Journal for Geometry and Graphics
Volume 2 (1998), No. 1, 45–53

Revision of Inconsistent Orthographic Views

Takashi Watanabe

School of Informatics and Sciences,
Nagoya University, Nagoya 464-8601, Japan
email: watanabe@info.human.nagoya-u.ac.jp

Abstract. Many algorithms for the generation of solid objects from two-dimensional (2D) orthographic views have bottom-up procedures which generate 3D segments, form 3D faces and construct polyhedrons. When they accept a set of consistent view drawings, they present at least one solid object. However, when the input views are not consistent, contradictions appear during the process in each step of the bottom-up approach. We examine incoherences found in each step of the bottom-up approach in these cases. Then we introduce a method which reveals the sources of incoherences in input views and suggests consistent views. Three kind of sources are considered: extra segments, improper designations of line types (visible lines or hidden lines) and missing segments. Extra segments and improper designations of line types are found by projecting the solid object constructed from subset of 2D segments onto view planes. Probable missing 3D segments are generated not from three views but from two views among three views. Some heuristic rules are proposed in order to select more probable missing segments and make the process efficient. The present method can be used to detect operator's mistakes in input views or misunderstandings of drawing recognition systems.

Key Words: solid modeling, orthographic views, inconsistent views, solid estimation, human-machine interface

1. Introduction

The development of computer system has revealed the requirement of a common understanding between users and computers. One of the realizations of the common understanding will be found in intelligent user interfaces which guess what their users are going to do, present helpful advice to the users and correct the user's mistakes.

Orthographic projections present a useful way to represent three-dimensional (3D) objects and the common understanding of views is beneficial to generate geometric models interactively. Some studies, therefore, have been conducted in order to build algorithms used to generate solid objects from view drawings. These methods expect complete input views and

ISSN 1433-8157/$ 2.50

they may fail to construct solid objects when the views are inconsistent. On the other hand, when we exchanges information on solid objects shown in views, we can usually understand what the views present even though the views include some inconsistencies. In this paper, we focus our attention on the inconsistent views and present a computer aided method which predicts the objects intended by designers and specifies erroneous 2D segments in views.

Bottom-up procedures are generally used in the algorithms for the generation of solid objects. They generate 3D vertices/edges from 2D views and form 3D faces by finding closed face-loops. Then they solve consistent labeling problems of selection of vertices, candidate vertex sequences and the sets of faces constructing solid objects [1] [2]. We also adopt a bottom-up approach which formulates the certainties or uncertainties of the presences of 3D segments and faces in boolean equations and solves consistent labeling problems. This algorithm is used only to generate solid objects. Many bottom-up approaches including the one used here do not present any solid objects for inconsistent input views. Therefore we introduce a new method to estimate complete 3D solid objects even when input views have incoherences.

2. Bottom-up approach of solid modeling

Though the bottom-up algorithms construct three-dimensional segments, faces and subobjects, many of these geometric or solid object elements are not used to construct final objects. This means that the geometric elements handled in each step of the algorithm are probable constituents of solid objects.

The first step of our algorithm is the formation of probable 3D segments from vertices and segments in 2D views. The 3D segments intersecting with another 3D segment are disconnected at the intersecting points and new two segments are created, and then the disjointed segment is deleted in order to remove overlapping segments. The next step is a selection of a set of segments on a plane. A probable surface is generated by tracing, from a 3D vertex, the leftmost or rightmost 3D segments on the plane and finding a closed loop. A surface included in another surface S determines a hole in the surface S. The third step is the construction of probable solid objects from surfaces and 3D segments. The line types of projections of 3D segments are not designated in this step. In the final step, we test the projections of probable solid objects onto the view planes against the original views, and we confirm that the projected segments meet the conditions for line types of segments in the original views.

We examine the incoherences appearing in the steps of bottom-up algorithms, and present a method to specify and correct inconsistencies. It is assumed that solid objects do not have any curved surfaces and no curved segment appears in views. The input orthographic views are written in numerical values and they are not image data.

3. Failure in the construction process of solid objects

The bottom-up approaches present erroneous graphical elements during the construction process of solid objects from inconsistent views. In this section, we introduce some of these cases [3].

When a 2D segment or an end view of a line in one view has no corresponding 2D segment with same coordinate values for the shared coordinate axis in another view, the former 2D segment is not used to generate a 3D segment. This case also happens when a view includes an extra segment or when a requisite segment does not belong to a view.

In case that a 3D segment has an endpoint to which no other segment connects, some 2D segments do not form any surfaces even though the segments are used to generate probable 3D segments. This case appears when there is an extra segment or a missing segment in a view. For example, an extra segment in a view generates a 3D segment, if it happens to have corresponding segments and/or end views of edges in other views.

When a 2D segment is missing in a view, a 3D segment is not generated as it should be, and another 3D segment which is originally designed to connect to the missing 3D segment has an endpoint and shows this incoherence.

Any solid object may not be constructed from a set of probable surfaces, when missing segment in views prevents to generate some of surfaces, or when an extra segment accidentally generates 3D segments and ineligible surfaces.

There may exit some 2D segments which are not used to construct solid objects, even though they are used to generate 3D segments and surfaces. When an extra 2D segment appears in input views, 3D segments and surfaces formed from the extra segment are determined as false and they are not used in 3D object generation. This case also arises if one view includes a missing 2D segment and some segments in other views generate false 3D segments and surfaces.

While some probable solid object are constructed, there may be no object which presents correct projections of visible lines or hidden lines in input views. When a 3D segment is obscured by some surface(s), the projection of the segment gives a hidden line. If there is no surface nearer to the view plane than a 3D segment, the projection of the segment gives a visible line. When a line in a view is incorrectly designated as a visible line or a hidden line, this case arises.

The above cases include three types of causes: extra 2D segment, missing 2D segment and improper designation of line types (solid lines or hidden lines).

Consistent view drawings do not always construct only one solid object, but it sometimes gives multiple solutions. In this study, these views with multiple solutions, as well as views with unique solutions are considered to be consistent, and the problems of multiple solutions is outside the scope.

4. Extra two-dimensional segments and missing 2D segments

Even if some extra 2D segments are combined with consistent input views, bottom-up algorithms assure us that the probable solid objects are constructed from subsets of 3D surfaces generated by input views. This means that the solid object intended by its designer is among the probable objects. In this case, extra segments in original views are detected as 2D segments which are not projections of 3D segments constructing the solid object.

Most of the bottom-up algorithms test the conditions for solid lines and hidden lines after constructing solid objects or subobjects. This means that probable objects are constructed from segments without designations of line types, and then the subobjects are projected on view planes to check the conditions for visible or hidden lines. For this reason, the improper designation of a visible line or hidden line can be determined by comparing the original views and the projections of each probable polyhedron on view planes.

In case of missing segments in views, we need some method to supply them. The missing segments may be found by a top-down way which assumes the intended solid object by using some heuristics and projects it on view planes. In this paper, we present a new method which generates 3D segments not from three views but from two views and select reasonable segments which may be regarded as missing segments.

The approach to the construction of solid objects from two views is similar to that from three views: probable 3D vertices and segments are generated by finding segments and/or vertices with the same shared coordinate values in multiple views. The principles of two-view projection are based on Descriptive Geometry [4] which allows a reversible transformation between the description of an object in 3D space and that in multiple 2D planes.

We estimate the intended solid object by constructing it from the subset of union of two sets: the set of 3D segments generated from three views and that obtained from arbitrary combinations of two views among the three views. Once a solid object is constructed, the segments missing in the original views can be suggested for the object. The resolution of incoherences in views is achieved in the processes of selection and addition of missing segments, construction of solid objects, and reprojection of the object on the view planes.

When we use two views among three views, the constraints on the construction of solid objects are relaxed and they give a larger number of 3D segments and probable solid objects than three views do. It is our experience that only a small fraction of the segments generated from two views are used to construct final solid objects and most of the segments are false. Considering that the construction of solid objects from 3D segments is a combinatorial problem, we need to select segments in some reasonable way to solve the problem efficiently. We, therefore, introduce some heuristics for the selection of segments.

5. Selection of more probable missing segments

The 3D probable missing segments are those which are generated not from three views but from arbitrary combinations of two views out of the original three views. When no segment connects with a probable missing segment, the segment is omitted because it is not used in the construction of solid objects.

The projection of missing 3D segment on view planes gives the probable missing 2D segment, and the geometric or solid object elements which are not generated from the original three views are obtained from the probable missing 3D segments formed from two views. The bottom-up algorithm of construction of solid objects restarts from the step of the generation of 3D segments including probable missing 3D segments. In this process, all 3D segments are temporarily regarded as visible lines until the condition of line types is tested.

We expect the solid object in the designer's mind is among the objects constructed by taking account of all the segments generated from three views and two views. Once a solid object is obtained, the missing segments, extra segments and improper designations of line types are found by projecting the solid object onto the view planes.

In order to efficiently solve the combinatorial problem, we introduce seven methods to select more probable missing segments and supply missing segments step by step. In the following, we refer the originally given views to as original input views, and we refer the views with supplemented missing segments to as input views.

In the first method, no segment is selected and the input views are the original input views. The solid objects constructed from the segments selected by this method show extra 2D segments and improper designations of line types in the original input views.

When a 2D segment has an endpoint, the probable missing 3D segment whose projection gives a 2D segment connecting to the former 2D segment at the endpoint is selected by the second method. This is because the 2D segment connecting to no segment at its endpoint does not contribute to the generation of surfaces.

As mentioned in section 3, some inconsistencies happen because there are 2D segments not used to generate 3D segments nor 3D surfaces. In the third method, the probable missing 3D segments whose projections onto view planes relieve these incoherences are selected. A segment in a view may cause incoherences because segments or vertices with the same shared coordinate value in another view are not found. This selection method is useful for the case that no solid object is constructed from probable surfaces.

Even though probable solid objects are constructed used in order to test the conditions for line types, there may be some 2D segments which generate only the 3D segments and do not contribute to the construction of solid objects. In this case, the probable missing 3D segments generated from these 2D segments are selected by the fourth method. Our experience tells that many of the probable missing segments selected by this method intersect with each other and most of them are false and it may be inefficient to select all of the intersecting segments. We therefore use all segments selected by this method when less than half of the probable missing segments intersect. Otherwise we use only non-intersecting segments.

The fifth method selects all the probable missing segments other than intersecting segments, and the sixth method selects all the probable missing segments.

In order to specify the inconsistencies in the input views, we use a step-by-step way in which we add probable missing 3D segments selected by each selection method and construct solid objects.

The supplement of missing segments does not always gives more probable solid objects. Therefore following ways are used to detect and reject improbable objects.

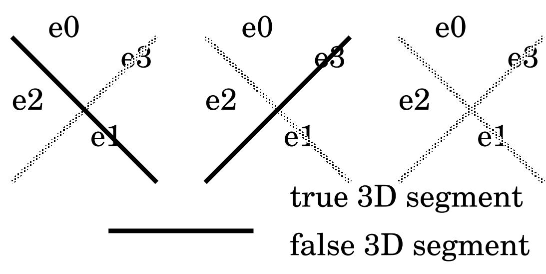

Figure 1: Intersecting 3D segments

As can be seen in Fig. 1, none of the probable missing 3D segments which intersect at 3D points are true, except that there coincidentally exists another 3D segment which passes the intersecting point and does not lie on 2D planes spanned by the probable missing 3D segments. Therefore the combinations of the probable missing 3D segments which do not form an angle of 180° and do not make straight lines at the intersecting points are rejected.

Let m_i be the maximum number of incoherences appearing in the solid objects constructed during the process using the i-th selection method. In the process using $(i+1)$-th selection method, objects with more probable missing segments than m_i are abandoned. This is because the solid objects with an excessive number of supplied segments are considered to be dissimilar to the objects intended by designers.

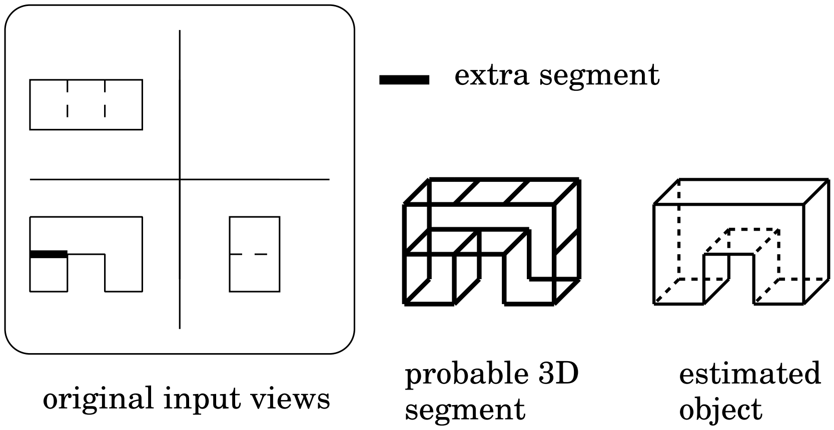

Figure 2: Inconsistent views with extra 2D segment

6. Construction of solid object with supplied segments

The case that the 3D geometric or solid object element generated from the probable missing segments is false is more probable than the case that the geometric or solid object element generated from the original input views is false. While the algorithm after supplement of missing segments is not much different from the algorithm described in section 2, we therefore modify it as follows:

In each step of the selection, the missing segment which are not chosen by the selection method are disregarded during the construction process of solid object.

The construction of surfaces start not from 2D segments but from 3D segments generated from the original input views and probable missing 3D segments.

Constraints on the construction of solid objects are used first for the graphical element generated from supplied segments, and then they are used for the segments given by the original input views.

The above modifications makes the bottom-up algorithm of reconstruction of solid objects more efficient.

Consistent input views give solid objects which are regarded as those intended by designers. If the original input views are inconsistent and consistent views are obtained by removing extra segments and by supplying missing segments, we conclude that the views modified by present methods and their solid objects present ideal solutions. This is not the case for every input views and we may fail to obtain ideal solutions even in the final selection step. Therefore we need criteria to halt the process and decide that reasonable result is found. One of these condition is that the number of incoherences must be less than a certain fraction of the number of segments in input views. Otherwise the type of incoherences must be limited to one among extra segment, missing segment and improper designation of line types. Another condition is that the number of incoherences of each type must be less than a certain fraction of the number of input 2D segments. The final solution depends on the condition we used. In this paper, we halt the process when the type of all incoherences is the missing segment.

7. Example

In this section, we show an example obtained with the program.

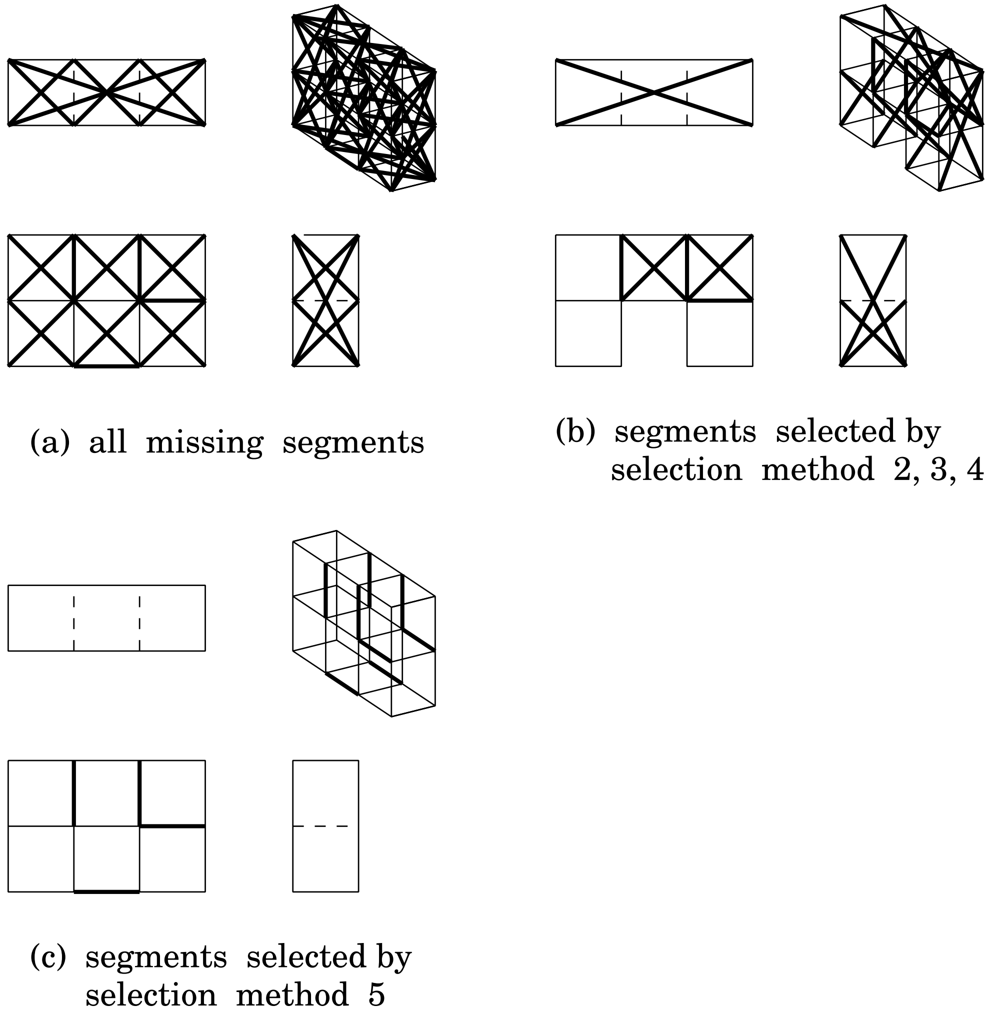

Figure 3: Resolution process of input view in Fig. 2

The inconsistent views in Fig. 2 include one extra 2D segment in the front view. Although 3D segments and surfaces are generated from these views, the extra 2D segment makes some of them false in the step of 3D object generation.

The resolution process in this case is shown in Fig. 3. Here, figure (a) shows the input views including all probable missing segments, figure (b) shows the views with probable missing segments selected by the second, third and fourth selection methods, and figure (c) shows the views with probable missing segments selected by the fifth selection method. Because the probable missing 2D segments in input views are determined by projecting the probable missing 3D segments, the segments in views do not always determine the 3D segments uniquely, whereas the 3D segments in the axonometric drawing have projections on view planes.

Some of solutions obtained from the input views are shown in Fig. 4. In this figure, dotted lines denote extra segments in the original input views and heavy solid lines denote missing segments. The first solution (`Sol_No.1`) gives the intended solid object, the second (`Sol_No.2`) shows three extra segments in the front views and it present an unexpected solid object. In this example, no solution whose incoherences are due only to missing segments is generated, even though all of the probable missing segments are supplied.

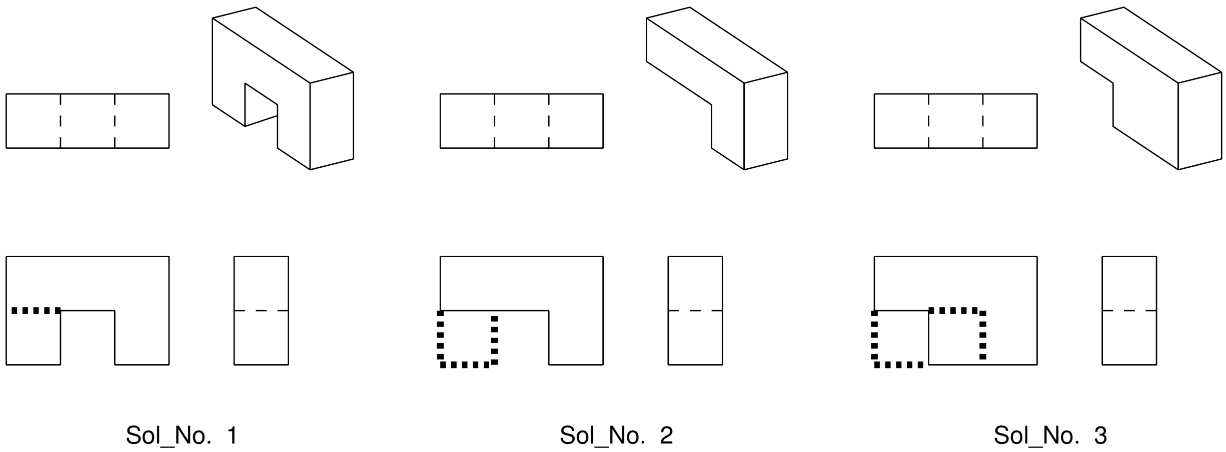

Figure 4: Estimated solid objects from input view in Fig. 2.

8. Discussion

The construction of solid objects gives a combinatorial problem and we can expect the solution process of the problem is easier if the number of graphical elements becomes smaller. In the above example, the number of all the 3D missing segments generated from two views is 143 and the number of 3D segments selected by the second, third and fourth selection methods is 84, and the number of 3D segments selected by the fifth method is 46. Another example shows the number of all the 3D missing segments is 75 and the number of 3D segments selected by the second, third and fourth selection methods is 34, and the number of 3D segments selected by the fifth method is 28. Considering that the simple bottom-up approach requires the number of operations proportional to the factorial of the number of graphical elements, the selection methods introduced in this paper are expected to save the computational resources very well.

In the computer aided geometric modeling, orthographic views present one of the convenient ways to input solid object data. The input work done by human operators may not be complete, and inadmissible views such as given in the example may often be created unintentionally. The method presented in this paper is helpful to reveal the inconsistencies and prompt the operators to fix them.

We have been developing a recognition system which understands handwritten mechanical drawings and presents their CAD data [5]. The system examines the local and global relationships for 2D segments in views, determines various types of lines, and reads dimension numerals and other symbols. However the system sometimes yields inconsistent results because it does not always confirm that the views represent solid objects and they do not include any incoherences. The present method in this paper is originally developed to overcome these difficulties. Assuming that the result presented by the recognition system gives inconsistent views in Fig. 2, the extra segments in the front view may be a part of a extension line in the original drawing. When the recognition system knows that the line is extra and it finds another connecting line in the original drawing, the recognition system may be able to restart its process on the assumption that the extra segment is an extension line.

9. Conclusion

We have examined the incoherences which appear at each step of the bottom-up algorithm for the construction of solid objects. Then we developed a new method which detects the sources of incoherences in input views and estimates solid objects intended by the designer. It resolves three kind of sources: extra segments, improper designations of line types and missing segments. While extra segments and improper designations of line types are determined by the projection of constructed solid objects on view planes, the probable missing 3D segments are generated from two views and they are selected by heuristic methods in order to make the combinatorial problems efficient.

The method we have presented will be useful to construct an intelligent human interface of solid modeler which estimates uncertainties of input drawings and suggests how to fix them to users. It will also be helpful to detect operator's mistakes in input views or misunderstandings of drawing recognition systems.

References

[1] K. Gu, Z. Tang, J. Sun: *Reconstruction of 3D Objects from Orthographic Projections*. Computer Graphics Forum **5**(4), 317–323 (1986).

[2] U.G. Gujar, I.V. Nagendra: *Construction of 3D Solid Objects from Orthographic Views*. Computer & Graphics **13**(4), 505–512 (1989).

[3] T. Watanabe, Y. Chao: *Settlement of Incoherences in the Recognition Process of Solid Object from Orthographic Views*. Proc. Int. Conf. CADEM'95, 284–289 (1995).

[4] L. Alexander, C. William: *Graphics in Engineering Design*. 3rd ed., John Wiley & Sons, New York 1980.

[5] T. Watanabe, T. Okui, T. Onoki, N. Kato, S. Fujii: *A Study on Recognition and Understanding of Mechanical Drawings (Representation and Rule-Based Understanding of Individual Projection)*. JSME International, Ser. III, **35**(8), 505–511 (1992).

Received November 26, 1996