junction—L junction,
(4) node constraint—binary linear equation,
(5) node constraint—L junction,
(6) binary linear equation—L junction.

We consider that case (1) also covers the case of a closed loop without any junction constraint.

It is impossible to encounter cases (1)–(5) in the final line drawing because the following rules would apply:

(1) At most one constraint rule.
(2) Two binary linear equations rule or Dominating value rule.
(3) Dominating value rule or At most one constraint rule.
(4) Dominating value rule or At most one constraint rule.
(5) Dominating value rule or At most one constraint rule.

This leaves alternating L junctions and binary linear equations (of at least two variables). Since all node constraints have been eliminated, the L junction and binary linear equation constraints must form at least one loop. But such loops are impossible by application of the alternating boundary rule.

We can conclude that all variables are eliminated by application of the four backtrack-free labelling rules. Successive application of the labelling algorithms corresponding to the backtrack-free labelling rules clearly produces a global consistent labelling. □

Theorem 6.6. *A line drawing for which vanishing points of all line-ends are known can be labelled* (*or the non-existence of a consistent labelling can be proved*) *in* $O(n)$ *time, where n is the number of lines in the drawing.*

Proof. By applying the rules in the right order we can guarantee convergence in a linear number of applications of the backtrack-free labelling rules. It is easily verified, by analysis of Fig. 14, that, by applying each individual rule until convergence in the order Rule 1, 2, 3, 4, no rule can trigger an earlier rule, and hence the final result is convergent for all rules. Fig. 14(b) illustrates a slightly weaker form of the Dominating value rule. Technically speaking it is this weaker form which cannot be triggered by application of later rules. It goes without saying that Theorem 6.5 still holds for this weaker version of the Dominating value rule.

The drawing can be identified with a graph whose vertices are the junctions and whose edges are the lines. Consider the subgraph G consisting just of Y(−) junctions. Applying the Two binary linear equations rule until convergence is equivalent to finding the connected components of G and, for each connected component G_i, finding the list of lines incident to exactly one junction in G_i. This can be achieved in $O(n)$ time. The Dominating value rule and the At most one constraint rule may lead to propagations, implying testing each rule at most twice for each line (once in an initialisation phase and once in a propagation phase). Efficient propagation techniques are standard in CSPs [2,17] and the propagation algorithm will not be detailed here. The Alternating boundary rule requires a single test for each face. The number of faces is clearly no greater than n, the number of lines in the drawing. Determining all faces can be achieved in $O(n)$ time. □

Corollary 6.7. *All N legal global labellings of a line drawing for which all the vanishing points of all line-ends are known can be output in $O(Nn^2)$ time.*

Proof. Identical to the proof of Corollary 3.4. □

We now consider the realisability problem for line drawings when all vanishing points are known. The position of a vanishing point determines the orientation in space of the corresponding bundle of straight lines. Consider a junction J in the drawing at which the three lines i, j, k meet. Let P_i, P_j, P_k be their respective vanishing points. It is an O(1) operation to determine the orientations in space e_i, e_j, e_k of the edges which project into lines i, j, k and hence to classify the junction as W(+), W(−), Y(+) or Y(−) (see Section 5). This classification is only impossible if the vanishing points P_i, P_j, P_k are collinear or if P_i, P_j, P_k are all points at infinity. This would indicate that the edge orientations e_i, e_j, e_k were coplanar, which would be in contradiction with our assumption of non-tangential C^3 surfaces and edges [19]. Otherwise a legal 3D vertex is constructible having the given labelling. It is easily verified that all legal labellings of terminal, curvature-L, L, 3-tangent and T junctions are also constructible even when the orientations of their edges are specified.

Having tested independently the constructibility of each junction in the drawings, we must now verify the possibility of putting together these constructions to form a legal 3D scene. In the absence of other constraints derived from, for example, the presence of straight lines or collinear points in the drawing, we can proceed as in the proof of Theorem 4.1.

Theorem 6.8. *The realisability of a line drawing of curved objects when all of the vanishing points of line-ends are known can be verified in $O(n)$ time.*

Proof. This theorem follows from Theorem 6.6 and the fact that the proof of Theorem 4.1 (the realisability theorem in the absence of knowledge of vanishing points) uses constructions that can easily be adapted to provide vertices whose edges have given orientations in 3D space. □

7. When not all vanishing points are known

Previous sections have considered the line drawing labelling problem when no vanishing points are known or when all vanishing points of tangents to line-ends are known. In practice, it is an unrealistic assumption to suppose that all vanishing points can be determined. If only some of the vanishing points are known, then the line drawing labelling problem may contain junctions from both the catalogue of Fig. 4 and the catalogue of Fig. 12. It turns out that this extra diversity of constraints produces a labelling problem which is NP-complete, even though both labelling problems corresponding to Figs. 4 and 12 are solvable in polynomial time. The constructions in the NP-completeness proof make use of L, W, Y($-$) and T junctions.

Theorem 7.1. *Labelling a line drawing of objects with C^3 surfaces when some of the vanishing points of tangents to line-ends are known is* NP*-complete.*

Proof. The problem is clearly in NP since the validity of a labelling can be checked in polynomial time. To complete the proof of NP-completeness it is sufficient to produce a polynomial transformation from a known NP-complete problem.

Lichtenstein [14] proved the NP-completeness of PLANAR 3SAT, a version of 3SAT in which the following bipartite graph G is planar: G has a vertex for each variable v, a vertex for each disjunction D and an edge for each pair (v, D) such that v or $\neg v$ is one of the three literals in D. In order to exhibit a polynomial reduction from PLANAR 3SAT to the line drawing labelling problem we need to specify the coding of variables, show how to generate many copies of the same variable, give a negation construction and give a construction for $u \vee v \vee w$.

We code "true" as "δ" and "false" as "$-$". In Figs. 15 and 16, all Y junctions are in fact Y($-$) junctions, but all W junctions are actual W junctions (and not W($+$) or W($-$)). Fig. 15 shows a construction to generate two copies y, z of the variable x. The only two legal labellings for this line drawing are shown. They correspond, respectively, to the assignments $x = y = z =$"$-$" and $x = y = z =$"δ". By chaining together $N - 1$ of these constructions we can generate N copies of the same variable x. Fig. 16(a) is a negation construction ($s = \neg r$). Since t must take on the value "δ", the only two legal assignments are ($r = -$; $s = \delta$) and ($r = \delta$; $s = -$). Fig. 16(b) is the construction for the disjunction of three literals u, v, w. It can easily be verified that all assignments of values to u, v, w are possible except $u = v = w =$"$-$". The construction thus imposes the condition $u \vee v \vee w$. □

The completely artificial nature of the constructions in the above NP-completeness proof leaves open the possibility of the existence of a polynomial-time heuristic which solves the

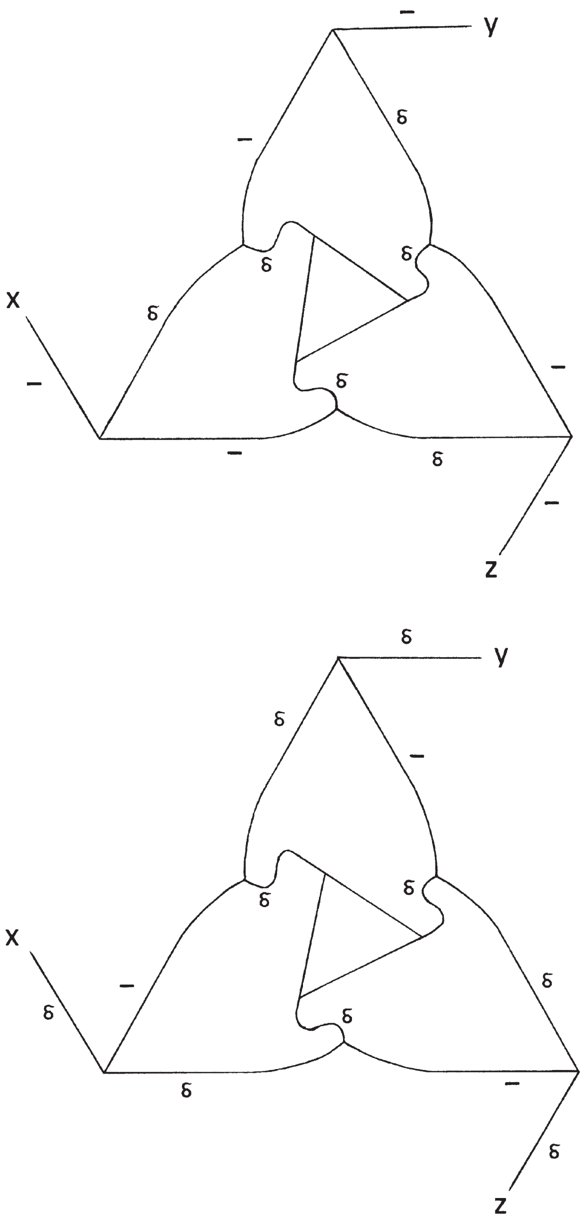
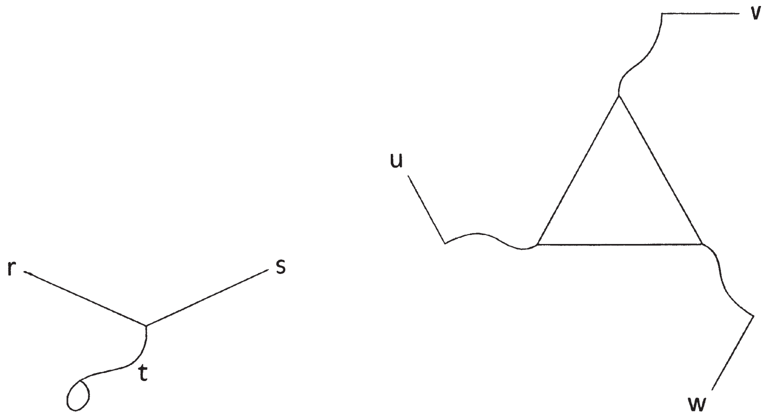


Fig. 15. The two legal labellings of the construction $x = y = z$ to make multiple copies of a variable.

labelling problem for almost all drawings which we are likely to encounter in practice. The four backtrack-free labelling rules given in Section 6 are clearly still applicable. However, in general, not all variables will be eliminated, due to the presence of W and Y junctions. In Appendix B we introduce another backtrack-free labelling rule to subtract out parts of



(a) (b)

Fig. 16. Constructions to simulate: (a) $r = \neg s$; (b) $u \vee v \vee w$.

the drawing which can be uniformly labelled “δ”. This *Uniform value rule* can be applied as follows. Let B be a part of the line drawing which contains only W, Y and L junctions. If the set of lines leaving B contains at most one line from each Y junction and only the middle line of any W junction, then B can be subtracted out with all its internal lines labelled “δ”. Limited experimental trials showed that the Uniform value rule together with the four backtrack-free labelling rules of Section 6 were very effective: for each drawing, either an inconsistency was detected or the line drawing was completely reduced by these five rules.

8. Parallel line-ends under orthographic projection

Under orthographic projection, vanishing points do not exist. Instead parallel lines in space project into parallel lines in the drawing D. Consider all the tangents to line-ends in the drawing D, and group them into bundles B_i of parallel lines. A bundle B_i may contain only a single line. The orientations in space e_i of the lines in bundle B_i can be considered as hidden variables which constrain the labellings of the line-ends in the drawing D.

For each Y or W junction J in D, the orientations of the three lines meeting at J determine the classification of J as $+$ or $-$ and hence its set of legal labellings (as explained in Section 5). Given the orientations of all bundles B_i, testing the realisability of the drawing D is equivalent to testing the existence of a legal global labelling according to the catalogue of labelled junctions shown in Fig. 12 (see proof of Theorem 6.8). We say that the set of orientations $\{e_i\}$ is consistent if the corresponding labelling problem has a solution. Testing the realisability of the drawing D is equivalent to testing the existence of a consistent set of orientations $\{e_i\}$ for the bundles of parallel lines in D.

We use the term “3-line junction” to denote either Y or W junctions. A pair of 3-line junctions J_1, J_2 are said to be parallel if each tangent to a line-end of J_1 is parallel to some tangent to a line-end in J_2. Consider the class of drawings D in which for each pair of

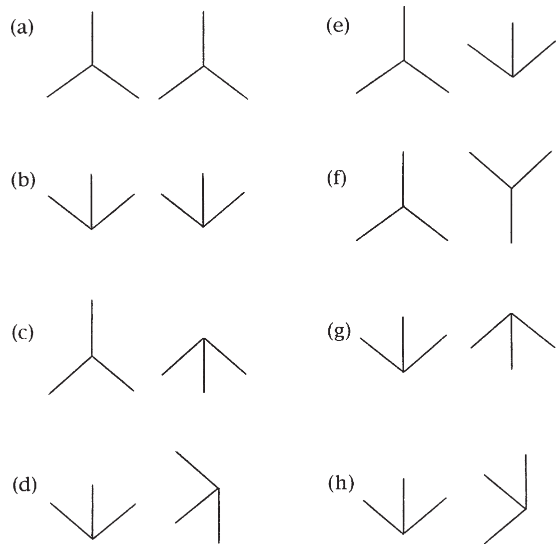


Fig. 17. Parallel junctions constraint: in cases (a)–(d) the junctions are of the same sign; in cases (e)–(h) they are of different sign.

3-line junctions J_1, J_2, either J_1, J_2 are parallel or none of the tangents to line-ends of J_1 are parallel to any of the tangents to line-ends of J_2. Under this simplifying assumption, the set of orientations $\{e_i\}$ can be partitioned into equivalence classes of size 1 or 3, two orientations e_i and e_j being in the same class if there exist line-ends corresponding to e_i and e_j which are incident to the same 3-line junction.

The decomposition of the set of vanishing points into independent subsets of size 1 or 3, implies that the only constraints that can be derived from parallel line-ends are constraints on the possible labellings of pairs of parallel 3-line junctions J_1, J_2. Indeed, there are no constraints on the classification as $+$ or $-$ of pairs of non-parallel junctions J_1, J_2. It is easy to verify, from Eq. (1) of Section 5, that the pairs of junctions shown in Figs. 17(a)–(d) must be classified as the same sign, whereas the pairs of junctions shown in Figs. 17(e)–(h) must be classified as different signs. For example, the two Y junctions in Fig. 17(a) must be either both Y(+) or both Y(−), since their line-ends have the same orientations. Similarly, Eq. (1) tells us that the Y and W junctions in Fig. 17(e) must be either Y(+) and W(−) or Y(−) and W(+).

Such constraints can easily be incorporated into the labelling problem by imposing an order 6 constraint on the line-ends of parallel 3-line junctions J_1 and J_2, this constraint being simply the list of all legal combinations of labellings for J_1 and J_2. As a concrete example, Fig. 18(a) shows two junctions J and K. The labelling $(\rightarrow, +, \rightarrow)$ for J identifies it as a W(+) junction. This implies that K is a Y(+) junction (case (c) of Fig. 17) which therefore must be labelled $(+, +, +)$. In this example, the labelling of K is uniquely determined by the labelling of J, but this is not always the case.

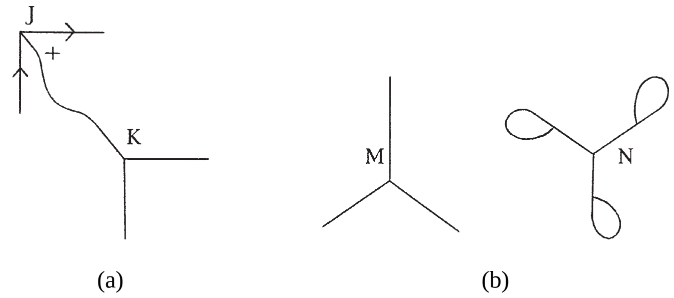


Fig. 18. (a) Example of the propagation of labels between parallel junctions; (b) construction of a Y(−) junction from constraints between parallel junctions.

Apart from the case when we limit the possible orientations of line-ends in the drawing to a very small number, there is no easy class of drawings under orthographic projection, equivalent to the case when all vanishing points are known.

Theorem 8.1. *Testing the existence of a legal labelling of a line drawing under orthographic projection of objects with C^3 surfaces when tangents to line-ends may be parallel is* NP-*complete.*

Proof. The proof uses the reduction from PLANAR 3SAT that was employed in the proof of Theorem 7.1. It is sufficient to show how we can constrain a Y junction to be a Y(−) junction, since the constructions of Figs. 15 and 16 use only Y(−), W, T and L junctions. Consider any Y junction M. Fig. 18(b) shows another junction N whose line-ends are all parallel to the line-ends in M, and whose labelling is (δ, δ, δ) in the reduced label set. This forces N to be a Y(+) junction, which in turn constrains M to be a Y(−) junction (case (f) of Fig. 17). □

An interesting question is whether an orthographic projection or a perspective projection of the same scene provides more information. We can consider that the classification of a 3-line junction as + or − provides 1 bit of information. Under perspective projection, each 3-line junction is thus worth 1 bit of information, provided that the vanishing points of all line-ends are known. Under orthographic projection, each 3-line junction J is worth 1 bit of information, provided that we have already seen another 3-line junction which is parallel to J. It is therefore possible to construct scenes whose orthographic projections are less ambiguous than their perspective projections and others whose perspective projections are less ambiguous than their orthographic projections.

9. Minimising the number of phantom junctions

We have given linear-time algorithms for finding a single labelling of a line drawing of curved objects, in the two cases when none or all of the vanishing points are known. The labelling uses the reduced label set $\{-, \delta, \Rightarrow, \Leftarrow\}$, but can easily be converted into a labelling in terms of the complete label set $\{-, +, \rightarrow, \leftarrow, \Rightarrow, \Leftarrow\}$ by the insertion of 0, 1 or 2 phantom junctions (C junctions) on each line labelled δ.

We call a labelling in terms of the complete label set a complete labelling. In the absence of other information it is clear that the most likely complete labellings are the ones which require the least number of phantom junctions. Unfortunately, it turns out that the problem of finding a complete labelling requiring the least number of phantom junctions is NP-hard. This is true in both cases: when none or all of the vanishing points are known. We call a complete labelling requiring no C junctions a phantom-free labelling.

Theorem 9.1. *Determining whether a line drawing of a C^3 scene has a legal phantom-free labelling is an* NP-*complete problem.*

Proof. The problem is in NP since the validity of a labelling and the absence of phantom junctions can clearly be verified in polynomial time. The proof is completed by noting that an algorithm to solve this problem could be used to solve the line drawing labelling problem for polyhedral scenes, a known NP-complete problem [13]. For details of this reduction, the reader is referred to a previous paper [5]. □

Theorem 9.2. *Determining whether a line drawing of a C^3 scene has a legal phantom-free labelling, consistent with the position of the vanishing point of the tangent to each line-end, is an* NP-*complete problem.*

Proof. Again the problem is clearly in NP. To prove NP-completeness we exhibit a polynomial reduction from PLANAR 3SAT.

Each variable v is transformed into a line which if labelled "$-$" indicates that $v = \text{false}$ and if labelled "$\rightarrow$" indicates that $v = \text{true}$. To generate N copies of the same variable v, we chain together $N - 1$ copies of the construction shown in Fig. 19(a). It can easily be verified that the labellings shown are the only two legal labellings of this construction, and hence that the construction simulates a = b = c. In Fig. 19 all Y junctions are assumed to have been classified as Y($-$) junctions and all W junctions as W($+$) junctions, after analysis of vanishing points.

It now suffices to give constructions simulating $\neg a$ and $a \vee b \vee c$. The construction in Fig. 19(b) has only two legal labellings corresponding to $(-, \rightarrow)$ and $(\rightarrow, -)$ and thus simulates negation $b = \neg a$. The construction in Fig. 19(c) has many labellings. It can easily be checked that all combinations of values for (a, b, c) are possible, except $(-, -, -)$. This construction therefore simulates $a \vee b \vee c$. □

Theorem 9.2 demonstrates the importance of a constraint known as the L-chain constraint in the labelling of perspective projections of polyhedral scenes when all vanishing points are known. Without this extra constraint, derived from the planarity of surfaces, the labelling problem is NP-complete, by the above proof. With the L-chain constraint the problem is solvable in polynomial time [19]. It is an open problem whether the physical realisability of drawings of polyhedral scenes is NP-complete or not when all vanishing points are known.

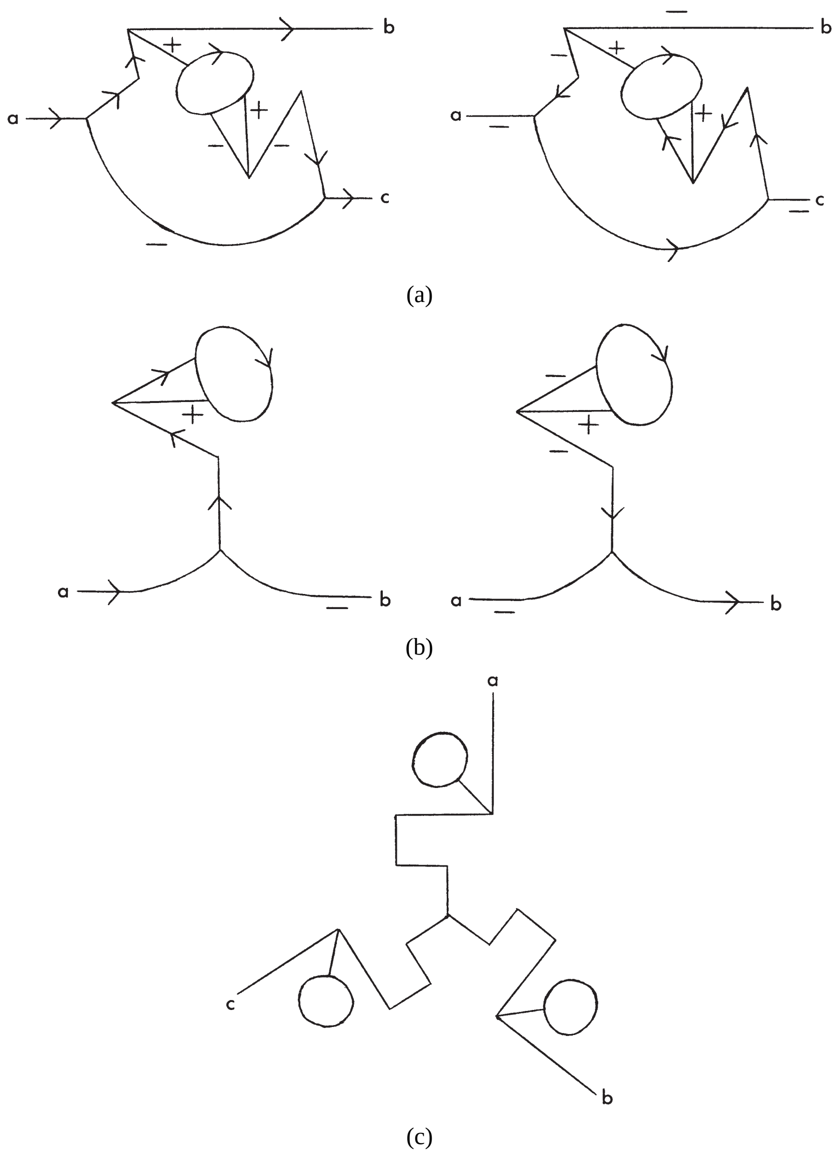


Fig. 19. Constructions to simulate: (a) $a = b = c$; (b) $a = \neg b$; (c) $a \vee b \vee c$.

10. Predictive power of the catalogues

To compare the utility of different catalogues, we can calculate the average number of bits of information per line-end that the catalogue provides when applied to a drawing. This quantity is also known as the predictive power (pp) of the catalogue. The calculation of pp is described in detail in a previous paper and will not be repeated here [5]. In order to

Table 1
Comparison of the quantity of information supplied by two different catalogues of labelled junctions

	pp	$\mathrm{pp}_{\max}/2$
C^3 surfaces	2.69	2.58
C^3 surfaces with knowledge of all vanishing points	2.89	2.58

calculate pp it is necessary to make arbitrary assumptions concerning the relative frequency of each junction type. For simplicity we always assume that each junction type in the catalogue occurs with equal frequency. Table 1 gives the values of pp for the catalogues of Figs. 3 and 11.

We say that a catalogue suffers from exponential weakness if the average number of legal interpretations of a drawing of n lines increases as an exponential function of n. For random line drawings, assuming independence between the set of legal labellings for each junction, the expected number of legal interpretations is

$$2^{2n((\mathrm{pp}_{\max}/2)-\mathrm{pp})},$$

where $\mathrm{pp}_{\max} = 2\log_2 6$ is the theoretical maximum value for pp [5]. The condition

$$\mathrm{pp} > \mathrm{pp}_{\max}/2$$

is therefore a necessary, although not sufficient, condition for a catalogue not to suffer from exponential weakness [5]. This condition is indeed satisfied by the two catalogues presented in this paper. We emphasise that this is not a sufficient condition for a catalogue to be free of exponential weakness. Besides, pp only tells us about the average case and not the worst case. Highly ambiguous drawings can easily be constructed even with knowledge of all vanishing points. For instance, if we consider the subclass of drawings containing only L and T junctions, then the value of pp falls well below the value $\mathrm{pp}_{\max}/2$, and both catalogues suffer from exponential weakness. As a concrete example of a drawing with an exponential number of legal labellings, consider an isolated chain of n L junctions. This drawing has $f(n+2)$ legal labellings, in the reduced label set $\{\delta, -\}$, where $f(n)$ is the nth Fibonacci number.

Other sources of information which can help to reduce ambiguity in line drawing interpretation include the occluding contour rule (see Section 3), local shape-from-shading analysis of a corresponding intensity image (to detect C junctions and extremal lines [16]) and information about hidden lines [9]. In the following section we study the possibility of incorporating collinearity constraints in order to reduce ambiguity.

11. Colinearity constraints

A drawing may contain sets of three or more junctions which are collinear. A tangent to a line-end may also pass through another junction. Both such examples of collinearity give rise to a constraint on the three-dimensional positions of the vertices of the scene since, by the general viewpoint assumption, collinear lines and points in the drawing must be the projections of collinear lines or points in space.

Let $V(j)$ represent the position in space of the vertex projecting into the junction j in the drawing. There are two types of constraints:

(1) If the three junctions i, j, k are collinear then $V(i)$, $V(j)$, $V(k)$ are collinear in space.

(2) If a line-end leaving junction i is collinear with junction j then its orientation is given by the orientation of the line joining $V(i)$ and $V(j)$.

These constraints provide a check on the realisability of the drawing. To give a concrete example, recall that the position of a vanishing point P determines the orientation in space of the bundle of straight lines whose projections converge to P. Given three straight lines forming a triangle in the drawing and two of the vanishing points of the lines, the third vanishing point is uniquely determined, because of the bijection between positions of vanishing points and orientations in space. This constraint can be viewed as a method for determining the third vanishing point or as a check on the realisability of the drawing when all vanishing points are known.

T junctions can give rise to a collinearity constraint among the orientations of line-ends and positions of vertices $V(i)$, but this time an inequality constraint, since the bar of the T must be in front of the stem of the T.

There is a different type of constraint linking the orientations in space of the three line-ends meeting at a Y or W junction and their semantic labels. This constraint was described in detail in Section 5 and illustrated by Figs. 10 and 11.

The interpretation of the drawing can thus be coded as a constraint satisfaction problem with three types of variables: semantic labels for line-ends, positions of vertices in space and orientations of line-ends in space. We study the complexity of this CSP in two cases: when all vanishing points are known, and when no vanishing points are known. We already know that it is NP-complete when some but not all vanishing points are known.

Under perspective projection, with knowledge of all vanishing points, there is independence between the labelling of line-ends and the determination of the positions of vertices in space, since the link between them, namely the orientations of line-ends in space, are fixed by knowledge of the vanishing points.

Number the junctions in the drawing from 1 to m. Let Z_i denote the Z-value of the vertex $V(i)$ projecting into junction i. Given the focal length f of the imaging device and the position (x_i, y_i) of junction i in the drawing, the value of Z_i completely determines the position of the corresponding vertex $V(i)$ in 3D space: $(x_i Z_i/f, y_i Z_i/f, Z_i)$. Note that we can assume the unary constraints

$$\forall i (Z_i \geqslant f).$$

We assume that the vanishing point is known for each line joining collinear points or line-ends in the drawing. In this case, the only non-unary constraints on the values of Z_i are either equality constraints or inequality constraints (derived from T junctions) of the form

$$Z_i = aZ_j \quad \text{or} \quad Z_i \geqslant aZ_j \tag{2}$$

where a is a positive constant. If (x_i, y_i), (x_j, y_j), $(x_{\mathrm{vp}}, y_{\mathrm{vp}})$ are the positions in the drawing of junction i, junction j and the vanishing point of the line passing through junctions i and j, then

$$a = (x_{\mathrm{vp}} - x_j)/(x_{\mathrm{vp}} - x_i) = (y_{\mathrm{vp}} - y_j)/(y_{\mathrm{vp}} - y_i) > 0.$$

The position of the vanishing point $(x_{\mathrm{vp}}, y_{\mathrm{vp}})$, and hence the value of a, will usually only be determined to within a certain error. This does not change the nature of the set of constraints (2), since we now have

$$(Z_j \geqslant (1/a_{\max})Z_i) \wedge (Z_i \geqslant a_{\min} Z_j) \quad \text{or} \quad Z_i \geqslant a_{\min} Z_j \tag{3}$$

if $a \in [a_{\min}, a_{\max}]$.

Such a system of constraints (3) can be solved using standard linear programming techniques.

We note also that these constraints are all binary and max-closed, and hence it is sufficient to establish arc consistency to find a solution [11]. Although arc consistency is not computable over infinite domains, we can fix an upper bound on the values Z_i and quantize the domain of their possible values to render the domains finite. Let D be the domain size and c the number of collinearity constraints. Then we can determine whether the set of constraints (3) has a solution, and return one if it exists, in $\mathrm{O}(D^2 c)$ time using an optimal arc consistency algorithm [2].

The following theorem is a consequence of the independence of the labelling problem and the determination of the values of Z_i, when all vanishing points are known. We already know, from Theorem 6.6, that the labelling problem is solvable in $\mathrm{O}(n)$ time, where n is the number of lines in the drawing.

Theorem 11.1. *Testing the realisability of a drawing of curved objects, which may contain collinear line-ends or points, when the vanishing points of all tangents to line-ends and of all lines joining collinear junctions are known, is solvable in* $\mathrm{O}(D^2 c + n)$ *time, where* c *is the number of collinearity constraints and* D *the domain size for* z*-coordinates.*

Proof. It follows immediately from the above discussion and Theorem 6.6. All the constructions used in the proof of Theorem 4.1 exist even when the orientations in space of the visible lines meeting at a vertex and the position in space of the vertex are all specified. □

The following theorem concerns the tractability of testing the realisability of drawings of three-dimensional scenes containing collinear points and lines when no vanishing points are known.

Theorem 11.2. *Testing the realisability of a drawing of curved objects which may contain collinear line-ends or points is* NP*-complete.*

Proof. To be able to use the same reduction from PLANAR 3SAT as in Theorem 7.1, it is sufficient to show how to impose the restriction that a Y junction be a Y(−) junction, using only collinearity constraints. Fig. 20 shows a 4-junction construction which constrains

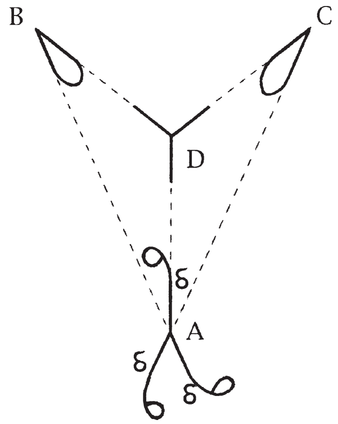


Fig. 20. Construction of a Y(−) junction in the presence of collinearity constraints.

junction D to be a Y(−) junction. The unique labelling (δ, δ, δ), in the reduced label set, for the Y junction A identifies A as a Y(+) junction.

Let e_{BA} represent the unit vector of the line in space which projects into BA (and similarly for other lines). We know from the characterisation of Y(+) junctions given in Section 5 that

$$-(e_{BA} \wedge e_{CA}).e_{AD} > 0.$$

The sign of the Y junction D is determined by the sign of

$$-(e_{DB} \wedge e_{DC}).e_{DA}.$$

But (for some positive scalar constants a, b, c, d)

$$\begin{aligned} -(e_{DB} \wedge e_{DC}).e_{DA} &= -\big((ae_{DA} + be_{AB}) \wedge (ce_{DA} + de_{AC})\big).e_{DA} \\ &= -(bde_{AB} \wedge e_{AC}).e_{DA} = bd(e_{BA} \wedge e_{CA}).e_{AD} < 0 \end{aligned}$$

and hence D is a Y(−) junction. □

In the presence of collinearity constraints, we have the same dichotomy as for the labelling of drawings of polyhedral scenes: the general problem is NP-complete [13], but becomes solvable in polynomial time when all vanishing points are known [19].

A strategic point is that the construction in the proof of Theorem 11.2 shows that collinearity constraints can determine the classification of a 3-line junction J as + or −, without actually determining the orientation in space of the three lines whose projections meet at J. It is possible to write down a constraint on all pairs of junctions A, D with collinear line-ends and such that the tangents to other line-ends meet at some junctions B and C (as in Fig. 20). In each case, the sign of A determines the sign of D. This is similar to the constraint between parallel 3-line junctions under orthographic projection (see Section 8). Such constraints provide necessary but not sufficient conditions for the drawing to be realisable. They should be considered as a way of rendering the constraints derivable from parallel lines and collinearity more explicit and more directly usable.

12. Discussion

As a concrete example, consider the line drawing shown in Fig. 9. Applying the catalogue of Fig. 3, and establishing pairwise consistency binds 12 out of 38 line-end labels to a unique value. The catalogue of Fig. 11, which uses information about all vanishing points, binds 22 out of 38 labels when applied to the same drawing. Minimising the number of phantom junctions has a similar effect, since 26 out of 38 labels are bound. Simultaneously using vanishing points and minimising the number of phantom junctions binds all 38 labels. An identical result (a unique label for each of the 38 line-ends) is obtained for this drawing by applying the occluding contour rule alone with the basic catalogue of Fig. 3.

We recommend using the occluding contour rule and information from vanishing points first, before embarking on any possibly combinatorial search, such as branch and bound, to minimise the number of phantom junctions. The linear-time algorithms, described in Sections 3 and 6, to determine the existence of a legal global labelling may be employed either during the search, to prune the search tree, or as a preprocessing step. For example, to establish global consistency (a state in which each element of each domain can be extended to at least one global consistent labelling), we can determine for each possible assignment of a label to a line whether this assignment can be extended to a global consistent labelling. This global consistency algorithm has time complexity $O(n^2)$.

13. Conclusion

This paper has analysed the problem of the interpretation of line drawings of scenes composed of curved objects with piecewise C^3 surfaces. A previously published catalogue of junction labellings has been shown to provide a necessary and sufficient condition for the physical realisability of a line drawing. Furthermore, the labelling problem can be solved in linear time. A linear-time test for physical realisability has also been given for the case in which the vanishing points of all line-ends are known.

Several intractability results show that these results are in some sense the best that we can do. Labelling a drawing when some but not all vanishing points are known or in the presence of parallel lines under orthographic projection are NP-complete problems. Minimising the number of phantom junctions (with or without knowledge of vanishing points) is NP-hard.

In the presence of collinear lines and points, the realisability problem is solvable in polynomial time when all vanishing points are known, but NP-complete otherwise.

An interesting avenue for future research is the search for other backtrack-free labelling rules and their application in the study of the tractability of other constraint satisfaction problems.

Appendix A

A different class of objects, in which discontinuities of surface curvature (smooth edges) are allowed, is studied in another paper [4]. A new labelling (δ, δ) is possible for curvature-L junctions due to the presence of a hidden smooth edge. The new curvature-L constraint

$$\{\Leftarrow\delta, \Leftarrow -, \delta\delta, \delta \Rightarrow, - \Rightarrow\}$$

is still max-closed under the above ordering, as are all other constraints derived from the catalogue of junction labellings when smooth edges are allowed [4]. It is easy to adapt the $O(n)$ labelling algorithm of Section 3 to this new catalogue.

Another possible extension of our class of objects is to allow non-occlusion T junctions. These junctions are the projections of the vertices formed when, for example, two planks of wood are nailed together to form a single object having the shape of a cross. Four such junctions occur at the join of the two planks. Their projections in the drawing are T junctions which are not caused by occlusion. The set of legal labelings for T junctions becomes

$$\{\delta\delta\delta, \delta\delta-, \delta\delta\Rightarrow, \delta\delta\Leftarrow, \Leftarrow\Leftarrow\delta, \Leftarrow\Leftarrow -, \Leftarrow\Leftarrow\Rightarrow, \Leftarrow\Leftarrow\Leftarrow, -\delta\delta, \delta-\delta\}.$$

All constraints derived from this set of labels are still max-closed under the ordering given in Section 3.

Appendix B. Uniform value rule

B is a subset of variables such that, for some value k, all constraints $C(P)$ such that $P \cap B \neq \emptyset$ satisfy the following *uniformity property*:

if $P = \{i_1, \ldots, i_p\}$ where $i_1, \ldots, i_r \in B$ then
$\forall (x_{r+1}, x_{r+2}, \ldots, x_p) \in A_{r+1} \times A_{r+2} \times \cdots \times A_p$
$(k, k, \ldots, k, x_{r+1}, x_{r+2}, \ldots, x_p) \in C(P)$.

In particular, if $P \subseteq B$ then $(k, \ldots, k) \in C(P)$. When $p = 1$, $C(P)$ is the domain of the variable i_1.

Labelling algorithm. Assign the value k to all variables in B.

In the context of the line drawing labelling problem, we only apply this rule with $k = \delta$. For example, in the case that no vanishing points are known, the Uniform value rule can be used to label the part of the drawing not containing terminal, curvature-L or 3-tangent junctions: all lines are labelled δ.

This rule can be applied in linear time to any CSP, for a given value of k. To show this we need the following lemmas.

Lemma B.1. *If all domains contain k and B satisfies the Uniform value rule for k, then $(k, \ldots, k) \notin C(P)$ implies that $P \cap B = \emptyset$.*

Lemma B.2. *If all domains contain k and $C(P)$ satisfies the uniformity property for B, but not for $B' \supseteq B$, then $P \cap B = \emptyset$.*

Proof. Suppose for a contradiction that $P \cap B \neq \emptyset$. Let $P = \{i_1, \ldots, i_r, i_{r+1}, \ldots, i_s, i_{s+1}, \ldots, i_p\}$ where $P \cap B = \{i_1, \ldots, i_r\} \neq \emptyset$ and $P \cap B' = \{i_1, \ldots, i_s\}$. Since $C(P)$ does not satisfy the uniformity property for B', there exists a tuple $(x_{s+1}, \ldots, x_p)$ such that $(k, \ldots, k, x_{s+1}, \ldots, x_p) \notin C(P)$. This implies that there exists a tuple $(y_{r+1}, \ldots, y_p) = (k, \ldots, k, x_{s+1}, \ldots, x_p)$ such that $(k, \ldots, k, y_{r+1}, \ldots, y_p) \notin C(P)$, which in turn implies that $C(P)$ does not satisfy the uniformity property for B. □

We can find a set B satisfying the uniform value rule for a value k by starting with $B = V$, the set of all variables in the CSP, and eliminating variables from B until the Uniform value rule is satisfied or $B = \emptyset$. Lemma B.1 tells us that we must eliminate from B all variables in the scope of constraints $C(P)$ such that $(k, \ldots, k) \notin C(P)$. Lemma B.2 tells us that we must eliminate from B all variables in the scope of constraints $C(P)$ such that $C(P)$ does not satisfy the uniformity property for B. It is clear that when the algorithm UVR-1, below, terminates either $B = \emptyset$ or B satisfies the Uniform value rule for k. Lemmas B.1 and B.2 tell us that B is maximal, in the sense that the only other sets satisfying the Uniform value rule for k are subsets of B.

Algorithm UVR-1.

```
B := V; {V = set of all variables of the CSP}
for all constraints C(P)
    if (k, ..., k) ∉ C(P) then B := B − P;
repeat
    for all constraints C(P)
        if P ∩ B ≠ ∅ and
            C(P) does not satisfy uniformity property for B and k
        then B := B − P;
until no eliminations from B in an iteration;
```

The following algorithm, UVR-2, is an optimised version of UVR-1 which only retests the uniformity property for $C(P)$ and B if $P \cap B$ has changed.

Algorithm UVR-2.

```
B := V;    Del := ∅;
for all constraints C(P)
    if (k, ..., k) ∉ C(P) then
    begin Del := Del ∪ (P ∩ B);
        B := B − P;
    end;
while Del ≠ ∅ do
begin Select and delete any variable v from Del;
    for all constraints C(P) such that v ∈ P do
        if C(P) does not satisfy uniformity property for B and k
        then begin Del := Del ∪ (P ∩ B);
                    B := B − P;
                end;
end;
```

Each variable v is added and hence deleted from *Del* at most once. Therefore the number of iterations of the while loop is $O(n)$. Suppose that each variable is in at most c_0 constraints, where c_0 is a constant, and that the uniformity property can be verified in O(1) time. (This is the case in the line drawing labelling problem.) Then the time and space complexity of UVR-2 is $O(n)$. Storing the set B as a boolean array of length n, such that $B[v] = \text{true}$ iff $v \in B$, allows us to implement the operations

$$Del := Del \cup (P \cap B);$$
$$B := B - P;$$

in $O(p)$ time where p is the cardinality of P. These operations will be executed at most once for each constraint $C(P)$. The linear complexity of UVR-2 follows from the fact that each variable occurs in at most c_0 constraints, where c_0 is a constant.

References

[1] G.A. Berkeley, New Theory of Vision in Philosophical Works (1709), Dent, Rowman and Littlefield, London, 1975.
[2] C. Bessière, Arc-consistency and arc-consistency again, Artificial Intelligence 65 (1994) 179–190.
[3] M.B. Clowes, On seeing things, Artificial Intelligence 2 (1971) 79–116.
[4] M.C. Cooper, Interpretation of line drawings of complex objects, Image and Vision Computing 11 (2) (1993) 82–90.
[5] M.C. Cooper, Interpreting line drawings of curved objects with tangential edges and surfaces, Image and Vision Computing 15 (1997) 263–276.
[6] N.D. Dendris, I.A. Kalafatis, L.M. Kirousis, An efficient parallel algorithm for geometrically characterising drawings of a class of 3-D objects, J. Math. Imaging and Vision 4 (1994) 375–387.
[7] G. Falk, Interpretation of imperfect line data as a three-dimensional scene, Artificial Intelligence 3 (1972) 101–144.
[8] A. Guzman, Decomposition of a visual scene into three-dimensional bodies, in: Proc. Fall Joint Comput. Conf., 1968, pp. 291–304.
[9] D.A. Huffman, Impossible objects as nonsense sentences, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence, Vol. 6, Edinburgh University Press, Edinburgh, 1971, pp. 295–323.
[10] P. Janssen, P. Jégou, B. Nougier, M.-C. Vilarem, A filtering process for general constraint satisfaction problems: achieving pair-wise consistency using an associated binary representation, in: Proceedings IEEE Workshop on Tools for Artificial Intelligence, 1989, pp. 420–427.
[11] P.G. Jeavons, M.C. Cooper, Tractable constraints on ordered domains, Artificial Intelligence 79 (1995) 327–339.
[12] T. Kanade, Recovery of the three-dimensional shape of an object from a single view, Artificial Intelligence 17 (1981) 409–460.
[13] L.M. Kirousis, C.H. Papadimitriou, The complexity of recognizing polyhedral scenes, J. Comput. System Sci. 37 (1) (1988) 14–38.
[14] D. Lichtenstein, Planar formulae and their uses, SIAM J. Comput. 11 (1982) 329–343.
[15] J. Malik, Interpreting line drawings of curved objects, Internat. J. Computer Vision 1 (1987) 73–103.
[16] J. Malik, D. Maydan, Recovering three-dimensional shape from a single image of curved objects, IEEE Trans. Pattern Anal. Machine Intelligence 11 (6) (1989) 555–566.
[17] R. Mohr, T.C. Henderson, Arc and path consistency revisited, Artificial Intelligence 28 (1986) 225–233.
[18] V.S. Nalwa, Line drawing interpretation: a mathematical framework, Internat. J. Computer Vision 2 (2) (1988).
[19] P. Parodi, V. Torre, On the complexity of labeling perspective projections of polyhedral scenes, Artificial Intelligence 70 (1994) 239–276.

[20] R. Shapira, H. Freeman, Computer description of bodies bounded by quadric surfaces from a set of imperfect projections, IEEE Trans. Comput. 27 (9) (1978) 841–854.
[21] R. Shapira, H. Freeman, The cyclic order property of vertices as an aid in scene analysis, Comm. ACM 22 (6) (1979) 368–375.
[22] K. Sugihara, A necessary and sufficient condition for a picture to represent a polyhedral scene, IEEE Trans. Pattern Anal. and Machine Intelligence 6 (5) (1984) 578–586.
[23] K. Sugihara, Machine Interpretation of Line Drawings, MIT Press, Cambridge, MA, 1986.
[24] E. Tsang, Foundations of Constraint Satisfaction, Academic Press, New York, 1993.
[25] D. Waltz, Understanding line drawings of scenes with shadows, in: P.H. Winston (Ed.), Psychology of Computer Vision, McGraw-Hill, New York, 1975, pp. 19–91.