Computer-Aided Design 35 (2003) 1235–1248

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Solid model reconstruction from engineering paper drawings using Genetic Algorithms

Ke-Zhang Chen[a,*], Xin-An Feng[b]

[a]*Department of Mechanical Engineering, The University of Hong Kong, Room 7-9, Haking wong Building, Pokfulam Road, Hong Kong, People's Republic of China*

[b]*Faculty of Mechanical Engineering, Dalian University of Technology, Dalian 116024, People's Republic of China*

Received 11 March 2002; received in revised form 19 October 2002; accepted 1 March 2003

Abstract

The wide applications of CAD/CAM technologies have promoted the development and application of many advanced technologies, such as digital manufacturing, rapid prototyping, enterprise resources planning, computer integrated manufacturing, concurrent engineering, virtual reality, and mass customization. These advanced technologies all need to be supported by three-dimensional (3D) solid models of products. Most enterprises, however, still organize their production using 2D paper drawings, which makes difficulties for the application of these advanced technologies. It is needed to convert engineering paper drawings into 3D solid models for applications. Based on holo-extraction of information from paper drawings, this paper develops a systematic 3D reconstruction method, which simulates an experienced human designer's thinking mode in transforming inaccurate outlines with inaccurate projection relationships in 2D drawings into 3D image, with the aid of Genetic Algorithms.

Keywords: Engineering drawing; Solid model reconstruction; Genetic Algorithms; Networks of single closed regions

1. Introduction

Engineering paper drawings have been used as graphic definitions of products for a long time. Currently, most enterprises still use them to organize their productions. With the wide applications of CAD/CAM technologies, many advanced technologies, such as digital manufacturing, rapid prototyping, enterprise resources planning, computer integrated manufacturing, concurrent engineering, virtual reality, and mass customization, have been developed and promoted in applications. These advanced technologies all need to be supported by three-dimensional (3D) solid models of products. The two-dimensional paper drawings have made difficulties for the application of these advanced technologies. It is needed to convert engineering paper drawings into 3D solid models for applications.

In order to convert 2D engineering paper drawings into 3D solid models, the 2D paper drawings are first scanned by an optical scanner and input to a computer in the form of a raster (binary) image. The conversion from the raster image to 3D model goes through two steps: understanding the raster image and 3D reconstruction [1–7]. For 3D reconstruction, two families of methods have been developed: the first, known as the 'fleshing out projections' concept [5], employs a wire-oriented bottom-up approach; the second uses a volume-oriented approach [6]. The former goes through the following processes: (1) search for the circuits on each view, by sorting the edges in trignometric order with respect to the normal to the surface, at each vertex, and by traversing the resulting graph of oriented edges; (2) create a face for each finite circuit and a hole for each infinite circuit; and (3) make up a real solid with a group of faces without violating the Moebius rule [7]. The latter is to decompose each view into several predefined types of sub-views or exterior boundary and interior boundary; to generate their 3D sub-parts; and then to combine them to build the complete object. But these methods were developed with the assumption that the drawings have accurate geometric outlines in each view and accurate orthographic projection relationships between views [1], which can only be drawn by a computer-aided

* Corresponding author. Tel.: +852-2859-2630; fax: +852-2858-5415.
E-mail address: kzchen@hkucc.hku.hk (K.-Z. Chen).

0010-4485/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0010-4485(03)00039-3

drafting system. It is obvious that engineering paper drawings do not meet those standards. Therefore, these two families of methods cannot reconstruct 3D solid models from paper drawings.

Based on holo-extraction of information from paper drawings [8], this paper develops a systematic 3D reconstruction method, which simulates an experienced human designer's thinking mode in transforming inaccurate outlines with inaccurate projection relationships in 2D paper drawings into 3D image, with the aid of Genetic Algorithms (GAs) [9–11].

2. Elements and workflow of the solid reconstruction method

The elements and workflow of the 3D reconstruction method developed is shown in Fig. 1 and illustrated as follows:

(1) Create an image consisting of single closed region of black pixels (SCRs) from a paper drawing:

The 2D paper drawing is first scanned by an optical scanner and input to a computer in the form of raster (binary) image, which is then converted into an image consisting of SCRs by using the method introduced in Ref. [8]. Since the area enclosed by a SCR is very narrow and looks like a line, a SCR is still represented by a line in the subsequent figures for simplification. For example, Fig. 2 shows an image consisting of SCRs for a paper drawing, where there are 10 (F_1–F_{10}), 14 (R_1–R_{14}), and 11 (B_1–B_{11}) SCRs in front, right, and bottom views, respectively. The image consisting of SCRs is different from the image consisting of vectors. In the latter, a straight line or a circle is represented by a vector, no matter how many segments it may be divided into by intersectional points. In the former, each segment is a SCR. For instance, in Fig. 2, F_7 and F_8 are two SCRs but form a vector ($F_7 + F_8$).

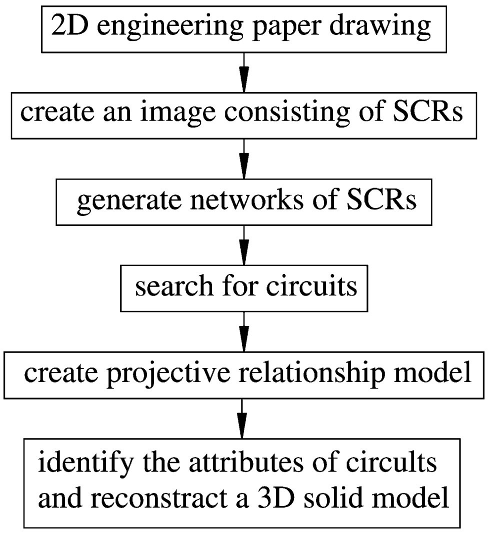


Fig. 1. The elements and workflow of the 3D reconstruction method.

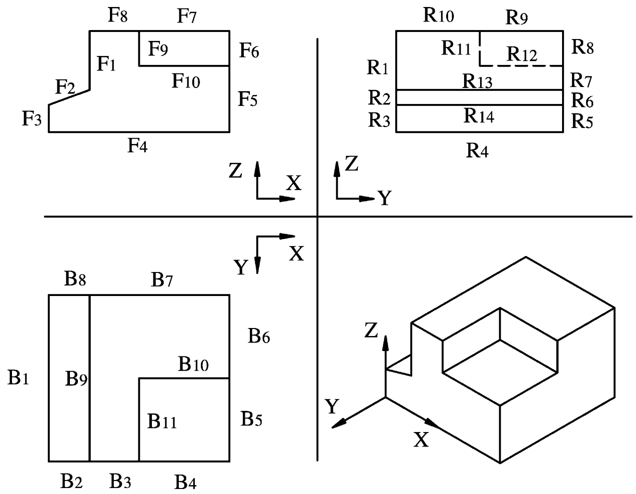


Fig. 2. An image consisting of SCRs for a paper drawing.

(2) Generate networks of SCRs from the image consisting of SCRs:

Using the method introduced in Ref. [8], networks of SCRs can be further constructed from the image consisting of SCRs. For the drawing represented by Fig. 2, three networks of SCRs can be generated, respectively, for the three views, as shown in Fig. 3. In the networks of SCRs, a node represents a SCR, and its node file records the characteristics of the SCR; and a line between two nodes represents the properties of relationship between the SCRs, and its line file records the type of linking point and the angle between them. For the SCR of a straight line, its characteristics are its line width, starting coordinates, end coordinates, length, slope and intercept. For the SCR of an arc, its characters cover its line width, the coordinates of its curvature center, its curvature radius, starting angle to x-axis (let the direction of arc be counter-clockwise.), and end angle to x-axis. Although the linking points can be branching, merging, cross, or tangential points, they can be divided into two categories: the linking point for two SCRs and the linking point for more than two SCRs, represented by 'V' and 'C', respectively. The former provides a path for forming a circuit and the latter offers a path for forming more than one circuit.

(3) Search for circuits or loops according to the networks of SCRs:

According to the standard as shown in Fig. 4 for the first angle drawing system, the name and the origin and axes of local coordinate system are assigned to each view interactively. The SCRs making a small angle with one of the coordinate axes is regarded as being parallel to the axis since the paper drawing is not very accurate. The circuit is a closed path along the SCRs that represent solid lines and/or dashed lines. The chain lines in Fig. 5 show some searched circuits for the views of the part represented in Fig. 2.

(4) Delete false circuits by using a knowledge-based expert system and build projective relationship model by using GAs as shown in Fig. 6.

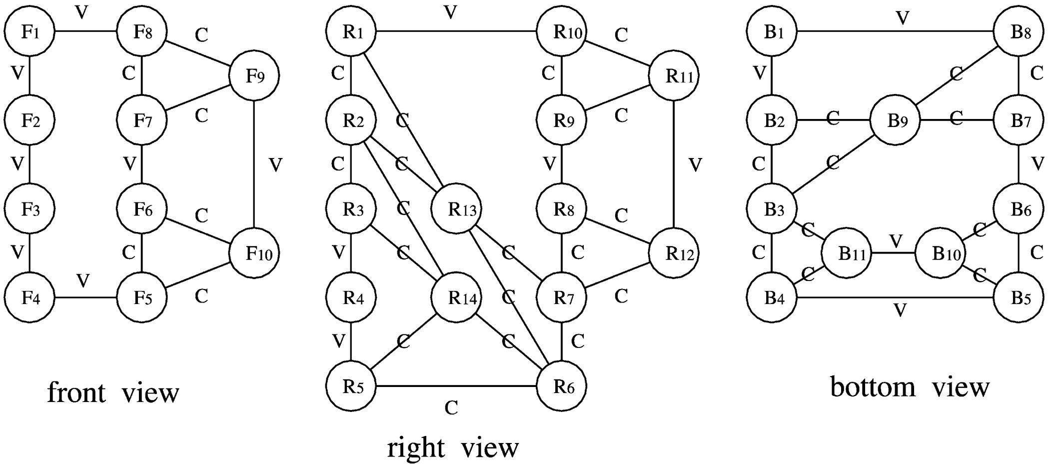


Fig. 3. Three network sets of SCRs generated, respectively, for the three views.

(5) Identify the attribute of each circuit (a face or a hole) by using a knowledge-based expert system and construct a 3D solid model by using B-rep data structure.

Since the first two steps apply the method introduced in Ref. [8] and are clearer, the following sections will introduce the last three steps in more detail.

3. Searching for the circuits for the outlines in views

The search of circuits for the outlines in views is implemented using a rule-based expert system. The rules are as follows:

Rule 1: Coding. Search for circuits, one by one, and assign a name to each circuit. The name is composed of letter ‘L’ and two subscripts. The first subscript represents the view where the circuit is, and includes F, R, B, L, T, and Re that represent front, right, bottom, left, top, and rear views, respectively. The second subscript indicates the serial number of a circuit in a view.

Rule 2: Reaching ‘V’ line. Searching for circuits starts from the first SCR in the network of SCRs for a view and a file numbered i ($i = 1$) is then created for the first circuit. The first SCR will be named as ‘first SCR’ and recorded in the file. From the first SCR, next SCRs will be searched for a circuit in a counter clockwise direction. If the linking point between two adjacent SCRs is ‘V’, the name of the next SCR can be collected into the file of the circuit.

Rule 3: Reaching ‘C’ line. When the linking point searched is ‘C’ (i.e. branching point), one of the SCRs connected with the previous SCR is selected randomly as the next SCR for the current circuit. The name of the next SCR will be collected in the file of the circuit and the names of other SCRs connected with the same linking point will also be recorded in pairs with the previous SCR in a stack.

Rule 4: Stopping criterion for searching a single circuit. When the next SCR searched is first SCR, the search for the current circuit is over. The file of this circuit will be stored in the database of circuits if there is no such a circuit in the database, and the last recorded pair of SCRs will be taken from the stack, so that the branching point with ‘C’ between the two SCRs of the pair becomes the new starting point for a new circuit and another SCR in the pair will be used as the starting SCR for a new circuit, named as first SCR, and recorded in a new file numbered $i + 1$. When there is such a circuit in the database already and this current circuit has branching point with ‘C’, delete the data collected after the branching point, take another SCR connected with the branching point as the next SCR to continue searching for the circuit, and record the next SCRs searched into the file of the current circuit. If there is such a circuit in the database already and the current circuit has no branching point, delete the circuit and its serial number.

Rule 5: Forbidden zone for searching circuits. In a network of SCR, a ‘C’ line always forms a sub-circuit with other ‘C’ lines, which indicates that several lines in a view intersect with each other at the same branching point. For instance, F_5, F_6, and F_{10} in Fig. 3 form a sub-circuit and represent that the three SCRs intersect with each other at a same point. To ensure that the circuit can be closed at first SCR, it is not allowed to take the ‘C’ lines in a sub-circuit as the path of a circuit twice.

Rule 6: Identifying ineffective circuits. When the ‘C’ line in a sub-circuit is taken as a path of a circuit for the second time, take the last recorded branching point from the stack to

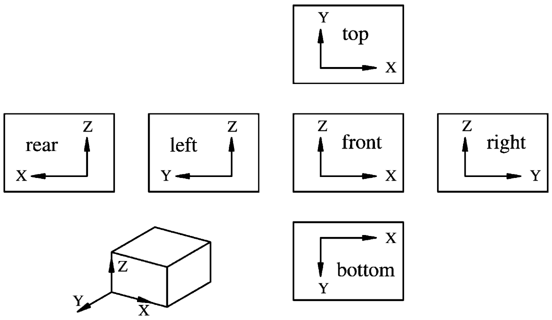


Fig. 4. The local coordinate system of each view of the first angle drawing system.

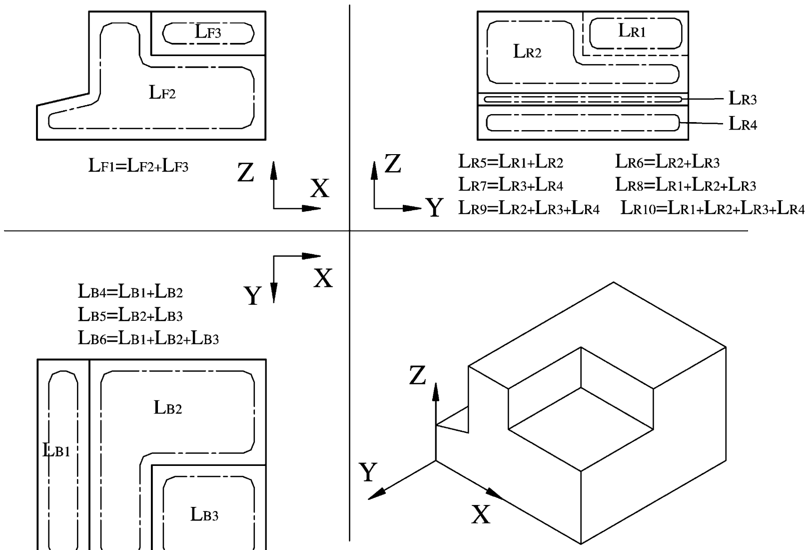


Fig. 5. Some searched circuits.

search for the circuit, delete all the data collected after the name of the branching point from the file of the circuit, and record the next SCR searched into the file of the circuit. If there is no recorded branching point, delete the circuit and its serial number.

Rule 7: Stopping criterion for searching circuits in a view. When there is no branching point stored in the stack, the search process is over, and the database of circuits has collected all the circuits searched.

According to the above rules, as an example, the search process in the first network of SCRs in Fig. 3 for the front view in Fig. 2 is shown in Fig. 7. The search process is started from F_1 that is then named first SCR and recorded in the file of the circuit numbered ‘1’. Then, F_2, F_3, F_4 and F_5 are successively searched following a counter-clockwise direction and recorded in the file since all the lines between two SCRs are ‘V’ (Rule 2). After F_5, a ‘C’ line is met, which indicates that a branching point is searched. In this case, the search for circuit No. 1 is continuous (i.e. F_6 will be selected randomly) and another SCR in the sub-circuit with ‘C’ (i.e. F_{10}) will be stored in pair with F_5 in a stack (Rule 3), as shown in Fig. 7b(1). After F_6 and F_7 are found, another ‘C’ line is met. The search for No. 1 circuit is continuous (i.e. F_8 will be selected randomly) and F_9 will be stored in pair with F_7 into the stack (Rule 3), as shown in Fig. 7b(2). Finally, F_1 (first SCR) will be searched and the searching process for circuit No. 1 is over. The file of circuit No. 1 will then be stored in the database of circuits (Rule 4).

After the first circuit is obtained, the last recorded pair is taken from the stack, so that branching point between F_7 and F_9 is taken as new starting point, F_7 is taken as the new first SCR, and the searching for the second circuit is started from F_7. After F_9 is found, F_8 cannot be taken as the next SCR due to Rule 5 and only F_{10} can be searched as the next SCR. After F_{10}, another ‘C’ line is met, which indicated another branching point is searched. The search for circuit No. 2 is continuous (i.e. F_6 will be selected randomly) and F_5 is stored in pair with F_{10} in the stack as shown in Fig. 7b(3). According to Rule 5, F_7, not F_5, is searched. Since F_7 is first SCR, the search for the second circuit is over. Then circuit No. 2 is compared with the circuit in the database (i.e. circuit No. 1). Since they are different, the file of circuit No. 2 can be stored into the database of circuits.

After the second circuit is obtained, the last recorded pair (i.e. $F_{10}-F_5$ shown in Fig. 7b(3)) is taken from the stack so that the branching point between F_{10} and F_5 is the new starting point, F_{10} is taken as the new first SCR, and the search for the third circuit is started from F_{10}. After F_5 is searched, only F_4 can be taken as the next SCR due to Rule 5. Then, F_3, F_2, F_1, and F_8 are successively searched from F_4

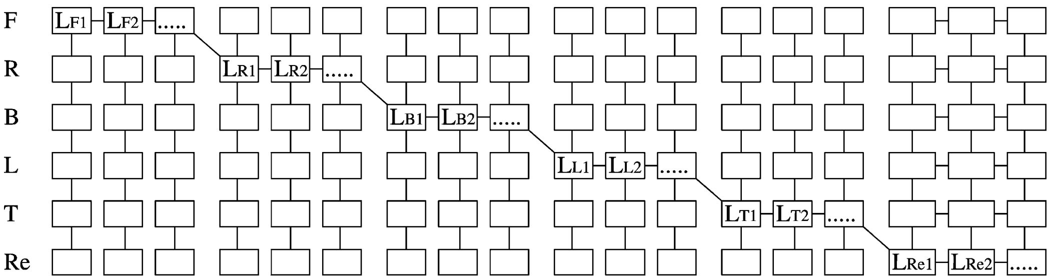


Fig. 6. A projective relationship model.

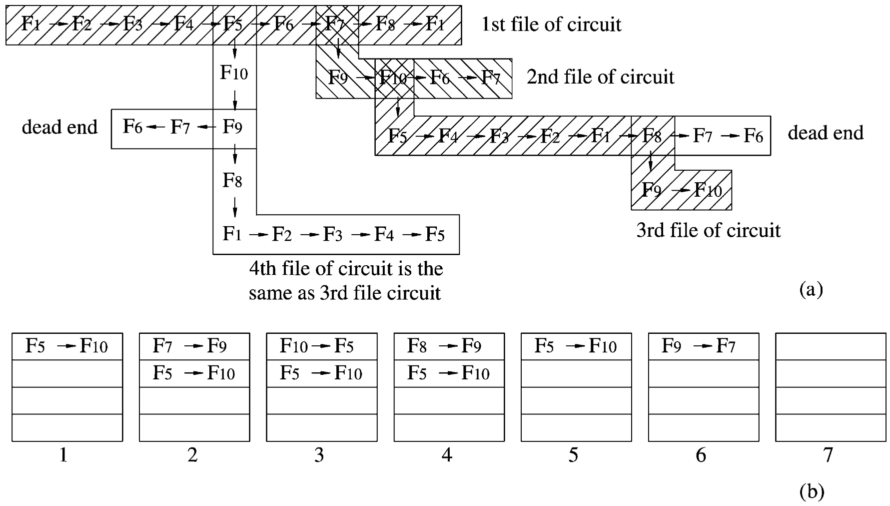


Fig. 7. The circuit search process for the first network of SCRs in Fig. 3.

for the third circuit. After F_8, 'C' lines are met again. If F_7 is taken as the next SCR, F_9 will be recorded in pair with F_8 into the stack as shown in Fig. 7b(4). Then, F_6, not F_9, is searched due to Rule 5. Since the search cannot be continuous to F_{10} or F_5 owing to Rule 5, Rule 6 is implemented. That is, the data collected after the last branching point (i.e. F_6 and F_7) will be deleted from the file of circuit No. 3 and the last recorded SCR pair in the stack (i.e. F_8–F_9) is taken from the stack for search. Then, F_{10}, not F_7, is searched due to Rule 5. Since F_{10} is first SCR, the search process for circuit No. 3 is over. The circuit No. 3 is compared with the circuits in the database (i.e. circuit Nos. 1 and 2). Since they are different, the file of circuit No. 3 is stored into database of circuits.

Now, there is only the pair F_5–F_{10} is left in the stack as shown in Fig. 7b(5). The search process for the fourth circuit can be started from F_5 that is then recorded as first SCR. After F_{10} and F_9 are searched, the search process for circuit No. 4 will be continuous (i.e. F_8 will be obtained randomly) and pair F_9–F_7 will be recorded into the stack as shown in Fig. 7b(6). Then, F_1, F_2, F_3, F_4, and F_5 are successively searched from F_8. Since F_5 is first SCR, the search for circuit No. 4 is over. After circuit No. 4 is compared with the circuits in the database (i.e. circuit Nos. 1–3), it is found that circuit No. 4 is the same as circuit No. 3. Since there is a branching point (i.e. F_9–F_7) recorded in the stack for the circuit, the SCRs collected after the point (i.e. F_8–F_5) will be deleted and the SCR pair in the stack (i.e. F_9–F_7 shown in Fig. 7b(6)) is taken from the stack for new search. Then, F_6, not F_8, is obtained and the search cannot be continued to F_5 or F_{10} both due to Rule 5. Since there is no any SCR pair left in the stack for returning to the first SCR, this circuit and its serial number will be deleted according to Rule 6.

Up to now, the stack is empty as shown in Fig. 7b(7) and the searching process for the front view is over (Rule 7). The database has now stored three circuits (i.e. circuit Nos. 1–3). With the same principle, the circuits for other two views (10 circuits for right view and six circuits for bottom view) can be obtained as shown in Fig. 5. It should be noticed that not all the circuits found are effective. Some false circuits will be identified and deleted in a subsequent process.

4. Projective relationship model and its building rules

A general model for projective relationships of a part in an engineering paper drawing is expressed in Fig. 6. The number of layers is the number of views in a drawing. The layers are arranged in a top–down manner from front (F), to right (R), to bottom (B), to left (L), to top (T), and to rear view (Re). If some views are not used, the lower layer used will be raised to the positions the missing layers are. The crucial path in the model comprises the identified circuits of each view. Each circuit is represented by a small rectangle with the name of circuit in it. Building the model should meet the following rules.

Rule 1: The projective relationship of planes, cylinders, spheres and tori in different views. The projection of a sphere on every view is a circular circuit. The projective relationships of a plane, cylinder, cone, and torus in different views are summarized in Tables 1–4, respectively.

Rule 2: Correlation of a part's projective positions in different views. The straight lines and circuits (or closed loops) in each column of Table 1 have certain position correlations. Based on these correlations, those straight lines and circuits for projections of a plane in different views can be identified. If a plane circuit is perpendicular to *Y*-axis (e.g. L_{F3} in Fig. 5), its projections on right, bottom, left, and top views all are the straight lines perpendicular to *Y*-axis with the same *Y*-coordinate. Its projections on front and rear views are a pair of mirrored circuits and the corresponding vertices of the circuits have the same coordinates. If a plane circuit is parallel to *Y*-axis (e.g. L_{B1} in Fig. 5), its projections on front and rear views all are inclined straight lines and those in other views are circuits. The corresponding vertices

Table 1
Projection relationship of planes in different views

View	Plan						
	Perpendicular to the coordinate axis of			Parallel to the coordinate axis of			Inclined
	X	*Y*	*Z*	*X*	*Y*	*Z*	
Front view	LPA	CL	LPA	CL	ISL	CL	CL
Right view	CL	LPA	LPA	ISL	CL	CL	CL
Bottom view	LPA	LPA	CL	CL	CL	ISL	CL
Left view	CL	LPA	LPA	ISL	CL	CL	CL
Top view	LPA	LPA	CL	CL	CL	ISL	CL
Rear view	LPA	CL	LPA	CL	ISL	CL	CL

Note: LPA—straight line perpendicular to coordinate axis; CL—circuit; ISL—inclined straight line.

of circuits and straight lines have the same coordinates. If a plane circuit is neither perpendicular nor parallel to any axis, its projections on all the six standard views are circuits. The corresponding vertices of these circuits have the same coordinates.

The rectangular circuits (or trapezoids) and circular circuits (concentric circles) in each column of Tables 2 and 3 have also certain position correlations. Taking Fig. 8 as an example, the cylindrical surface is parallel to *Z*-axis and its projection in bottom view is a circular circuit that is formed by two superimposed circular circuits. Its projection in front view is a rectangular circuit with a width equal to the diameter of the circular circuit and a height equal to the length of the cylinder. The *X*-coordinate of its center axis is same as the *X*-coordinate of the center point of the circular circuit. Its projection in right view is also a rectangular circuit with the same width and height as those of its front view. The *Y*-coordinate of its center axis is same as the *Y*-coordinate of the center point of the circular circuit on bottom view. The *Z*-coordinates of its top and bottom sides in right view are equal to those in front view, respectively.

Table 2
Projection relationship of cylinders in different views

View	The axis of cylinder			
	Parallel to the coordinate axis of			Inclined
	X	*Y*	*Z*	
Front view	RC	CC	RC	Inclined rectangle but its two subtenses are replaced with ellipses
Right view	CC	RC	RC	Same as above
Bottom view	RC	RC	CC	Same as above
Left view	CC	RC	RC	Same as above
Top view	RC	RC	CC	Same as above
Rear view	RC	CC	RC	Same as above

Note: RC—rectangular circuit; CC—circular circuit.

Table 3
Projection relationship of cones in different views

View	The axis of cone			
	Parallel to the coordinate axis of			Inclined to any axis
	X	*Y*	*Z*	
Front view	TC	TCCC	TC	Inclined trapezoid but its top and bottom edges are replaced with ellipses
Right view	TCCC	TC	TC	Same as above
Bottom view	TC	TC	TCCC	Same as above
Left view	TCCC	TC	TC	Same as above
Top view	TC	TC	TCCC	Same as above
Rear view	TC	TCCC	TC	Same as above

Note: TC—trapezoidal circuit; TCCC—two concentric circular circuits.

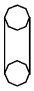
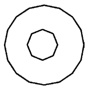
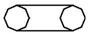
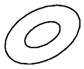
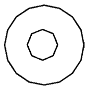
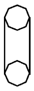
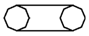
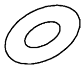
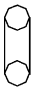
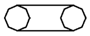
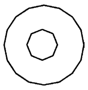
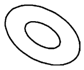
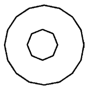
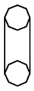
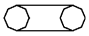
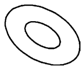
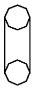
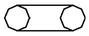
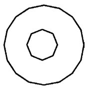
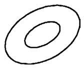
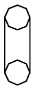
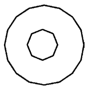
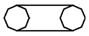
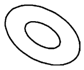
For the torus in Table 4, the centers of its projections in all the views have the same coordinates, and the external and internal diameters of its projections in all the views should be equal to each other.

Rule 3: Correlations of a part's projective topologies in different views. The projection of a plane circuit parallel or perpendicular to a coordinate axis in another view is a straight line. If there are n points in the straight line, the straight line consists of $(n-1)$ SCRs. For example, circuit L_{F2} in Fig. 5 has eight vertices and its projection in right view is a straight line with five projective points. The straight line thus consists of four SCRs $(5-1=4)$, R_8, R_7, R_6, and R_5, as shown in Fig. 2. The end points of the four SCRs have correlation with the five projective points. If the projection of a circuit in another view is also a circuit, the two circuits should have same number of vertices, and corresponding vertices have position correlations. If the circuit is a circle, there is no other vertex or split point since a circle always consists of two SCRs for two half cycles after constructing a network of SCRs, as shown in Fig. 8. Therefore, the rules for it should be different from those for other circuits and is based on the center of circle. For example, the circular circuit in bottom view consists of two SCRs, B_1 and B_4, and the coordinates of its center point is (x_0, y_0, z_0). The projections of the circle in front or right views are two horizontal lines with the length equal to the diameter of the circle. The coordinates of their middle points should be (x_0, y_0).

Rule 4: Superimposed circuits. A circuit can be formed by superimposing multiple circuits with the same shape. For instance, L_{F1} of the front view in Fig. 5 are superimposed by two circuits. In its projective relation model, they should both be indicated with one more subscript: (1) or (2).

Rule 5: Deleting the ineffective circuits. The ineffective circuit is the circuit consisting of the lines on different planes. For instance, L_{R7} and L_{R6} in Fig. 5 are composed of

Table 4
Projective relationship between views for tori

View name	Axis of tori			
	Parallel to the coordinate of			Not parallel to any coordinates
	X	*Y*	*Z*	
Front view				
Right view				
Bottom view				
Left view				
Top view				
Rear view				

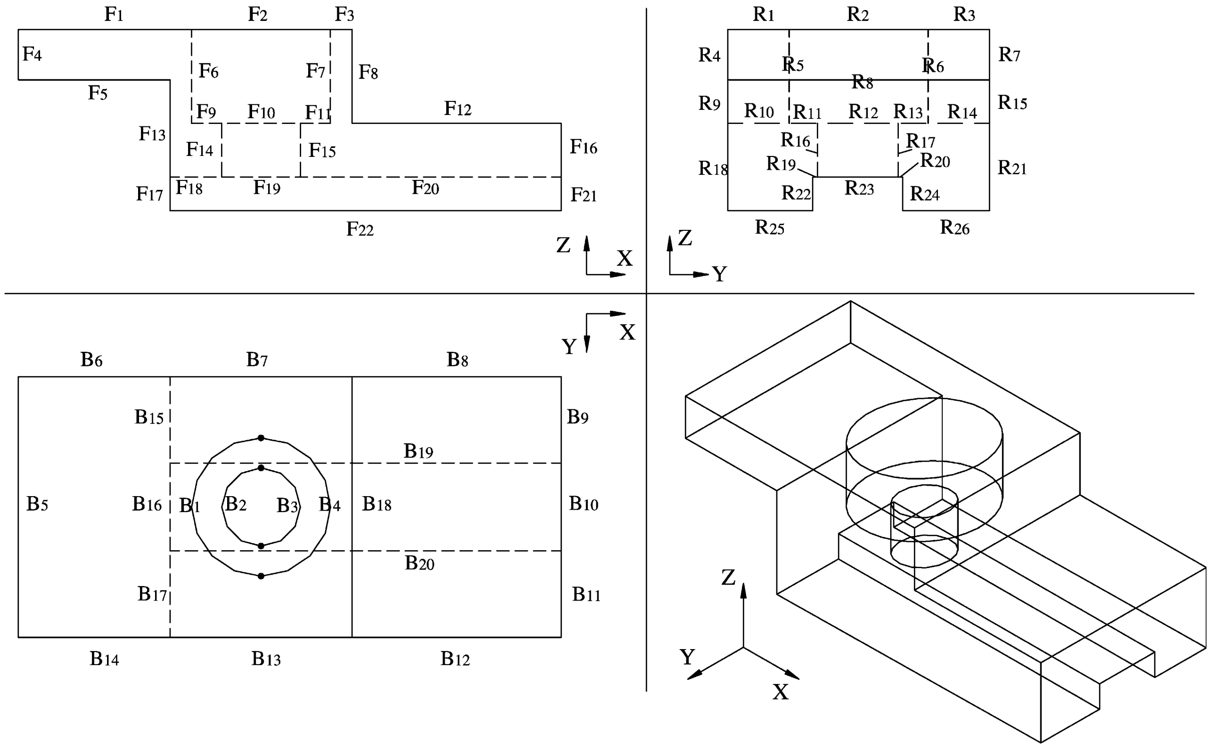


Fig. 8. The image consisting of SCRs for another part.

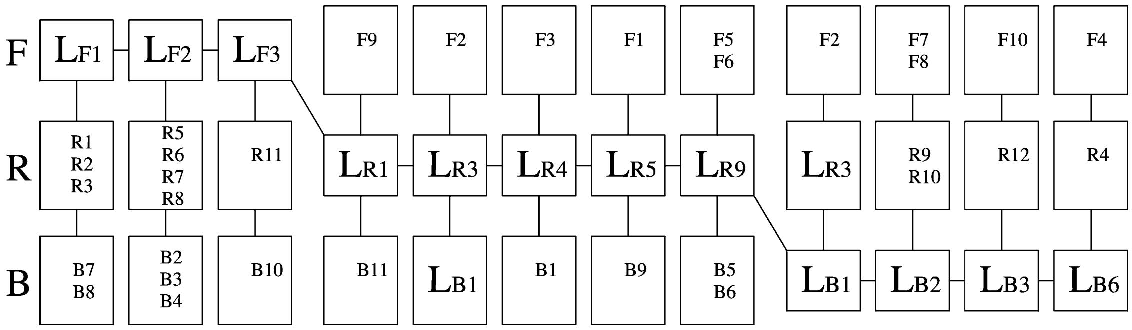


Fig. 9. The projective relation model for the part represented in Fig. 3.

lines in different planes and belong to the ineffective circuits. The ineffective circuits also possibly include those consisting of dashed lines partly or wholly. L_{R1} and L_{R2} in Fig. 5, for instance, also belong to ineffective circuits. The approach for deleting ineffective circuit is to delete the circuit if the line or circuit in other views that has projective position correlation and projective topological correlation with the circuit cannot be found. For instance, the six vertices of L_{R6} in Fig. 5 should have three corresponding projective points in front or bottom view. But in front or bottom view, no straight line consisting of two SCRs $(3 - 1)$ can be found along which there are three projective points having projective position correlation and projective topological correlation with the six vertices. Thus, L_{R6} should be deleted. Based on this rule, L_{R2}, L_{R6}, L_{R7}, L_{R8} and L_{R10} on right view and L_{B4} and L_{B5} on bottom view all belong to ineffective circuits and should be deleted.

According to the above rules, a projective relation model for the part represented in Fig. 2 can be built as shown in Fig. 9. In the model, there are three layers, and there are three, five, and four effective circuits on the F, R, and B layers, respectively. The projections of each circuit in other views are annotated in corresponding rectangles on corresponding layers. For instance, the projection of L_{F2} in right view is a straight line consisting of four SCRs: R_5, R_6, R_7, and R_8 that are recorded in the second small rectangle of R layer. Its projection in bottom view is a straight line consisting of three SCRs: R_2, R_3, and R_4 that are recorded in the second rectangle of B layer.

5. Generating the projective relation model

5.1. Uncertainty of projective relationships

The uncertainty of projective relationship is caused by the following factors:

(1) Inaccurate projective relationship in paper drawing:

If a 2D drawing is absolutely accurate, the projective relations among views can be easily determined using reasoning and ineffective circuits can be deleted according to the above rules. But engineering paper drawings are made by human beings and always have inaccurate outlines with inaccurate projective relationship. The raster images scanned from paper drawings and the SCRs generated subsequently will be more inaccurate. Therefore, the coordinates of their vertices or end points are not accurate. Thus, it is necessary to collect the vertices or end points of SCRs in a certain tolerance region for the analysis of projective position correlation and topological correlation, which causes the uncertainty of projective position relationship since other irrelative points will be taken into account.

(2) Uncertainty of projective topological correlation:

A circuit can be generated by projecting an orthogonal plane, inclined plane, cylindrical face, conical face, spherical face, or toroidal face. It is possible to have several projective schemes in other views, which satisfy projective position correlation and topological correlation with the circuit. This causes the uncertainty of projective topological correlation.

(3) Uncertainty of projective superimposition:

As mentioned in Section 4, the projections of the above different faces or several same faces can be superimposed with each other. It is possible for a circuit to represent multiple projections, which causes the uncertainty of projective superimposition.

(4) Uncertainty of the length of projective relationship model:

It is possible that some of the circuits obtained after search are not effective and must be deleted. This causes the uncertainty of the length of projective relationship model. The key to generate a correct projective relation model is how to find out and delete ineffective circuits under these uncertainties and identify the types of effective circuits.

5.2. Generating projective relation model with uncertainties

Since there is uncertainty of projective relations, it is possible to have several possible projective schemes in other views for a circuit searched. In the beginning, a projective relation model with uncertainties can be generated. The approach for it is as follows:

(1) Analyze each circuit and generate two coordinate sets for each circuit:

If there are n vertices in a circuit and two coordinates in the local coordinate system of the view are A and B, respectively, the coordinates of each vertex can be expressed as (a_i, b_i), $i = 1, 2, 3, \ldots, n$. Thus, based on the two coordinate directions, two sets of coordinates can be formed as:

$$(a_1, a_2, \ldots, a_K) \in (a_1, a_2, \ldots, a_n)$$

$$(b_1, b_2, \ldots, b_L) \in (b_1, b_2, \ldots, b_n)$$

where the lengths of sets, K and L, are smaller than or equal to n since it is possible for these vertices to have same A or B coordinates.

(2) Determine each element in the sub-set of a polygon circuit:

Since a polygon circuit can be obtained by projecting a plane, cylindrical face or conical face, its corresponding projections in other views (i.e. the elements in its sub-set) can be straight lines, circular, or polygon circuits. If one of the coordinate directions (say A) of a local coordinate system in the view (say bottom view), where the element is, is the same as one of the coordinate directions of a local coordinate system in the view (say front view) where the circuit is, search in bottom view for the orthogonal lines, inclined lines, or circuits with K or more vertices or splitting points. The A-coordinates of these points should be in the ranges: $(a_1 \pm \Delta), (a_2 \pm \Delta), \ldots, (a_K \pm \Delta)$, respectively, where Δ is a specified deviation and is normally 1–5 mm according to the size of drawing. If the two coordinate directions in the view, where the element is, are the same as those in the view where the circuit is, search in the view where the element is for the circuits with n or more vertices or splitting points that are in the ranges: $((a_i \pm \Delta), (b_i \pm \Delta), i = 1, 2, \ldots, n)$.

It is possible to obtain more than one projective schemes after the above search. All the schemes should be recorded into corresponding element in a database of projective relation models.

(3) Determine each element in the sub-set of a circular circuit:

If the circuit is circular, it can be the projection of cylinder, cone, or sphere. Its radius (R_0) and the coordinates (A_0, B_0) of its center point are first determined. If there is a coordinate direction (say A) in the view, where the circuit is, is the same as one coordinate direction in the view where the element is, it can be known from Table 2 that the projection of the circular circuit in the view where the element is can be a rectangle or circle. If it is a rectangle, the circular circuit is the projection of a cylinder and is formed by the superimposed projections of two end faces of the cylinder. In this case, this circuit should be decomposed into two circuits in the projective relation model. Then, search in the view where element is for a straight line with two or more than two vertices or splitting points. The two end points of a straight line can be in the ranges: $(A_0 + R_0 \pm \Delta)$ and $(A_0 - R_0 \pm \Delta)$, respectively. If it is a circle, the center point can be in the range $(R_0 \pm \Delta)$.

If the circuits are two concentric circles, they can be the projections of the concentric cylindrical surfaces, conical surface, or toroidal surface. It can be known from Tables 2–4 that their corresponding projections in other views are different. When it is from the concentric cylindrical surfaces, their projection in another view is a rectangle. If it is from a conical surface, its corresponding projection in another view is a trapezoid. When it is from toroidal surface, its corresponding projection in another view is a circuit consisting of arcs and lines.

It is possible to obtain more than one projective schemes after the above search. All the schemes should be recorded into corresponding element in a database of projective relation models.

5.3. Search for the correct projective scheme using Genetic Algorithms

When the experienced human designers read an engineering paper drawing, they will try to find all the corresponding projective relationships for all the geometric elements in different views by trial and error. If successful, the 3D image can be generated correctly in their brains. Otherwise, they will continue to try using the trial and error method until they are successful. The GAs method [10] can be, in fact, corresponding to this trial and error searching method. We applied GAs to simulate this human thinking mode to find the correct projective scheme (i.e. global optimum) and have got success.

Taking the circuits in Fig. 9 as the example, its projective relation model has three layers (F, R, and B) that have three, five, and four circuits, respectively, and has total 12 columns. The crucial path in the model is a chromosome according to GAs. Each circuit with its sub-sets is a gene in the chromosome.

The circuits searched can be the plan parallel or perpendicular to coordinate axis; inclined plan; cylindrical, conical, or toroidal surface parallel or not parallel to coordinate axis; or their combinations (the superimposed projection of same or different types of surfaces).

In fact, more than one corresponding projections in another view or schemes can be searched from a circuit. If there are n_i $(i = 1, 2, \ldots, k)$ schemes searched, the domain space of genes has n_g schemes

$$n_g = \prod_{i=1}^{k} n_i \quad (1)$$

where k is number of views minus 1.

Based on the projective relation model with uncertainty, each scheme is checked using the rules introduced in Section 4. If the scheme is not feasible, this scheme will be deleted. When there is no feasible scheme, this circuit is not feasible and will be deleted.

Each scheme in the domain space of genes represents a projective scheme. The fitness of a scheme is a function of projective error

$$F_i^{(j)} = \frac{n_j}{\sqrt{\sum_{m=1}^{n_j} \Delta_{i,m}^2}} \tag{2}$$

where $F_i^{(j)}$ is the fitness of the ith scheme in the jth gene, n_j is the number of projective point pairs of the ith scheme in the jth gene, and $\Delta_{i,m}$ is the projective error of the mth projective point pair of the ith scheme.

During evolution process, the probability of selecting a scheme from the domain space as a gene should be the function of the fitness of the scheme. The larger its fitness is, the higher the probability of selecting this scheme will be.

If there are $n_g^{(i)}$ scheme in the ith gene of a chromosome, the total population (TP) can be calculated by $\prod_{i=1}^{n} n_g^{(i)}$. The size of the population (L) should have less chromosomes and is determined according to the TP by

$$L = \begin{cases} \text{TP}, & \text{if TP} < 20 \\ 0.2 \times \text{TP}, & \text{if TP} \geq 20 \text{ and } L > 20 \\ 20, & \text{if TP} \geq 20 \text{ and } L \leq 20 \end{cases} \tag{3}$$

The first generation of the population is selected ramdonly from the initial population using a roulette wheel [10] constructed based on their fitness obtained from Eq. (2).

After the first generation is generated, the crossover, mutation and reproduction operations will be repeated until the best chromosome is obtained. Since genetic operation is implemented randomly, its fitness is not always increased continuously. It is not correct to stop the optimization process when there is no increment or improvement for the fitness after a new generation is obtained. Therefore, a threshold is used for the number of generations that have the same best chromosomes. That is, if $n_i - n^*$ is greater than a threshold $q = 20$, the iteration process can be stopped, where n_i is the current generation number and n^* denotes the generation number when the best chromosome, among all generations, is first found. Then the best chromosome can be output as the optimal solution. Once the types of these circuits are identified, the projective scheme of each circuit in each view can be determined. Thus, the projective relation model is created.

6. Attributes of circuits and their identifying axioms

6.1. Attributes of circuits

The attributes of circuits include (1) the circuit represents a face or a hole, and (2) which segment of circuitis represents an external surface if it is a face. The former can be represented by the normal vectors of segments of the circuit. As shown in Fig. 10(a) the circuit with outward normal vectors is a face, and the circuit with inward vectors is a hole. The latter can be indicated by the normal vector of the face, which is specified to point outwards. In 3D solid model, the normal vector of a face is determined according to the right-hand rule based on the order of storing the sides (E_1, E_2, E_3, and E_4) as shown in Fig. 10(b).

6.2. Axioms identifying the attributes

Axiom 1. If a segment of a circuit is an intersectional line between faces, there is no need to define the direction of normal vector for the circuit. In Fig. 11(a), the plane or face represented by circuit L_2 is higher than those represented by L_1 and L_3. The segment (1–2) and (2–3) of L_1 is the intersectional lines between the face represented by L_1 and the two faces represented by the two segments of L_2. The segments (3–4) and (4–1) of L_3 is the intersectional lines between the face represented by L_3 and the two faces represented by the two segments of

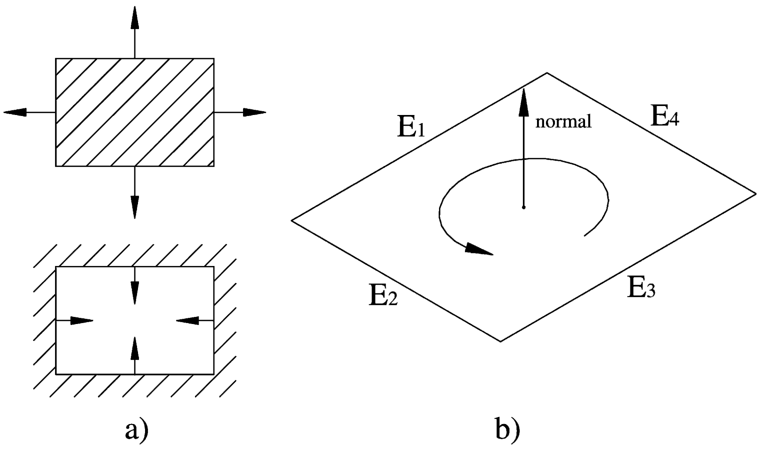


Fig. 10. The attributes of circuits.

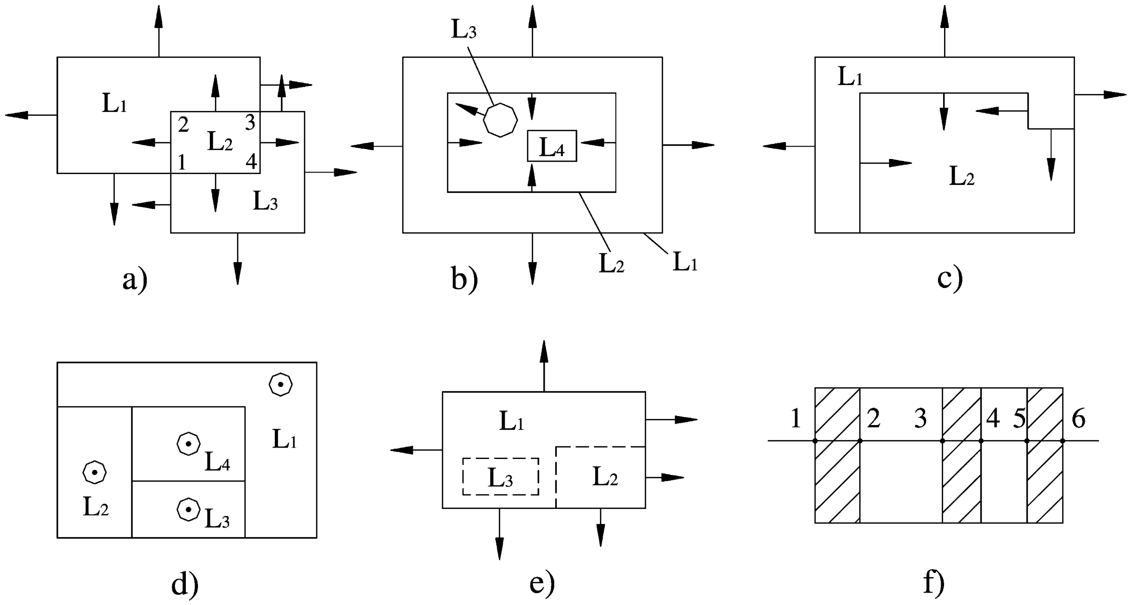


Fig. 11. Identifying the attributes of circuits.

L_2. Therefore, there is no need to specify the normal vector direction for the circuit L_2.

Axiom 2. If a segment of a circuit is the projection of a face, it is necessary to determine its normal vector direction that belongs to both the face and the segment. The normal vector directions of all the segments representing faces' projections in a circuit are the same, i.e. outward or inward.

Axiom 3. If all the segments of a circuit are the most external outlines in a view (e.g. L_1 in Fig. 11(b)), this circuit represents a face and the normal vectors of the circuit are outward. The normal vectors of segments of the circuit are also outward. The segments can be the the projections of the faces with outward normal vector or the intersectional lines between the face with outward normal vector and the face represented by the circuit.

Axiom 4. If partial segments of a circuit are the most external outlines in a view (e.g. L_1 in Fig. 11(c)), this circuit must represent a face, and the normal vectors of both the face and the most external segments of circuit are outward. If the face represented by L_1 is higher than that by L_2, the normal vectors of internal segments are also outward.

Axiom 5. If the smaller circuits are inside the larger circuits and these circuits are at the same height (e.g. L_1, L_2, and L_3 in Fig. 11(b)), the normal vector of segments of the largest circuit are outward, those of smaller circuit are inward, and those of the smallest circuits are outward again, on the analogy of this. When they are not at the same height and the inside circuit is higher than the outside one, the inside one must be the projection of a face. If the inside circuit is lower than the outside one, the inside one must be the projection of a hole. Whether it is a through hole or not is determined according to other views.

Axiom 6. If the circuit consisting of thick lines has at least one segment that is the most external (e.g. L_1, L_2, and L_3 in Fig. 11(d)), this circuit must be the projection of a face.

Axiom 7. If the circuit consisting of thick lines is the projection of a face (e.g. L_4 in Fig. 11(d)), the normal vector of this face must be upward, even if it is at a different height from other planes or faces.

Axiom 8. If the circuit consists of dashed lines partly or wholly (e.g. L_2 and L_3 in Fig. 11(e)), the normal vector directions of its dashed line segments cannot be determined. Whether it represents a face or a hole cannot also be determined without other views.

Axiom 9 [12]. If a straight line intersects with a 3D solid (as shown in Fig. 11(f)), the number of their intersectional points must be even number. The part between a pair of intersectional points starting from odd number (e.g. 1–2, 3–4, and 5–6) must be a solid part. The part between a pair of intersectional points starting from even number (e.g. 2–3 and 4–5) must be empty.

7. Process of identifying the attributes of circuits

The process of identifying the attributes of circuits is introduced by using the example shown in Fig. 8 that is a converted SCRs image from a raster image. Applying the above method, the circuits in front view are searched as follows:

$L_{F1}(2)$: $F_1–F_2–F_3–F_8–F_{12}–F_{16}–F_{21}–F_{22}–F_{17}–F_{13}–F_5–F_4–F_1$
$L_{F2}(2)$: $F_{18}–F_{19}–F_{20}–F_{21}–F_{22}–F_{17}–F_{18}$
L_{F3}: $F_2–F_7–F_{11}–F_{10}–F_9–F_6–F_2$
L_{F4}: $F_{10}–F_{15}–F_{19}–F_{14}–F_{10}$

where the number in brackets indicates the number of layers this circuit appears. The circuits in right view are obtained as:

L_{R1}: $R_1-R_2-R_3-R_7-R_8-R_4-R_1$
L_{R2}: $R_8-R_{15}-R_{21}-R_{26}-R_{24}-R_{20}-R_{23}-R_{19}-R_{22}-R_{25}$
$-R_{18}-R_9-R_8$
L_{R3}: $R_1-R_2-R_3-R_7-R_{15}-R_{14}-R_{13}-R_{12}-R_{11}-R_{10}$
$-R_9-R_4-R_1$
L_{R4}: $R_{10}-R_{11}-R_{12}-R_{13}-R_{14}-R_{21}-R_{26}-R_{24}-R_{20}$
$-R_{23}-R_{19}-R_{22}-R_{25}-R_{18}-R_{10}$
L_{R5}: $R_2-R_6-R_{13}-R_{12}-R_{11}-R_5-R_2$
L_{R6}: $R_{12}-R_{17}-R_{23}-R_{16}-R_{12}$

The circuits searched in bottom view are as follows:

L_{B1}: $B_5-B_6-B_7-B_{18}-R_{13}-R_{14}-R_5$
L_{B2}: $B_5-B_6-B_{15}-B_{16}-R_{17}-R_{14}-R_5$
L_{B3}: $B_8-B_9-B_{10}-B_{11}-R_{12}-R_{18}-R_8$
L_{B4}: $B_7-B_8-B_9-B_{19}-R_{15}-R_7$
L_{B5}: $B_{20}-B_{11}-B_{12}-B_{13}-R_{17}-R_{20}$
L_{B6}: $B_{19}-B_{10}-B_{20}-B_{16}-R_{19}$
L_{B7}: $B_1-B_4-B_1$
L_{B8}: $B_2-B_3-B_2$

The projective relation model is created as shown in Fig. 12.

Its process of identifying the attributes of circuits is illustrated as follows:

(1) From circuits of front view:

According to Axiom 3, L_{F1} in front view can be identified to represent two faces. The normal vectors of the two faces are outward. The normal vectors of all the segments of the circuit are also outward. It can be known from its projective relation model (Fig. 12) that the segments, (F_1, F_2, F_3), F_5, F_{12}, F_{22} and F_{22}, correspond to the circuits in bottom view: L_{B1}, L_{B2}, L_{B3}, L_{B4}, and L_{B5}, respectively. The segments, F_4, (F_{13}, F_{17}), F_8, and (F_{16} and F_{21}), correspond to the circuits in right view: L_{R1}, L_{R2}, L_{R3}, and L_{R4}, respectively. These circuits are all the projections of faces and their normal vectors are outward.

According to Axiom 7, since circuits L_{F2}, L_{F3}, and L_{F4} have dashed line segments, their attributes are uncertain and can be identfied with further information from other views.

(2) From circuits on right view:

In right view, the attributes of circuits L_{R1}, L_{R2}, L_{R3}, and L_{R4} have been determined from the front view. According to Axiom 4, the normal vectors of segments (R_{22}, R_{23} and R_{24}) of circuit L_{R2} must be outward. It can be known from projective relation model that two circuits L_{F2} represent faces, and its normal vector points to the symmetrical plane. Circuit L_{B6} represents a face and its normal vector is outward. The rest circuits, L_{R5} and L_{R6}, have dashed line segments, and their attributes are uncertain and can be identfied with further information from other views.

(3) From circuits on bottom view:

In bottom view, the attributes of circuits L_{B1}–L_{B6} have been determined from front view and right view. Circuit L_{B7} has two projections in both front and right view, respectively. One is at the same height as circuit L_{B1} and another one is at the same height as L_{B8}. According to Axiom 5, the normal vectors of all the segments of the former circuit L_{B7} are inward, and this circuit represents a hole. Its corresponding circuits L_{F3} and L_{R5} in front and right view have dashed line segments and the normal vectors of these dashed line segments are inward. The latter is a blind hole and the normal vector of the bottom of the hole is upward according to Axiom 6. Circuit L_{B8} represents a through hole since its projection on right view is R_{20} that is the most external outline.

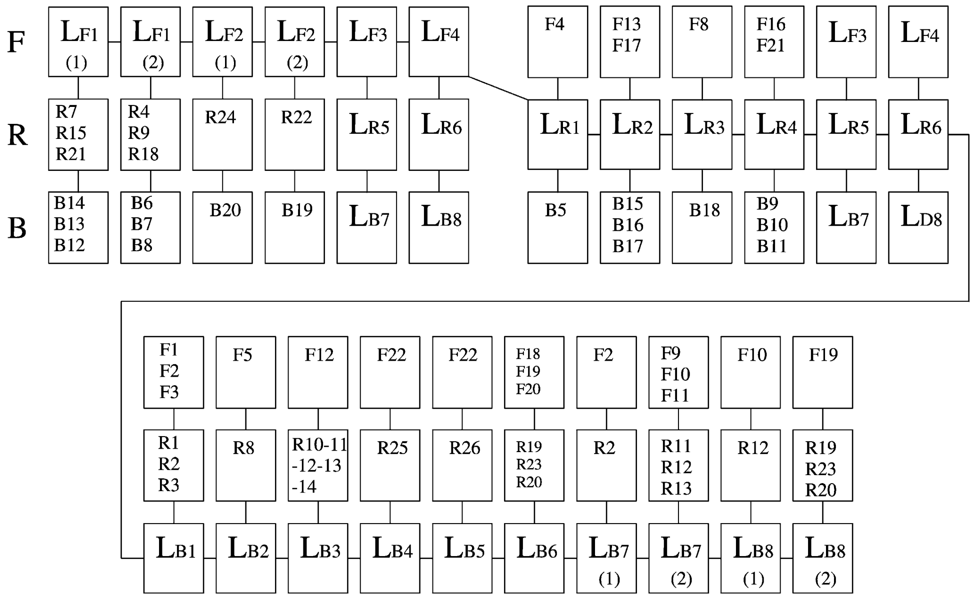


Fig. 12. The projective relation model for the part represented in Fig. 8.

8. Creating a solid model based on B-Rep data structure

(1) Assign the attributes about normal vectors to the circuit representing faces:

This is implemented by arranging the segments of the circuit based on right-hand rule and by combining the related straight lines and/or arcs into vectors. The circuits in front view are converted into the following six faces:

L_{F1-1}: (F_3, F_2, F_1)−F_4−F_5−(F_{13}−F_{17})−F_{22}−(F_{21}, F_{16}) −F_{12}−F_8−(F_3, F_2, F_1)
L_{F1-2}: (F_1, F_2, F_3)−F_8−F_{12}−(F_{16}−F_{21})−F_{22}−(F_{17}, F_{13}) −F_5−F_4−(F_1, F_2, F_3)
L_{F2-1}: (F_{18}, F_{19}, F_{20})−F_{17}−F_{22}−F_{21}−(F_{20}, F_{19}, F_{18})
L_{F2-2}: (F_{20}, F_{19}, F_{18})−F_{21}−F_{22}−F_{17}−(F_{18}, F_{19}, F_{20})
L_{F3}: F_2−F_7−(F_{11}, F_{10}, F_9)−F_6−F_2
L_{F4}: F_{10}−F_{15}−F_{19}−F_{14}−F_{10}

where single SCR and a set of SCRs in brackets are the vectors. The circuits identified in right view are:

L_{R1}: (R_3, R_2, R_1)−R_4−R_8−R_7−(R_3, R_2, R_1)
L_{R2}: R_8−(R_9, R_{18})−R_{25}−R_{22}−(R_{19}, R_{23}−R_{20})−R_{24} −R_{26}−(R_{21}, R_{15})−R_8
L_{R3}: (R_1, R_2, R_3)−(R_7, R_{15})−(R_{14}, R_{13}, R_{12}, R_{11}, R_{10}) −(R_9, R_4)−(R_1, R_2, R_3)
L_{R4}: (R_{10}, R_{11}, R_{12}, R_{13}, R_{14})−R_{21}−R_{26}−R_{24}−(R_{20} −R_{23}−R_{19})−R_{22}−R_{25}−R_{18}−(R_{10}, R_{11}, R_{12}, R_{13}, R_{14})
L_{R5}: R_2−R_6−(R_{13}, R_{12}, R_{11})−R_5−R_2
L_{R6}: R_{12}−R_{17}−R_{23}−R_{16}−R_{12}

The circuits identified in bottom view are:

L_{B1}: B_5−B_6−B_7−B_{18}−R_{13}−R_{14}−R_5
L_{B2}: B_5−B_6−B_{15}−B_{16}−R_{17}−R_{14}−R_5
L_{B3}: B_8−B_9−B_{10}−B_{11}−R_{12}−R_{18}−R_8
L_{B4}: B_7−B_8−B_9−B_{19}−R_{15}−R_7
L_{B5}: B_{20}−B_{11}−B_{12}−B_{13}−R_{17}−R_{20}
L_{B6}: B_{19}−B_{10}−B_{20}−B_{16}−R_{19}
L_{B7}: B_1−B_4−B_1
L_{B8}: B_2−B_3−B_2

(2) Converting the circuits into the face-edge set of solid model [12]:

The method for this conversion is to merge the circuits representing the faces on a same plane into a face and combine the circuits representing the faces with the same meaning into a face. As shown in Fig. 13, those circuits representing the faces on a same plane are L_{B1} and L_{B7-1}, L_{B7-2} and L_{B8-1}, L_{B6} and L_{B8-2}; and those representing faces with the same meaning are L_{F3} and L_{R5}, L_{F4} and L_{R6}.

(3) Creating the table for edge−vertices versus vertex-coordinates:

The table for edge-vertices and vertex-coordinates is specified as follows:

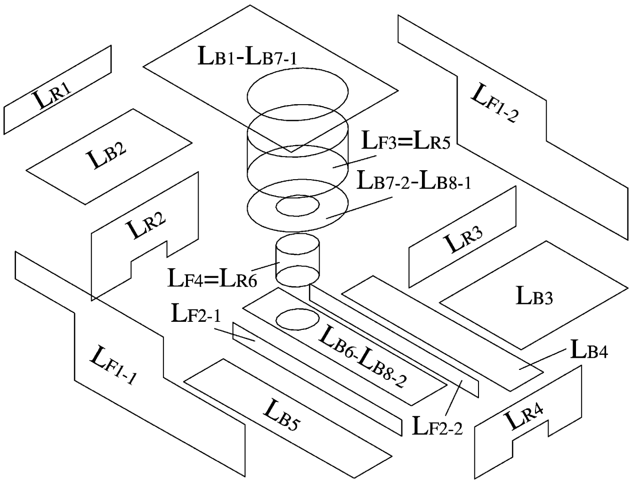


Fig. 13. Converting the circuits into the face-edge set of solid model.

Edge	Vertices
e_1	v_1, v_2
e_2	v_2, v_3
e_3	v_3, v_4
⋮	⋮

Vertices	Coordinates
v_1	x_1, y_1, z_1
v_2	x_2, y_2, z_2
v_3	x_3, y_3, z_3
⋮	⋮

where edge is the segment of the circuit. If the segment consists of one SCR, the two end points of the edge are the end points of the SCR. When the sigment of the circuit is composed of a set of SCRs, the two end points of the edge are the starting point of the first SCR and the ending point of the last SCR in the set of SCRs.

9. Conclusions

Based on holo-extraction of information from paper drawings, this paper develops a systematic 3D reconstruction method, which allows paper drawings to have inaccurate outlines and inaccurate projective relationships among views and simulates an experienced human designer's thinking mode in transforming inaccurate 2D outlines with inaccurate projection relationships in paper drawings into 3D image. This method has been verified and can be used for 3D reconstruction of the components consisting of planes, cylindrical faces, conical faces and toroidal faces that appear on most mechanical parts. This also provides a base for the 3D reconstruction of those components having arbitrarily curved surfaces.

Acknowledgements

The reported research is supported by University Research Committee Grants (CRCG) in the University of Hong Kong. The financial contribution is gratefully acknowledged.

References

[1] Dori D, Tombre K. From engineering drawings to 3D CAD models: are we ready now? Comput-Aided Des 1995;27(4): 243–54.
[2] Wan W, Ventura JA. Segmentation of planar curves into straight-line segments and elliptical arcs. Graph Models Image Process 1997; 59(6):484–94.
[3] Teh CH, Chin RT. On the detection of dominant points on digital curves. IEEE Trans Pattern Anal Mach Intell 1989;11(8): 859–72.
[4] Rosin PL, West GAW. Nonparametric segmentation of curves into various representations. IEEE Trans Pattern Anal Mach Intell 1995; 17(12):1140–53.
[5] Preiss K. Constructing the solid representation from engineering projections. Comput Graph 1984;8(4):381–9.
[6] Yoshiura H, Fujimura K, Kumii L. Top–down construction of 3-D mechanical object shapes from engineering drawings. IEEE Comput Mag 1984;17(12):32–40.
[7] Ah-Soon C, Tombre K. A step towards reconstruction of 3-D CAD models from engineering drawings. Proceedings of the Third International Conference on Document Analysis and Recognition, Montreal, Canada; 1995. p. 331–4.
[8] Chen KZ, Zhang XW, Ou ZY, Feng XA. Holo-extraction of information from paper drawings for 3D reconstruction. Comput-Aided Des 2002;34(9):665–77.
[9] Gen M, Cheng R. Genetic Algorithms and engineering design. New York: Wiley; 1997.
[10] Chen KZ, Feng XA. Computer-aided design method for the components made of heterogeneous materials. Comput-Aided Des 2003;35(5):453–66.
[11] Chen KZ, Zhang XW, Ou ZY, Feng XA. Recognition of digital curves scanned from paper drawings using Genetic Algorithms. Pattern Recog 2003;36(1):123–30.
[12] Mantyla M. An introduction to solid modeling. Rockville, MD: Computer Science Press; 1988.

Ke-Zhang Chen received his PhD degree from the University of Manchester (UMIST), UK in 1984. After he had worked at the University of Manchester (UMIST) in UK, Donghua University in China, MIT in the USA, and the University of California at Davis in USA, he joined the Department of Mechanical Engineering in the University of Hong Kong in 1996. His current research interests include design theory and methodology, concurrent engineering, intelligent CAD, 3D reconstruction, evolutionary design, and design for heterogeneous components.

Xin-An Feng received his BS degree from the Department of Mechanical Engineering, Dalian University of Technology, China, in 1956. He has since then been working in the university, and is now a professor, deputy director of CIMS Research Center, standing council member of China Association for Mechatronics Technology and Applications, and council member of China Institution for Production Engineering. From 1980 to 1982, he worked as a visiting scholar in the University of Manchester (UMIST), UK. His current research interests include CAD/CAM, intelligent CAD, concurrent engineering, and CIMS.