ELSEVIER

Artificial Intelligence 119 (2000) 235–258

Artificial Intelligence

www.elsevier.com/locate/artint

Linear constraints for the interpretation of line drawings of curved objects

Martin C. Cooper [1]

Université de Toulouse III, IRIT, 118, route de Narbonne, 31062 Toulouse, France

Received 28 May 1999

Abstract

Drawings of curved objects often contain many linear features: straight lines, colinear or coplanar points, parallel lines and vanishing points. These linear features give rise to linear constraints on the 3D position of scene points. The resulting problem can be solved by standard linear programming techniques. An important characteristic of this approach is that instead of making a strong assumption, such as all surfaces are planar, only a very weak assumption, which disallows coincidences and highly improbable objects, needs to be made to be able to deduce planarity.

The linear constraints, combined with junction-labelling constraints, are a powerful means of discriminating between possible and impossible line drawings. They provide an important tool for the machine reconstruction of a 3D scene from a human-entered line drawing. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Line drawing labelling; Coplanarity constraints; Colinear points; Parallel lines; Vanishing points; Linear programming

1. Discriminating between possible and impossible drawings

Human vision combines evidence from many sources in order to choose a unique interpretation for a line drawing. Information extracted from straight lines, colinear points, parallel lines and junctions in the drawing is rapidly integrated to reduce the inherent ambiguity due to the loss of one dimension when a 3D scene is projected into a 2D drawing. The speed of human visual interpretation of line drawings is no doubt partly due to the abundance of such linear constraints in most drawings encountered in practice.

Pioneering work on machine interpretation of line drawings concentrated almost exclusively on assigning semantic labels to lines in drawings of polyhedral scenes [1,9,10].

[1] Email: cooper@irit.fr.

PII: S0004-3702(00)00008-4

Sugihara [20,21] was able to state necessary and sufficient conditions for a drawing to be the projection of a polyhedral scene, by using not only these semantic labelling constraints but also constraints derived from the assumption that all faces were planar. He expressed these planarity constraints as linear equations between variables corresponding to the parameters of the object faces. Constraints derived from a given semantic labelling of the drawing were expressed as linear inequalities. The result was a standard linear programming problem.

Unfortunately, the restriction to planar-faced objects means that Sugihara's work cannot be directly applied to line drawings of curved objects. Nevertheless, the spirit of his work can be retained whenever linear constraints, such as colinearity or coplanarity, are applicable. Many drawings of curved objects contain straight lines, colinear points, parallel lines or coplanar points which are, in fact, the key to the interpretation of the drawing. Given an arbitrary 2D curve there is an infinite family of 3D curves which project into it, whereas a straight line in the drawing can, barring coincidences, be assumed to be the projection of a straight line in 3D.

In this paper, a drawing consists of a set of junctions linked by a set of possibly curved lines. Unlike Sugihara, who used the assumption of polyhedral objects, object faces can be of any C^3 shape. The aim will, therefore, not be to determine the 3D equations of faces, but simply to determine the positions in 3D space of each vertex projecting into a visible junction in the drawing. The following sections give the mathematical derivation of linear constraints both in the case of orthographic and perspective projection. This section will simply demonstrate the power of linear constraints and semantic labels to distinguish between possible and impossible drawings.

Drawings are assumed to be perfect projections, from a general viewpoint, of scenes containing objects composed of opaque C^3 surfaces separated by C^3 surface-normal discontinuity edges meeting non-tangentially at vertices. Straight edges are assumed to be formed by the intersection of locally planar surfaces. This means that objects behave locally as polyhedra in the vicinity of vertices and straight edges. However, no extra restriction needs to be imposed, such as trihedral vertices [1,9] or the cyclic-order property [16,17].

The propagation of semantic labels for lines is sufficient to identify certain well-known examples of drawings as impossible. For example, Fig. 1(a) is clearly a drawing of an impossible object and this can easily be detected by the fact that, for example, the line *AB* transforms itself from an occluding edge at A to an extremal edge at B. Such a label transition is illegal under the assumptions about object shape and image formation stated above. *Occluding* edges are formed by the intersection of two surfaces, only one of which is visible, and are denoted by an arrow. *Extremal* edges are formed by a single curved surface being tangential to the line of sight, and are denoted by a double-headed arrow. For example, the contour of a cube is made up of occluding edges, whereas the contour of a sphere is an extremal edge.

Fig. 1(b) shows a more subtle example of an impossible object, under the common assumption that all object vertices are trihedral. The line *AB* contains a label transition from a convex edge at A to an occluding edge at B. A *convex* edge is formed by the intersection of two surfaces (both of which are visible) at an angle of less than π within the object, and is denoted by "+". If the angle is greater than π, then the edge is *concave*

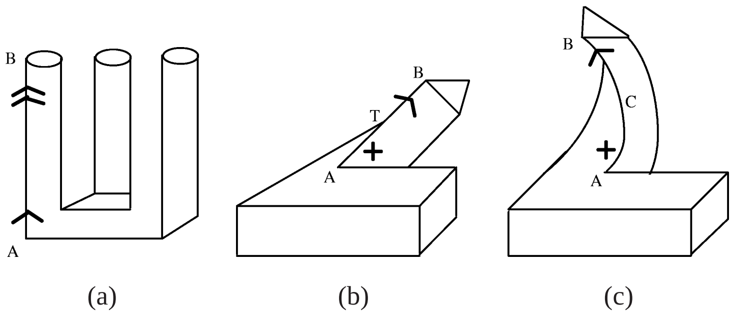

Fig. 1. (a) An impossible object. (b) An object which is impossible if surfaces are planar in the vicinity of straight edges. (c) An object which is possible due to the presence of curved surfaces.

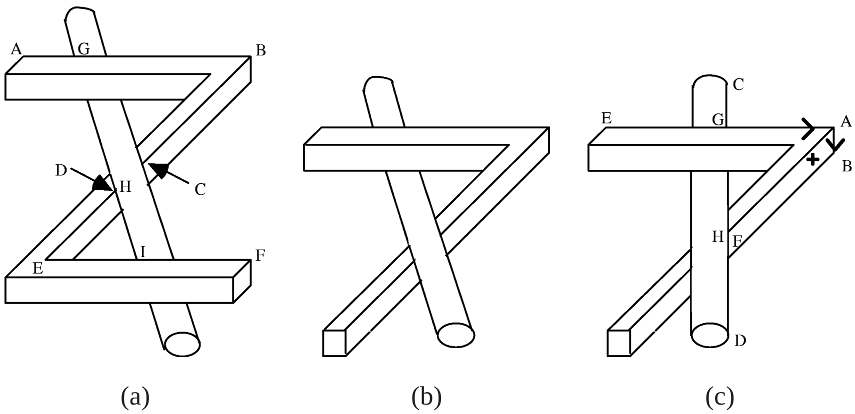

Fig. 2. (a) An impossible drawing. (b) A possible drawing. (c) A drawing which is impossible due to the fact that the cylinder is parallel to line *AB*.

and is denoted by "$-$". The presence of a label transition along *AB* follows from the fact that junction T provides evidence of occlusion and from the list of possible labellings of junction A as a projection of a trihedral vertex [1,9]. However, under the assumption that straight lines are projections of straight edges formed by the intersection of locally planar surfaces, label transitions on straight lines are illegal.

Nevertheless, when the surfaces which meet to form an edge may be curved, undetectable transitions from convex to occluding labels are possible and are known as C-junctions or phantom junctions. Fig. 1(c) shows such a transition at the point C lying somewhere between A and B, which renders this drawing physically realisable.

Fig. 2(a) illustrates another example of a physically unrealisable drawing. Firstly, observe that the points A, B, C, D, E and F are coplanar in 3D. This follows from the colinearity of B, C, D and E together with the fact that lines *AB* and *EF* are parallel. Secondly, observe that the line *GHI* lies behind the plane *ABCDEF* at point G, in front of it at H and then behind it again at I, which is clearly impossible for a straight line.

Fig. 2(b) shows a physically realisable drawing. However, it can be rendered unrealisable by turning the cylinder so that it is parallel to another line in the drawing, as has been done in Fig. 2(c). Under the assumption of a general viewpoint, edges *AB* and *CD* must be parallel in 3D. But the semantic junction labelling shown in the figure implies that point B is on the opposite side of the plane *EAF* to the viewpoint. Since *GH* is parallel to *AB*, the point H must also be on the opposite side of plane *EAF* to the viewpoint, which contradicts the obvious occlusion of the plane *EAF* by the cylinder at H.

Fig. 3 shows another impossible drawing. By colinearity of A, B, C, D and the presence of parallel lines *AE*, *BF*, *CG*, we can deduce that the 3D points A, B, C, D, E, F, G are all coplanar. Points H, I, J, K, L are also colinear. However, I is above the plane *ABCDEFG*, K below it and L above it, all by occlusion at T-junctions. This is clearly a contradiction.

Fig. 4 illustrates an object which is impossible because of coplanarity constraints (which are given in detail, below, in Section 3). It is possible to deduce that A, B, C, D, E are all coplanar by the form of the corresponding junctions in the drawing and by the assumption that surfaces are planar in the vicinity of straight lines. Similarly, the 3D points A, B, F, G, E, D must be coplanar. Together, these two facts imply that the points A, B, E, D are colinear since they all lie on the intersection of these two planes. However, the projections of A, B, E, D in the drawing are clearly not colinear.

The following section gives a formal mathematical statement of the constraints that have been used in this section to demonstrate the impossibility of drawings in Figs. 1–4. The main aim of these constraints is to allow a machine vision system to interpret line drawings of real 3D scenes, by determining the position in 3D of object vertices. The fact that

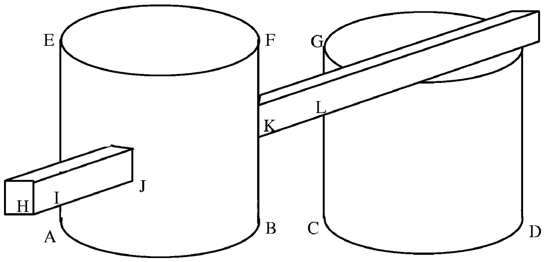

Fig. 3. A drawing which is impossible due to the coplanarity of points A, B, C, D, E, F, G and the contradictory occlusion constraints at I, K and L.

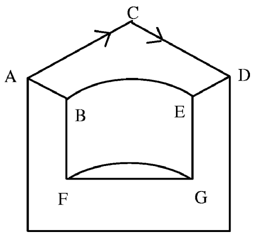

Fig. 4. An object which is impossible because coplanarity constraints imply that points A, B, E, D should be colinear.

these constraints can successfully distinguish between possible and impossible drawings demonstrates their power but is not their main purpose. For simplicity of presentation, in this introductory section we have assumed that drawings are formed by orthographic projection, implying that parallel lines in 3D project into parallel lines in the drawing. In the body of the paper we consider both perspective and orthographic projections.

2. Mathematical formulation of linear constraints

A perspective projection is assumed with focal length f. Note that an orthographic projection produces slightly different constraints, which are given, below, in Section 4. Let (X, Y, Z) represent the 3D scene coordinates and (x, y) the 2D image coordinates. Under perspective projection,

$$(x, y) = (Xf/Z, Yf/Z). \qquad (1)$$

The unknowns of the problem are the Z-coordinates of each junction in the drawing together with the semantic labels of each line-end. Each line-end must be assigned a semantic label such as: concave, convex, occluding or extremal. The two ends of the same curved line do not necessarily have the same label, as was illustrated by the example of Fig. 1(c). Semantic labelling of drawings of curved objects has been discussed extensively in previous papers [2,3,5,12], and we do not repeat here the constraints on junction labellings since they are a function of the different restrictions placed on the shape of objects. Instead we concentrate on the constraints involving the Z-coordinates of junctions. Note that some constraints concern both the line labels and the Z-coordinates, meaning that the line-labelling problem and the determination of the Z-coordinates cannot be solved independently.

Suppose that the junctions are numbered from 1 to n, and let (x_i, y_i) be the image coordinates of junction i, and (X_i, Y_i, Z_i) the scene coordinates of the vertex which projects into junction i. Under perspective projection, the constraints give rise to linear equations and inequalities not between the values Z_i, but between their inverses, which we denote by $t_i = 1/Z_i$. Three types of constraint must be expressed between 3D points h, i, j, k in terms of the values of the variables t_h, t_i, t_j, t_k:

- point i is nearer (further) than point j; or i and j are at equal distance from the projection plane,
- points i, j, k are colinear,
- point h lies in front of (behind) the plane of points i, j, k; or points h, i, j, k are coplanar.

The constraint that point i is nearer than point j is encoded by $t_i > t_j$ and equidistance of i and j from the projection plane is encoded by $t_i = t_j$.

It is easily shown that points i, j, k are colinear iff

$$d_{jk}t_i + d_{ki}t_j + d_{ij}t_k = 0,$$

where $d_{ij} = x_i - x_j$ (and similarly for d_{jk} and d_{ki}) unless $x_i = x_j = x_k$, in which case $d_{ij} = y_i - y_j$.

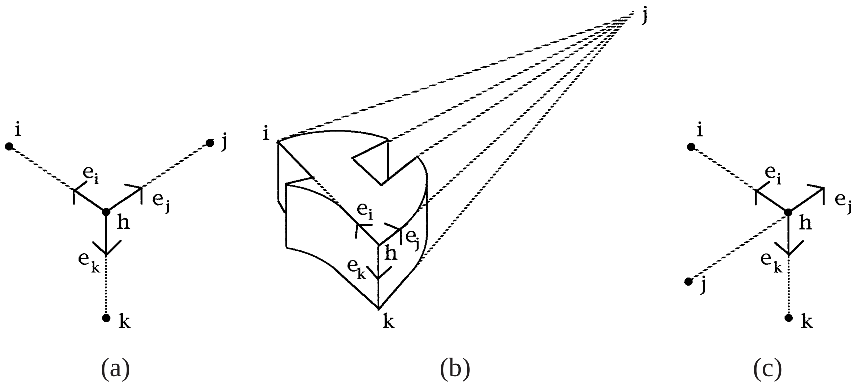

Fig. 5. (a) e_i, e_j, e_k are the 3D orientations of the edges meeting at vertex h. (b) An example of a vertex h involving a vanishing point j. (c) The case in which the vanishing point j is an the opposite side of the vertex h.

In Fig. 5(a), four 3D points h, i, j, k are shown. The unit vectors e_i, e_j, e_k are the orientations in 3D space of the lines hi, hj, hk. As a concrete example, we can consider e_i, e_j, e_k to be the orientations of the tangents to the three edges leaving vertex h in the direction of points i, j, k. Each of hi, hj, hk is shown as a broken line since there may be no continuous straight edge joining h to i, j, k. The points i, j, k may even be vanishing points rather than vertices. Point j in Fig. 5(b) is an example of a vanishing point. In Fig. 5(b), the edge leaving h in direction of i is curved. The vector e_i is the orientation of the tangent to this curve at h. Since, in the drawing, the projection of e_i coincides with the 2D line hi, by implicitly using a colinearity constraint, we deduce that e_i coincides with the line hi in 3D space.

Returning to Fig. 5(a), a test for whether h is in front of (i.e., on the same side as the centre of projection of) the plane ijk is

$$(e_i \wedge e_j).\, e_k < 0 \tag{2}$$

assuming a left-hand coordinate system. This is true for any set of four 3D points h, i, j, k, provided that the triangle of the projections of points i, j, k has a clockwise orientation in the drawing. An anticlockwise orientation simply produces a change of sign and the test (2) becomes $(e_i \wedge e_j).\, e_k > 0$.

For notational convenience, let v_i represent (x_i, y_i, f), where f is the focal length of the perspective projection. In fact, for the constraints given in this paper, knowledge of f is not necessary and we can simply set $f = 1$. By the definition of the perspective projection in Eq. (1), $(X_i, Y_i, Z_i) = v_i Z_i / f$. It follows that

$$e_i = a(v_i Z_i - v_h Z_h),$$
$$e_j = b(v_j Z_j - v_h Z_h),$$
$$e_k = c(v_k Z_k - v_h Z_h),$$

for some strictly positive constants a, b, c. Substituting these equations in (2) and simplifying, gives the following criterion for deciding whether h is in front of the plane ijk:

$$a_{ijk}t_h + a_{jih}t_k + a_{hik}t_j + a_{jhk}t_i < 0, \tag{3}$$

where $a_{ijk} = (v_i \wedge v_j).\, v_k$, with $a_{jih}, a_{hik}, a_{jhk}$ defined similarly. This is a linear inequality in terms of t_h, t_i, t_j, t_k. Recall that $t_i = 1/Z_i$ with t_h, t_j, t_k defined similarly.

Note that there is a complete change of sign for each point i, j or k whose projection is actually on the opposite side of the junction. For example, if the vanishing point j were, in fact, as shown in Fig. 5(c), then the condition (3) would become

$$a_{ijk}t_h + a_{jih}t_k + a_{hik}t_j + a_{jhk}t_i > 0.$$

The final constraint which we have to formulate mathematically is that four points h, i, j, k are coplanar. This constraint is given by the linear equality

$$a_{ijk}t_h + a_{jih}t_k + a_{hik}t_j + a_{jhk}t_i = 0. \tag{4}$$

3. Deriving linear constraints from a drawing

3.1. Vanishing point constraint

A classical assumption in the machine interpretation of line drawings is of trihedral vertices, i.e., at most three edges meet at each object vertex. A natural extension of this assumption is to say that no four (non-colinear) edges would meet at a point in 3D even if they were extended to any finite distance in both directions. It follows, from this assumption and an assumption of a general viewpoint, that the intersection of at least four extensions of lines in the drawing provides evidence of a vanishing point. Various authors have described practical techniques for the detection of vanishing points in images of real scenes [18,22].

Vanishing points are a useful visual cue in scenes containing many parallel lines. All lines converging to a given vanishing point are projections of parallel 3D lines. These 3D lines can be considered to meet at a point which lies at an infinite distance from the viewer. If we have deduced that the projection of i is a vanishing point, then we can apply the vanishing point constraint:

$$t_i = 0.$$

3.2. Constraints on colinearity of three points or intersection of three lines

Section 2 described how to express the colinearity of three scene points in terms of the variables t_i. The colinearity constraint simply states that any three colinear points in the drawing are projections of colinear points in 3D. Note that one of the points in the drawing may be a vanishing point. It is by the combined use of the vanishing point constraint and the colinearity constraint that we exploit the presence of parallel lines under perspective projection.

The dual of the colinearity constraint states that any three straight lines meeting at a point P in the drawing are projections of straight lines meeting at a point q in 3D space.

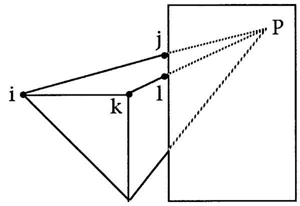

Fig. 6. The intersection of three straight lines at P implies that i, j, k, l are coplanar since the lines ij and kl must intersect in 3D space.

No essentially new constraint is involved, just the identification of q as a 3D point whose coordinates are to be determined. The colinearity constraint can then be applied to each of the lines L meeting at q; if i and j lie on L, then i, j, q are colinear. This dual constraint is redundant when P is a junction or has already been identified as a vanishing point, but is useful, for example, when P is the intersection of the extensions of three lines in the drawing, as in Fig. 6. In this case, although it cannot be determined whether P is a vanishing point or the projection of a hidden or truncated trihedral vertex, we can still deduce, for example, that i, j, k, l are coplanar in 3D.

3.3. T-junction constraint

In a perfect projection of a scene from a general viewpoint, the bar of a T-junction is the projection of a single edge which passes in front of or touches the edge projecting into the stem of the T.

In order to express this mathematically, we need to distinguish two points at every T-junction, as shown in Fig. 7(a). The 3D points i and j project into the same T-junction, but they lie on the two different edges. The basic T-junction constraint is thus

$$t_i \leqslant t_j.$$

However, if it is known that the two edges do not touch, which is the case, for example, if either line has been labelled as extremal, then the point j is strictly nearer to the viewer than i:

$$t_i < t_j.$$

On the other hand, if it is known that the two edges *do* touch, then the constraint is clearly

$$t_i = t_j.$$

There are certain restrictions to the application of the colinearity constraint (described in Section 3.2) to T-junctions. If the occluding edge E is a straight edge, then the occluded point i should clearly not be considered to be colinear with E in 3D space. Similarly, if the occluded edge E' is a straight edge, then the occluding point j should not be considered to be colinear with E'. However, in all other cases, the fact that a T-junction is a member of a set of colinear points (at least two of which are viewpoint independent) implies, under the general viewpoint assumption, that the two edges touch in 3D and hence $t_i = t_j$. The presence in the drawing of shadows or reflections leaving a T-junction can also provide

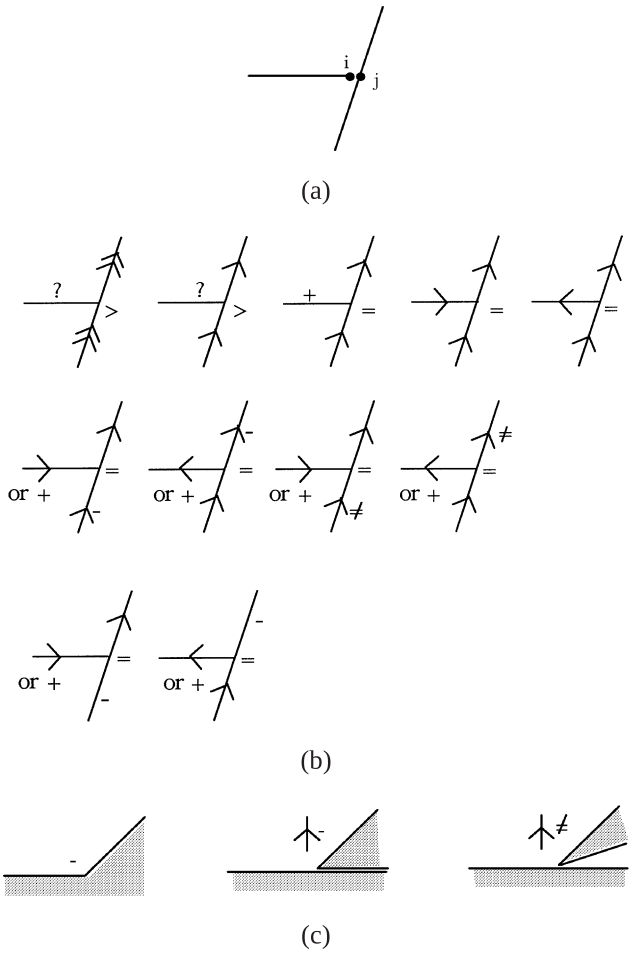

Fig. 7. (a) The two points in 3D which project into a T-junction. (b) The list of legal labellings for T-junctions, including the label ("$>$" or "$=$") for the junction itself. (c) The three types of concave edges ("$-$", "$\rightarrow -$" and "$\rightarrow \neq$").

evidence that the two edges touch in 3D, under an assumption of general viewpoint and general light-source positions.

3.4. T-junction constraint for objects with trihedral vertices

To distinguish the two distinct types of T-junction (edges touching or not touching), we propose that a semantic label be assigned to each T-junction in the drawing. The labels "$=$" and "$>$" indicate, respectively, that the two edges touch or do not touch at a 3D point projecting into the T-junction. The introduction of a new label for T-junctions ($=$ or $>$) is a way of ensuring that the determination of the values of the variables t_i remains a classical linear programming problem.

Assigning a label to each T-junction produces a new list of legal labellings for T-junctions, as shown in Fig. 7(b). All object vertices are assumed to be trihedral. A question mark indicates that any label is valid. We distinguish three different types of concave edge: a concave edge (labelled "$-$") which is part of a single object, a concave edge (labelled "$\rightarrow -$") separable into two objects which are such that a face of one of the objects touches a face of the second object, a concave edge (labelled "$\rightarrow \neq$") separable into two objects which are such that only the edge of one of the objects touches a face of the second object. The cross-section of these three types of concave edges are shown in Fig. 7(c), where, in each case, the edge is viewed from above. Edges of type "$\rightarrow -$" and "$\rightarrow \neq$" can reasonably be assumed to only occur in multi-object drawings.

Note that we have not assumed that a T-junction is necessarily caused by occlusion. Non-occlusion T-junctions (which occur, for example, at the join of two long rectangular blocks which are stuck together in the form of a cross [2]) have been incorporated in the list of labellings of T junctions. Disallowing non-occlusion T-junctions simply means disallowing the label "$-$" for the bar of the T. If non-occlusion T-junctions are allowed, then the label "$\rightarrow -$" becomes superfluous: in all legal interpretations of the drawing, all "$\rightarrow -$" labels can be replaced by "$-$" to leave another legal interpretation. (In the notation of the constraint satisfaction problem, label "$-$" is neighbourhood substitutable for "$\rightarrow -$" in all domains [4,7].) For drawings of a single object in which non-occlusion T-junctions are disallowed, the labelling scheme is much simpler, since there are no "$=$" T-junctions and no "$\rightarrow -$" or "$\rightarrow \neq$" edges.

The basic constraints on a T-junction consist of the list of junction labellings of Fig. 7(b) together with the constraint that $t_i = t_j$ if the junction is labelled "$=$" and $t_i < t_j$ if the junction is labelled "$>$". Further constraints on T-junctions exist, but only in the case that the junction is labelled "$=$" and it is part of a pair of junctions joined by a straight line. Such constraints involving pairs of adjacent junctions are described below, in Sections 3.6 and 3.7.

In this section we have assumed that T-junctions are caused by an edge occluding another edge ("$>$") or by two surface-normal discontinuity edges intersecting in 3D space ("$=$"). If shadows, cracks [24], reflections or other surface markings may occur in the drawing, then this gives rise to yet another type of T-junction, which is neither "$>$" nor "$=$".

3.5. Convex/concave edge constraints

Various catalogues of legal junction labellings exist for drawings of objects with curved surfaces [2,3,5,12]. Different assumptions on object shape give rise to different catalogues. In order to be able to state very general constraints derived from linear features in the drawing, we do not specify which catalogue is used. Nonetheless, we suppose that some such catalogue has been applied to gain certain information about line labels. The constraints described in this section are derived from knowledge that a line-end or a straight line is labelled as concave ($+$) or as convex ($-$). Whereas Parodi and Torre [15] deduced line label information from knowledge of the directions of the three edges meeting at a vertex, we deduce information concerning edge directions from knowledge of line labels.

The resulting constraints are not simple inverses of those stated by Parodi and Torre but also generalisations to curved surfaces meeting at non-trihedral vertices.

In order to be able to deduce useful constraints, we require that objects behave locally as polyhedra in the vicinity of vertices or straight edges. To be more specific, we require that:

- For any two object surfaces S_1, S_2 meeting along an edge E incident to a vertex V, the tangent-planes T_1, T_2 to S_1, S_2 at V intersect in a straight line L which is tangential to E at V. (This follows, for example, from the assumption that objects have C^3 surfaces, and that no edges or surfaces meet tangentially.)
- For any straight edge E on a surface S, there is a unique tangent-plane to S along the whole length of E.

Note, however, that we do not need to make the assumption that vertices are trihedral.

Sugihara [20] formulated a basic constraint on convex/concave edges formed by the intersection of two planar faces: When two planar faces F_1, F_2 meet at an edge E and point i lies on F_1 (but not in the plane of F_2) then the semantic label (convex or concave) for E determines whether i lies in front of or behind the plane of F_2. For example, in Fig. 8(a), the fact that jk is a concave edge implies that i is necessarily in front of the plane jkl. To apply this constraint to curved objects we have to consider the polyhedral vertex formed by the tangent planes to the surfaces, rather the surfaces themselves.

Suppose that the edge E, incident to a vertex j, is the intersection of two surfaces S_1 and S_2 which are both visible in the drawing. E is thus either a convex or a concave edge. Suppose that k lies on the straight line in 3D which is tangent to E at j. Let T_1, T_2 be the tangent-planes to S_1, S_2 at j and suppose that we know that points i, l lie on T_1, T_2, respectively, but not on their intersection (which, by the first assumption above, is the line jk). Knowing whether E is a convex or concave edge, we can deduce whether i is in front of or behind the plane passing through j, k, l (which is exactly T_2). Section 2 described how to code such constraints mathematically as strict linear inequalities.

In Fig. 8(a), the concavity of jk implies that i is in front of the plane jkl. In fact, this remains true even if the edge jl is an occluding edge passing in front of the surface incident to the concave edge, as illustrated in Fig. 8(b). Indeed, in this particular case, in which jk is a concave edge and the projections of i and l lie on opposite sides of the projection of jk, the constraint holds whatever the labelling of edges ij and jl.

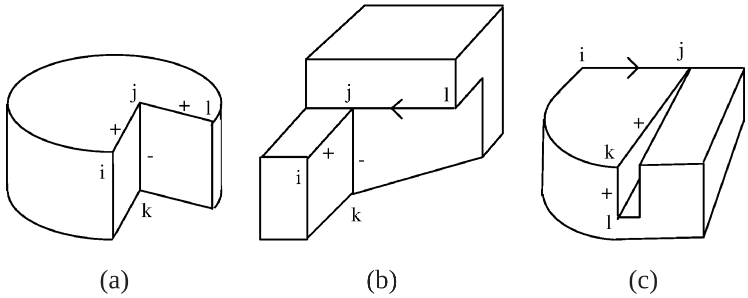

Fig. 8. In (a) and (b) the fact that jk is a concave edge implies that i lies in front of the plane jkl. In (c) the convexity of the edge jk implies that i lies behind the plane jkl.

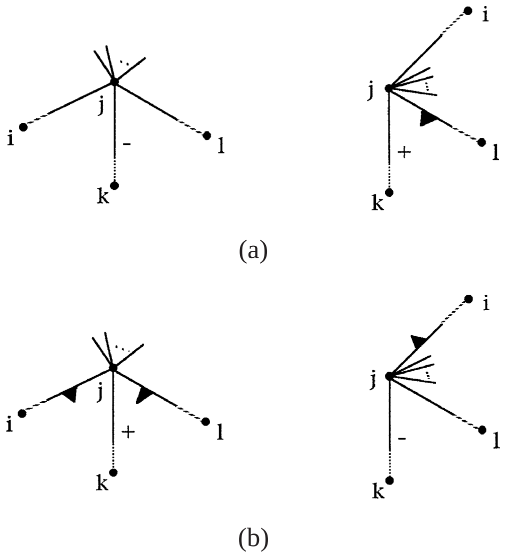

Fig. 9. Constraints derived from a convex or concave label for a line-end: (a) i is in front of the plane jkl; (b) i is behind the plane jkl.

In all, there are four distinct cases to consider, depending on whether the edge jk is labelled convex or concave and whether the projections in the drawing of the points i, l lie on the same or different sides of the projection of jk. These four cases are illustrated in Fig. 9: in the two cases shown in Fig. 9(a), i must be in front of the plane jkl; in the two cases shown in Fig. 9(b), i must be behind the plane jkl.

There is no upper bound on the number of lines which meet at the junction in the drawing. As we walk around the junction in a clockwise direction, starting from the projection of jk, the first line we encounter is the projection of the edge whose tangent is ji; walking in an anticlockwise direction, the first line we encounter is the projection of the edge whose tangent is jl. The only restriction to the application of this constraint is that the projections of points i, j, l are not colinear. We introduce a new label ▼ which, on a horizontal line, means that the edge which projects into this line lies on the surface which is visible just below the line. It is thus a generic label representing any semantic label (e.g., $+, -, \rightarrow, \rightarrow -, \rightarrow \neq, - \leftarrow, \neq \leftarrow$, shadow, crack, reflection, surface mark, ramp line [2]) except an occluding edge with the occluding object above the line. The label ▼ is not always necessary, as was illustrated by the example of Fig. 8(b), in which i lies in front of the plane jkl even when jl is an occluding edge.

To be able to deduce an inequality constraint from a junction, we require at least one line label ($+$ or $-$), together with the three points i, k, l (such as vertices or vanishing points) which are colinear with the tangents to three distinct edges meeting at the vertex j. If the tangent to an edge at the vertex j does not pass through a vertex or vanishing point, then an artificial point i (or k or l) can be created on this tangent.

To encode mathematically the fact that an edge is concave (or convex), we require points i and l which lie on the tangent-planes T_1 and T_2 to the two surfaces S_1 and S_2 which

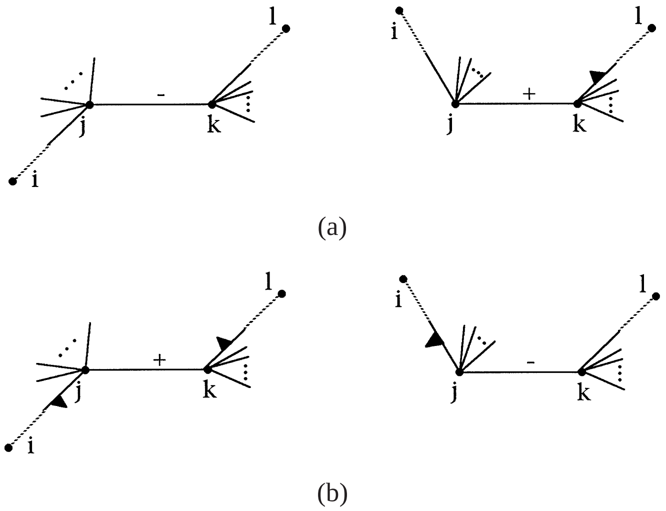

Fig. 10. (a) i is in front of the plane jkl; (b) i is behind the plane jkl.

intersect to form the edge. If, for example, the first edge visible in the drawing when leaving jk in an anticlockwise direction is an edge which occludes the surface S_2 (e.g., edge jl in Fig. 8(b)), then we will have to look further afield to find a point l which actually lies on the tangent-plane T_2 to S_2. If jk is a straight edge, then we may be able to find such a point l, lying on T_2, on the tangent to an edge at the vertex k. An example is shown in Fig. 8(c), where the convexity of edge jk implies that i is behind the plane jkl. In generalising this constraint, there are again four distinct cases to consider, depending on whether the straight edge jk is convex or concave and whether the projections of i and l lie on the same or opposite sides of the projection of jk. The four cases are shown in Fig. 10. In the cases given in Fig.10(a), i must lie in front of the plane jkl; in Fig. 10(b) i must lie behind the plane jkl. The only restrictions to the application of this constraint are that the projections of j and k are not $T^{>}$ junctions and that neither i nor l is colinear with the line jk.

For simplicity of presentation, a concave edge jk is labelled "$-$" in Figs. 9 and 10. Note, however, that the same constraints apply when the label "$-$" is replaced by any other of the labels for concave edges: $\rightarrow -$, $\rightarrow \neq$, $\leftarrow -$ or, $\leftarrow \neq$. Reflected versions of the constraints in Figs. 9 and 10 also exist but have been omitted for brevity of presentation.

3.6. Coplanarity constraints derived from a straight line

Sugihara's classic work [20,21] used an assumption of planarity of object surfaces to deduce information about the relative depth of scene points. Rather than making the restrictive assumption that objects are polyhedra, we make only very weak assumptions disallowing coincidences and improbable objects. We show in this section that planarity can, in fact, be deduced from these weak assumptions.

As in the previous section, we assume that a straight line L in the drawing is the projection of a straight edge E formed by the intersection of two surfaces (or three surfaces

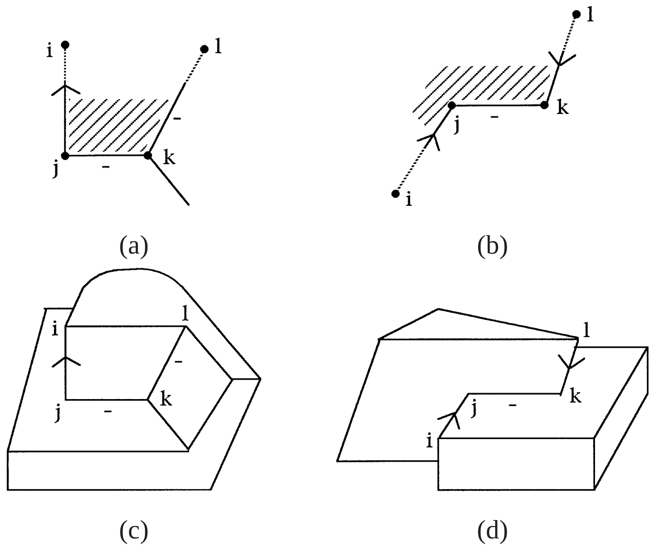

Fig. 11. (a) A junction pair in which points i, j, k, l must be coplanar. (b) A junction pair in which i must be in front of the plane of jkl. (c) An object containing the junction pair of (a). (d) An object containing the junction pair of (b).

in the case of $\rightarrow -$ or $\rightarrow \neq$ edges) which are planar in the vicinity of E. Similarly, every junction is assumed to be the projection of a vertex which behaves locally as a polyhedral vertex. Based on these assumptions, it is often possible to deduce a constraint concerning the relative positions in 3D space of E and other edges emanating from the vertices at either end of E. It is either an equality or inequality constraint depending on the labellings of the junctions at the ends of L.

Fig. 11(a) shows an example where points i, j, k, l are necessarily coplanar. Fig. 11(b) shows an example where the 3D point i is necessarily in front of the plane of jkl. Fig. 11(c) (Fig. 11(d)) illustrates a concrete realisation of the junction pair shown in Fig. 11(a) (Fig. 11(b)).

In the example in Fig. 11(a), the occluding and concave edge labels indicate that the four points i, j, k, l all lie on the shaded face. The example in Fig. 11(b) differs in that the occluding edge label for ji indicates that i is, in fact, in front of the shaded face. Although, in the examples given in Figs. 11(c) and 11(d), ij and kl are straight edges, this is not necessarily the case. The edge leaving j in direction of i (and the edge leaving k in direction of l) may be curved. The points i and l may be vanishing points, vertices which just happen to lie on the tangent to the edge or artificial points (as described in Section 3.5) lying on the tangent to the edge.

Fig. 12 shows a general coplanarity constraint. The projections of the points i and l may lie on either side of the projection of the straight edge jk. As in Section 3.5, the generic label ▲ on a horizontal line represents any type of edge which lies on the surface projecting into the region above the line. The label ▲ for ij, jk and kl indicate that the points i, j, k, l

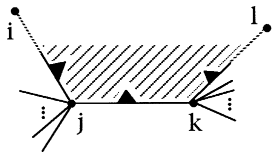

Fig. 12. The coplanarity of i, j, k, l follows from the fact that jk is a straight edge.

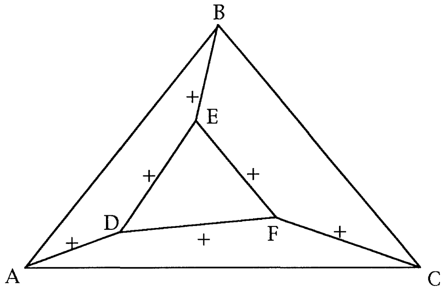

Fig. 13. An impossible object since the lines *AD*, *BE* and *CF*, when extended, should meet at a point, which is not the case.

are coplanar, since they all lie on the plane which passes through jk and is tangential to the surface which is visible above edge jk. This surface is shown shaded in the figure.

As an example of the use of the coplanarity constraint, described above, consider the drawing in Fig. 13 (adapted from [19]). The coplanarity constraint applied to the straight line *DE* implies that the points A, D, E, B are coplanar. Similarly, B, E, F, C are coplanar, as are C, F, D, A. However, it is impossible to find Z-coordinates for points A to F satisfying these coplanarity constraints. To see this, consider the three lines *AD*, *BE* and *CF* which, when extended, should meet at a point, which is not the case. The coplanarity constraints coded as linear equations thus show that this is an impossible figure.

Certain coplanarity constraints are clearly redundant and need not be taken into account. For example, the constraint "i, j, k, l coplanar" provides no information if $i = l$ or if i, j, l (or i, k, l) are known to be colinear. Similarly, an entirely visible planar face bounded by r edges can give rise to a coplanarity constraint on each set of four consecutive vertices. Since only $r - 3$ such constraints are necessary to establish the coplanarity of all vertices on the face, three of these constraints are redundant and can be ignored.

Fig. 14 shows how the presence of occluding labels gives rise to an inequality constraint rather than an equality constraint. In Fig. 14(a), the fact that ij is the tangent to an occluding edge means that i lies in front of the plane passing through edge jk and tangential to the surface which is visible in the drawing above jk (and shown shaded in the figure). Similarly, in Fig. 14(b), both i and l lie above this tangent-plane. The resulting constraint is the same as for Fig. 14(a), that i must lie in front of the plane jkl.

In Fig. 14(a) the projections of i and l may lie either above or below the projection of the edge jk, whereas in Fig. 14(b) the projections of i and l must lie on opposite sides of this line. In both cases, the constraint does not apply if the projections of j or k are $T^{>}$

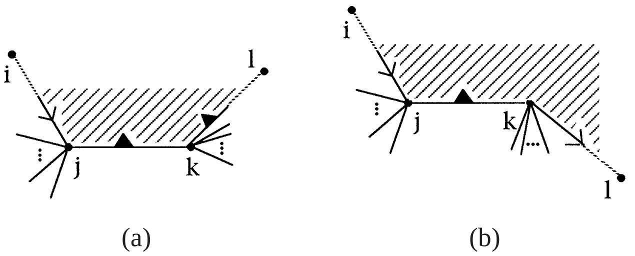

Fig. 14. The presence of occluding labels imply that i lies in front of the plane jkl.

junctions. A reflected version of the constraint in Fig. 14(a) also exists with the occluding edge on the right hand side.

If objects may have tangential edges and surfaces [3], then the constraint is less strong, in that i may be in front of *or on* the plane jkl.

In the constraints given in Figs. 12 and 14, the projections of j and k need not be adjacent junctions. The constraints still hold even when there are any number of T junctions along the projection of jk in the drawing; we only require that the projection of jk be a continuous straight line.

3.7. Hidden-surface coplanarity constraints

Under the more restrictive assumption of trihedral vertices, other constraints exist in which the plane on which points j, k, l lie is not a tangent-plane to a visible surface, but to a hidden surface. An example is shown in Fig. 15, where the points i, j, k, l are coplanar: they all lie on the same hidden face.

Fig. 15 illustrates just one example of a very general constraint given in Fig. 16. The points i, j, k, l must be coplanar (since they lie on the same tangent-plane to the hidden surface, shown shaded) if the trihedral vertices j and k are joined by a straight occluding edge, as shown in the figure. The projections of j and k may be any Y or W junctions, and the projections of i and l may lie on any side of the the projection of edge jk. The occluding edge label "$\rightarrow$" may be replaced by either "$\rightarrow -$" or "$\rightarrow \neq$": the constraint still holds.

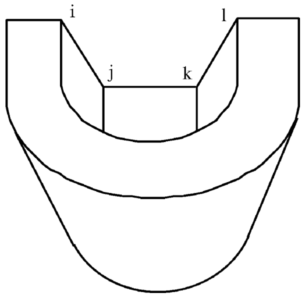

Fig. 15. An example in which the coplanar points i, j, k, l lie on a hidden face. Vertices j and k are assumed to be trihedral.

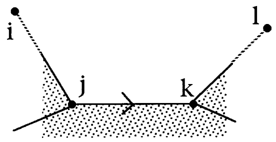

Fig. 16. If j and k are projections of trihedral vertices then i, j, k, l are coplanar.

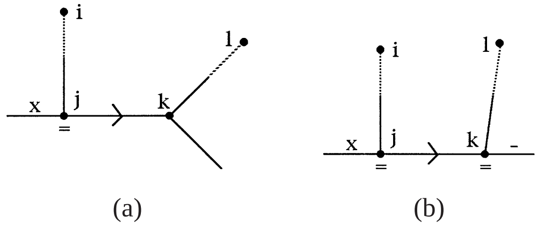

Fig. 17. The point i lies in front of or on the plane jkl, since the continuation of ij passes behind the hidden surface.

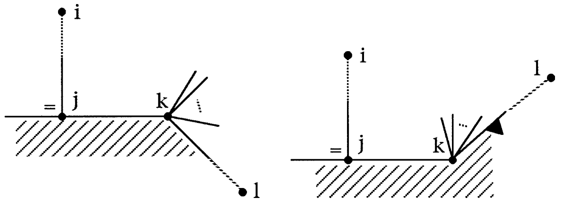

Fig. 18. The point i must lie in front of the plane jkl, since the continuation of ij passes behind the visible shaded surface.

It is worth observing that the occluding edge label is inessential in the statement of this constraint; if the edge jk has any other label then i, j, k, l are still coplanar by the constraint of Fig. 12. This result follows from the catalogue of possible labellings for Y and W junctions as projections of trihedral vertices [1,9].

Hidden-surface constraints also arise when the bar of a $T^=$ junction is joined to another junction by a straight line, since the continuation of the stem of the $T^=$ junction must pass behind the hidden surface. In the two cases shown in Fig. 17, i must lie in front of or on the plane jkl. Furthermore, if it is known that the label x is either "$-$" or "$\rightarrow -$", then the constraint is much stricter: i, j, k, l must be coplanar. In Fig. 17(a), the projection of vertex k is any Y or W junction. The line labelled "$\rightarrow$" in Fig. 17(a) could be labelled "$\rightarrow \neq$". The line labelled "$-$" in Fig. 17(b) could be labelled "$\rightarrow -$". In both cases, the same constraints still hold.

Although the continuation of the line ij, in Fig. 18, may touch the hidden surface, it certainly passes behind the visible shaded surface. Thus the constraint in this case is that i lies in front of the plane jkl. If tangential edges and surfaces are allowed, then the constraint is weaker: i lies in front of or on the plane jkl.

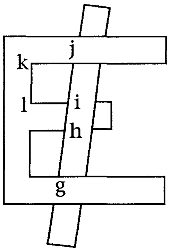

Fig. 19. A drawing which cannot be realised as the projection of objects with solid trihedral vertices.

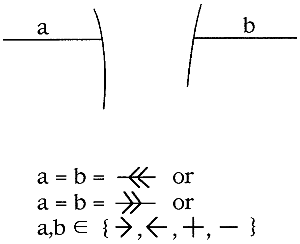

Fig. 20. Labelling constraint on colinear straight-line segments.

The projection of vertex k in Fig. 18 can be any junction except a $T^{>}$ junction. Since vertices are assumed to be trihedral, at most three lines meet at this junction. The label ▼ is only necessary when the projections of i and l lie on the same side of the projection of jk, as shown in the right hand side of Fig. 18.

In the constraints given in Figs. 16, 17 and 18, the projections of j and k need not be adjacent junctions. The constraints still hold even when there are any number of T junctions along the projection of jk in the drawing. Reflected versions of the constraints in Figs. 17 and 18 clearly also exist. Note that the constraints in Figs. 17 and 18 do not hold if j can be a non-trihedral vertex, since in this case i could lie behind the plane jkl.

As an example of the strength of the hidden-surface constraints described in this section, consider the drawing in Fig. 19. This drawing represents two interlocking wafer-thin objects, a letter 'E' and a letter 'I'. The planarity of the visible surface of the 'E' can be deduced from the presence of parallel lines. It follows that the T-junctions g, h, i, j are all $T^{=}$ junctions and that the points i, j, k, l are coplanar. A contradiction arises, when trying to interpret the drawing as a projection of objects with solid trihedral vertices, through the application of the constraint illustrated in Fig. 18, which implies that i lies in front of the plane jkl.

3.8. Straight line labelling constraints

The assumption that a straight line is a projection of the intersection of two surfaces which are planar in the vicinity of the intersection, implies that there are no label transitions on a straight line. This constraint is useful during the determination of a semantic labelling of the drawing, and has been implicitly applied in the derivation of the constraints in Sections 3.5, 3.6 and 3.7, above.

When two straight-line segments are colinear, as in Fig. 20, they do not necessarily have the same semantic label. However, under the general viewpoint assumption, a transition from a viewpoint-dependent to a viewpoint-independent label is impossible. This leaves the combinations of labels (a, b) which are given in Fig. 20.

4. Orthographic projection

Under orthographic projection, as opposed to perspective projection, the values to be determined are not the inverses $t_i = 1/Z_i$, but the distances Z_i themselves. It turns out that, to mathematically encode the three basic constraints (equidistance, colinearity, coplanarity) under orthographic projection, it is sufficient to replace t_i by Z_i in the equations given in Section 2. Thus, the mathematical formulation of the constraint that i and j are equidistant from the projection plane is $Z_i = Z_j$; the constraint that i, j, k are colinear is $d_{jk}Z_i + d_{ki}Z_j + d_{ij}Z_k = 0$; the constraint that h, i, j, k are coplanar is $a_{ijk}Z_h + a_{jih}Z_k + a_{hik}Z_j + a_{jhk}Z_i = 0$. The coefficients d_{ij} and a_{ijk} are as defined in Section 2. Note that a_{ijk} is defined as $(v_i \wedge v_j).\, v_k$, where v_i represents $(x_i, y_i, 1)$.

In the inequality constraints there is a change of sign, due to the fact that the inequalities concern the Z_i rather than their inverses t_i. Thus, the constraint that i is nearer than j is $Z_i < Z_j$; the constraint that h lies in front of the plane ijk (where h, i, j, k are as shown in Fig. 5(a)) is now $a_{ijk}Z_h + a_{jih}Z_k + a_{hik}Z_j + a_{jhk}Z_i > 0$.

Under orthographic projection, there are no vanishing points (and hence no vanishing point constraint). Parallel lines in 3D project into parallel lines in the drawing, which can theoretically be detected. Let i, j, k, l be four points in 3D such that the projections of the lines ij and kl are parallel in the drawing. Then the parallel lines constraint states that

$$d_{kl}(Z_i - Z_j) - d_{ij}(Z_k - Z_l) = 0,$$

where d_{kl}, d_{ij} are defined as in Section 2. Note that this constraint could be applied even when there is no line joining i and j (or k and l) in the drawing.

All other constraints are identical to the perspective projection case. We will not discuss any further the case of orthographic projection, since it produces a computational problem of exactly the same type as for perspective projection, namely a line-labelling problem together with a system of linear equations and inequalities.

5. Physical realisability of drawings

Sugihara [20,21] showed that line-labelling constraints together with linear constraints provide necessary and sufficient conditions for a drawing of a polyhedral scene to be physically realisable. In a previous paper [5] we showed that line-labelling constraints alone provide necessary and sufficient conditions for a drawing to be physically realisable under the assumptions that object faces are arbitrary C^3 curved surfaces, that vertices are trihedral and that all lines (even straight lines) can be projections of arbitrary C^3 curves. The linear constraints described in this paper complement the line-labelling constraints for curved objects [5,12]. They correct the rather unreasonable assumption that a straight line can be the projection of a curved edge.

It is straightforward to modify the proof given in [5] to show that the line-labelling constraints together with the linear constraints given in this paper provide necessary and sufficient conditions for the physical realisability of a drawing. They are clearly necessary. Given a legal line-labelling and the positions $(Z_1, \ldots, Z_n)$ of all points projecting into junctions in the drawing, then a 3D scene whose projection is identical to the drawing can be constructed as in [5] thanks to the large freedom of choice in the shape of object faces and those edges which do not project into straight lines.

The linear constraints in this paper are inspired by Sugihara's [20] constraints for polyhedral objects but they go further in that they include constraints on colinear points, parallel lines and hidden faces. Most importantly, they can be applied without the restrictive assumption of planar surfaces.

6. The computational problem

The constraints described in Section 3 when applied to a labelled drawing produce a set of linear equations

$$At = 0, \tag{5}$$

and a set of linear inequalities

$$\begin{aligned} Bt &> 0, \\ Ct &\geqslant 0, \end{aligned} \tag{6}$$

where $t = (t_1, \ldots, t_n)$.

Let $d = (\delta, \ldots, \delta)$, where δ is any strictly positive constant. The system of equations (5) and inequalities (6) has a solution iff the following set of inequalities together with (5) has a solution:

$$\begin{aligned} Bt &\geqslant d, \\ Ct &\geqslant 0. \end{aligned} \tag{7}$$

This result follows from the size/distance ambiguity inherent in any drawing, meaning that any solution $(t_1, \ldots, t_n)$ to (5) and (6) can simply be scaled up uniformly so as to satisfy (7). Thus, the linear constraints applied to a labelled line drawing produce a standard linear programming problem.

Suppose that, by applying the constraints described in Section 3 to a drawing, we have deduced that points A, B, C, D are coplanar, A, B, E are colinear and that B, C, D, E are coplanar. The third of these constraints is redundant since it could be logically deduced from the other two. In the presence of such linear dependence between equations in the system (5), any errors in the position of junctions in the drawing will mean that the system of equations (5) has no solution. Indeed, errors are inevitable due to the limited precision of floating-point numbers stored in a computer.

One solution is a scheme in which the combined system of equations (5) and inequalities (7) is replaced by the inequalities

$$\begin{aligned} e \geqslant At &\geqslant -e, \\ Bt &\geqslant d - e', \\ Ct &\geqslant -e'', \end{aligned} \tag{8}$$

where $e = (\varepsilon_1, \ldots, \varepsilon_n)$, $e' = (\varepsilon'_1, \ldots, \varepsilon'_n)$, $e'' = (\varepsilon''_1, \ldots, \varepsilon''_n)$ for some small $\varepsilon_i, \varepsilon'_i, \varepsilon''_i > 0$. It is clearly essential to choose $\delta > \varepsilon'_i$ (for at least one i), otherwise the system (8) has a trivial solution in which all points are coplanar. The values chosen for $\varepsilon_i, \varepsilon'_i, \varepsilon''_i$ determine tolerances in the linear constraints, for example, the extent to which a supposedly planar surface can actually be curved (i.e., the distance of a vertex from the plane in which it is supposed to lie). The value chosen for δ, on the other hand, imposes a lower bound on the distance of a vertex V from a plane P, when it is known that V lies in front of (or behind) P. In order to avoid physically unrealisable drawings being accepted as realisable, due to the tolerances $\varepsilon_i, \varepsilon'_i, \varepsilon''_i$, we recommend choosing δ to be an order of magnitude greater than all of $\varepsilon_i, \varepsilon'_i, \varepsilon''_i$.

Sugihara [19] suggested finding a maximal linearly-independent subset of Eq. (5); a solution t to this subsystem can then be used to determine whether corrections in the positions (x_i, y_i) of junctions in the drawing can be made so that the whole system (5) has a solution. Unfortunately, by introducing new constraints compared with Sugihara (concerning colinearity and parallel lines), potential redundancy in the system (5) is such that the number of equations may be greater than the total number of variables (even when all the values of x_i, y_i, t_i are treated as variables). Thus, although the technique can correct certain drawings, the inability to find a correction does not imply the physical impossibility of the drawing.

Whichever technique is used to solve the linear programming problem given by (5) and (7), this is only part of the problem. Such a linear programming problem may have to be solved for each legal labelling of the drawing, of which there may be an exponential number. Even determining whether a line drawing of a polyhedron with trihedral vertices has a single legal semantic labelling is unfortunately an NP-complete problem [11]. The straight line labelling constraint (Section 3.8), which prohibits label transitions on straight lines in the drawing, implies that the labelling problem for curved objects with trihedral vertices is also NP-complete, since it contains the labelling problem for drawings of polyhedral scenes as a subproblem.

In fact, it was shown in [5] that the realisability problem (the problem of deciding whether a drawing is realisable as the projection of a 3D scene) for drawings of curved objects with trihedral vertices is NP-complete under either perspective or orthographic projection due to the combination of junction labelling constraints and constraints derived from the presence of parallel lines in 3D. The introduction of additional constraints in this paper does not change this result; in the constructions in the proof of NP-completeness in [5] it is sufficient to avoid the presence of colinear points and straight lines so that none of the additional constraints apply.

Nevertheless, the local propagation of line labels is a powerful tool in eliminating illegal labels for line-ends. Furthermore, we have presented the linear constraints of

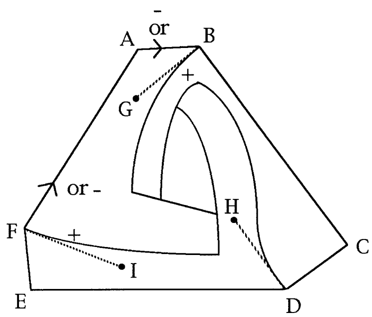

Fig. 21. A figure which is discovered to be impossible by the propagation of junction labelling constraints for trihedral vertices followed by the application of the coplanarity constraint.

Section 3 in such a way that they can be applied with minimal knowledge of actual line labels. An incomplete algorithm, although very effective on all the example drawings given in this paper, is to propagate line labels through local consistency operations until convergence [14,23,24], build the corresponding linear constraints and solve the resulting linear programming problem.

As an example, consider the drawing in Fig. 21 which is a curved version of the well-known Penrose triangle. Under the assumption of trihedral vertices, the catalogue of legal junction labellings [2,12] provides sufficient information about line labels to deduce that I, F, A, B are coplanar and F, A, B, G are coplanar, by two applications of the coplanarity constraint given in Fig. 12. Thus, I, F, A, B, G are coplanar. By symmetry, G, B, C, D, H are coplanar and H, D, E, F, I are coplanar. These three planes should then intersect at a point in 3D space, which implies that the projections of *BG*, *DH* and *FI*, when extended, should meet at a point in the drawing. This is clearly not the case. Since the corresponding linear programming problem has no solution, we can deduce that this is an impossible figure.

This example illustrates the utility of creating artificial points (G, H, I in Fig. 21) on tangents to edges. Note that the edges *FA* and *AB* are not assigned a unique label after local propagation of constraints; the coplanarity constraint can nevertheless still be applied since "$\rightarrow$" and "$-$" both belong to the set of labels represented by the generic label ▼.

7. Heuristics to reduce residual ambiguity

All drawings contain some unresolvable ambiguity, due to the loss of depth information when the scene is projected into the drawing. As an example, consider the drawing of Fig. 22(a), which we see as a cube, the only ambiguity remaining concerning its distance/size. However, this drawing could be of any of a whole family of parallelepipeds. In identifying it as a cube, we make the hidden, and purely heuristic, assumption that the three lines *AB*, *AC*, *AD* meet at right angles in 3D.

Fig. 22(b) shows a figure which remains ambiguous even after assuming that A is the intersection of three orthogonal lines in 3D. Among the infinite number of remaining interpretations of this drawing, we tend to prefer two particularly simple interpretations,

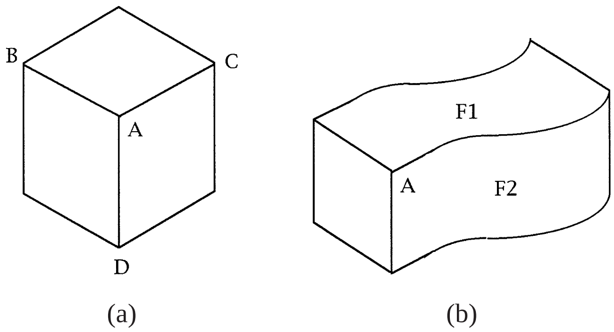

Fig. 22. (a) This parallelepiped is seen as a cube when it is assumed that lines AB, AC, AD meet at right angles. (b) The two most likely interpretations of this shape occur when $F1$ or $F2$ is seen as a planar surface.

in which either face $F1$ or face $F2$ is assumed to be planar. In either case, the 3D positions of all vertices can be deduced from this assumption.

These examples illustrate our preference for right angles and planar faces. A more general heuristic, than a simple preference for planar faces, is to minimise the curvature of 3D curves and surfaces, since the smoothest curve or surface provides the simplest interpretation. Unfortunately, if the conclusions that can be drawn from the application of heuristics can be contradictory (as in Fig. 22(b) where $F1$ and $F2$ cannot both be planar) then applying a global optimisation criterion is likely to produce a theoretically intractable problem.

8. Conclusion

The constraints stated in this paper represent a generalisation of known constraints for polyhedral objects to the class of objects with curved surfaces. These constraints are capable of identifying classic examples of impossible figures involving polyhedra but also many more new examples of impossible figures involving curved objects. Instead of assuming planarity of surfaces, the constraints allow us to deduce planarity from labelled pairs of junctions joined by a straight line.

Although differentiating between possible and impossible figures is theoretically an NP-complete problem, polynomial-time algorithms for the propagation of semantic labels, together with the linear programming approach proposed in this paper, provide a practical (though incomplete) test for physical realisability of figures.

The successful integration of information from various sources (junctions, colinear points, straight lines, vanishing points), in order to recover the depth of object vertices, can be considered as an essential first step towards a complete computer vision system which would also incorporate information from other depth recovery techniques, such as shape-from-shading [8,13] and stereovision [6]. The constraints presented in this paper are based on the assumption that the drawing is a perfect projection of a 3D scene. They have direct applications in the reconstruction of a 3D scene from a user-entered drawing.

References

[1] M.B. Clowes, On seeing things, Artificial Intelligence 2 (1971) 79–116.
[2] M.C. Cooper, Interpretation of line drawings of complex objects, Image and Vision Computing 11 (2) (1993) 82–90.
[3] M.C. Cooper, Interpreting line drawings of curved objects with tangential edges and surfaces, Image and Vision Computing 15 (1997) 263–276.
[4] M.C. Cooper, Fundamental properties of neighbourhood substitution in constraint satisfaction problems, Artificial Intelligence 90 (1997) 1–24.
[5] M.C. Cooper, Linear-time algorithms for testing the realisability of line drawings of curved objects, Artificial Intelligence 108 (1999) 31–67.
[6] O. Faugeras, Three-Dimensional Computer Vision, MIT Press, Cambridge, MA, 1993.
[7] E.C. Freuder, Eliminating interchangeable values in constraint satisfaction problems, in: Proc. AAAI-91, Anaheim, CA, 1991, pp. 227–233.
[8] B.K.P. Horn, Robot Vision, MIT Press, Cambridge, MA, 1986.
[9] D.A. Huffman, Impossible objects as nonsense sentences, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence, Edinburgh University Press, Edinburgh, 1971, pp. 295–323.
[10] T. Kanade, Recovery of the three-dimensional shape of an object from a single view, Artificial Intelligence 17 (1981) 409–460.
[11] L.M. Kirousis, C.H. Papadimitriou, The complexity of recognizing polyhedral scenes, J. Comput. System Sci. 37 (1) (1988) 14–38.
[12] J. Malik, Interpreting line drawings of curved objects, Internat. J. Computer Vision 1 (1987) 73–103.
[13] J. Malik, D. Maydan, Recovering three-dimensional shape from a single image of curved objects, IEEE Trans. Pattern Anal. Machine Intell. 11 (6) (1989) 555–566.
[14] R. Mohr, T.C. Henderson, Arc and path consistency revisited, Artificial Intelligence 28 (1986) 225–233.
[15] P. Parodi, V. Torre, On the complexity of labeling perspective projections of polyhedral scenes, Artificial Intelligence 70 (1994) 239–276.
[16] 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.
[17] R. Shapira, H. Freeman, The cyclic order property of vertices as an aid in scene analysis, Comm. ACM 22 (6) (1979) 368–375.
[18] M. Straforini, C. Coelha, M. Campani, Extraction of vanishing points from images of indoor and outdoor scenes, Image and Vision Computing 11 (2) (1993) 91–99.
[19] K. Sugihara, Mathematical structures of line drawings of polyhedrons—Towards man-machine communication by means of line drawings, IEEE Trans. Pattern Anal. Machine Intell. 4 (5) (1982) 458–469.
[20] K. Sugihara, A necessary and sufficient condition for a picture to represent a polyhedral scene, IEEE Trans. Pattern Anal. Machine Intell. 6 (5) (1984) 578–586.
[21] K. Sugihara, Machine Interpretation of Line Drawings, MIT Press, Cambridge, MA, 1986.
[22] A. Tai, J. Kittler, M. Petrou, T. Windeatt, Vanishing point detection, Image and Vision Computing 11 (4) (1993) 240–245.
[23] E. Tsang, Foundations of Constraint Satisfaction, Academic Press, New York, 1993.
[24] 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.