Identifying features in reconstructing 3D solids from sectional views

Yamei Wen  
School of Software, Tsinghua University, Beijing, China  
Email: wenym06@mails.tsinghua.edu.cn

Hui Zhang  
School of Software, Tsinghua University, Beijing, China  
Email: huizhang@tsinghua.edu.cn

Abstract

Sectional views are widely used in engineering practice due to their clear and concise expression. However, large numbers of entities are missing in sectional views that makes it hard for computer to understand drawings. Feature identification is the key procedure for solids reconstruction from sectional views. Based on the analysis of the default drawing rules appearing in sectional views, a new algorithm is presented to identify and construct elemental objects in the procedure of volume-based 3D solids reconstruction. In this algorithm, a hint-based search strategy with priority principles is developed for identifying elemental features, even some of them do not have complete expression in 2D views. The algorithm is suitable for full sections, partial sections and offset sections, and there is no restriction on the number of sectional views included in one drawing. Experimental results validate our algorithm.

1. Introduction

Although solid models play a central role in modern mechanical design, 2D CAD systems are still commonly used for designing regular-shaped products[1]. Many product designs, including legacy data, are definitively expressed in the form of 2D engineering drawings. Reconstructing 3D solid models from 2D views is needful in MPET(Mechanical and Production Engineering Technology).

In engineering practice, sectional views are more commonly used than three orthographic views. Without sectional views, the hidden part from the projection direction is drawn by dash lines. However, the intricate internal structures make the views too complicated to understand. A sectional view can clearly and simply describe a complicated object by using one or more plans to cut the object and remove the part in front of the observer. There are different types of sectional views to satisfy various kinds of objects. The definitions of types are listed by Dimri[2].

Sectional views have been a staple way to describe product designs. Reconstructing solid models from sectional views is consequently required in practice. However, related algorithms rarely addressed this problem in detail. Wesley and Markowsky[3] are the first researchers who referred to sectional views using their wireframe approach. However, there is no experimental result provided to demonstrate its practicabilities. Aldefeld[4] and Ho[5] considered sectional views in an alternative volume-based method. However, these methods need manual intervention to guide the construction of primitives. Lu[6] presented a reconstruction method based on the understanding of semantic information. Full sections and half sections were discussed. Geng[7] made a combination of the wireframe approach and the volume-based method to handle 2D views including full sectional views. Nevertheless, a sectional view is only regarded as an auxiliary of an orthographic view in the same view direction instead of being handled directly. Dimri[2] proposed an algorithm built upon the volume-based method. The additional onto-formations are recognized from sectional views according to loops and edges in both sectional views and regular views. Nonetheless, the limitation of their method is that it just allows one sectional view appearing in the input views. More recently, Gong and Zhang[1] presented a method to label sectional views from engineering drawings based on evidence theory in preprocess. Their method can handle full sections, offset sections and aligned sections, and multiple-sections in different arrangements were considered. However, it is incompetent for constructing 3D models after pretreatment.

Sectional views can cause omission of plenty of edges in correlative orthographic views, which leads to the fuzziness and polysemy of generating 3D edges. Compared to the wireframe method, it is believed that the volume-based method is more suitable to handle sectional views[2]. This paper describes a feature identification algorithm to handle sectional views in volume-based approach(described by Soni[8] and Dimri[2]). Based on the analysis of a certain amount of real engineering drawings, we summarize the common characteristics and drawing rules in sectional views. Our algorithm starts with the loops in each view. Then, elemental features are recognized from the loops by hint-based search strategy, and the cases that the expression of features is missing in 2D views are discussed. Finally, elemental objects are constructed and assembled into the final solid model. The algorithm can handle full sectional views, partial sectional views, offset sections views as well as regular views, and there is no limitation on the number of views.

In the following of this paper, we first give some defini-
2. Terminology and Definitions

We first introduce some important terminology and definitions that are used in this paper. Some of them are from Dimri[2].

**Definition 1 (Common coordinate axis).** It is the coordinate axis that is common between a pair of views. In Fig.(1), the common coordinate axis between view 1 and view 2 is X-axis.

**Definition 2 (Loop).** A simple closed cycle of edges in a view is defined as a loop. In Fig.(1), the closed cycles \( \{1, 2, 3, 6, 5, 4, 1\} \), \( \{5, 6, 7, 8, 5\} \) and \( \{7, 8, 9, 10, 7\} \) in view 2 are all loops.

**Definition 3 (Matching loops).** A pair of loops \( G_i L \in G_i \) and \( G_j L \in G_j \) are said to be matching loops, if \( \text{project}_A(B(G_i L)) = \text{project}_A(B(G_j L)) \), where \( A \) is the common axis between view \( G_i \) and view \( G_j \), and \( \text{project}_A(B) \) signifies the projected interzone on \( A \)-axis of bounding box \( B \). In Fig(1), the circle loop \( G_1 L_{C_1} \) in view 1 and the loop \( G_2 L_1 \{1, 2, 3, 6, 5, 4, 1\} \) in view 2 are matching loops on their common coordinate axis \( X \). In the case of there are more than two loops, they could be treated as matching loops when each pairs of them are matching loops.

**Definition 4 (Feature).** An elemental object corresponding to a group of matching loops is called a feature. For example in Fig(1), the matching loops \( G_1 L_{C_1} \) in view 1 and \( G_2 L_1 \{1, 2, 3, 6, 5, 4, 1\} \) in view 2 represent a cylinder feature.

3. Volume-based method to reconstruct 3D solids from sectional views

3.1. Overview of reconstruction algorithm

An outline of the procedure of volume-based algorithm[8] to reconstruct 3D solid models from engineering drawings is shown in Fig(2).

The key step of the volume-based algorithm is the identification and construction of elementary solids( This step is called match loops and form onto-formations by Soni and Dimri ). It is not difficult to find a feature when there is a group of loops that are completely matched with each other in the views[8]. Technology of handling missing lines in views is greatly influence the level of practical utilization. This problem has first been touched by Dimri[2], who handled the omission as a problem of degeneration. That is to say, if there is no loop matched with a loop \( l \), the algorithm then finds matching edges or edge of \( l \) and considers them as a group of matching loops. However, the method only focus on a small part of sectional views. In fact, the loop \( l \) might find nothing matched with it in sections commonly. Moreover, both Soni and Dimri used geometric information of loops to generate matching loops without understanding the implicit information expressed by the loops. Our hint-based feature identification algorithm aims at understanding the implicit information of 2D loops and improving the ability of reconstruction algorithm to handle sectional views. The details of our algorithm are described in the next subsection.

The other main steps involved in the reconstruction algorithm are described as follows:

1. **Preprocess.** Engineering drawings are imported in .DFX or .DWG format. The geometric entities are converted...
to planar graphs and symbolic entities are filtered through for further processing. After validating view relation, multiple views including sectional views are folded and cutting planes are generated, as described by Gong[1].

(2) Search loops. After preprocess, we have collected information about vertices and edges, including the geometric parameters and the topological relations. Then we need to decompose each view into loops. The leftmost and rightmost loop-searching method described by Yan[9] is widely used in volume-based reconstruction method[8], [2]. As a matter of practice, we also use Yan’s[9] loop-searching method to form loops in the views.

(3) Sweep operations. The elemental objects corresponding to certain entities in the 3D space might be objects of revolutions or extrusions, and these are constructed by rotational sweep and extruded sweep respectively. It is the fact that some irregularity objects, such as skewed quadric surfaces, can not be constructed by sweep operations directly. Fortunately, these kinds of objects do not happen frequently in sectional views.

(4) Boolean operations. Elemental objects, obtained by revolution and extrusion, are combined by union and difference operations to get the final object. The details are described by Dimri[2].

3.2. Feature identification algorithm

Our main work in this paper is to use our knowledge of sectional views to identify and construct features in the case of incomplete information in sectional views. Some important points about sectional views should be concerned.

(1) Many dashed lines are omitted in views because sectional views normally make the drawings more concise and easier to understand. A sectional view, its matching dashed lines in other views might be discarded. Dashed lines behind the cutting plane are usually omitted in the views as well[2]. In fact, even though those dashed lines are missing in views, the information is still enough to catch the views by human being.

(2) The most common features, such as blind holes, through holes, cylinder bosses and so on, have typical and regular projected profiles in views. It is possible that only one projected profile of a feature is enough to identify what it is. The result is that the projection of the feature in other views can be left out and does not affect the recognition. For example in Fig.(3), the rectangle loop \( l_8 \) \{8, 9, 10, 11, 8\} in view 1 presents a tapped hole. Its projection in three views is indicated by the arrow. It is noticed that a solid circle loop and a rectangle loop are missing in other views. However, the loop \( l_8 \) represents a feature of holes obviously. Therefore, not only dashed lines but also solid lines might be missing when the views show us sufficient and unambiguous information about the objects.

Figure 3. an example of missing entities in sectionas.

As discussed above, matching loops becomes less in sectional views. According to the characteristics of sectional views obtained by observing and practical experience, we present a new algorithm to identify elemental features from sectional views.

Loops are the particular characteristics of features presented in 2D views, and some kinds of loops could be taken as hints to guide the identification for features. By observing a numbers of real engineering drawings, there are three types of hints as follows:

- circle-hint: Circles are typical loops which present the projection of large numbers of common features such as holes, blends and cylinder bosses in sectional views. So a circle hints that there might be a hole, a blend or a cylinder boss contained in the 3D solid.
- polygonal with centerline-hint: A single polygonal loop which has a centerline throughout indicates a revolution feature as usual.
- polygonal without centerline-hint: A polygonal loop without a centerline could indicate a polyhedron feature.

Considering that there exist some differences among the reliability of these hints, it is necessary to get some priority principles. It is sure that a circle implies one or more revolutions, so the circle-hints are assigned with the highest priority. A single loop failed to find its matching loop would get the lowest priority because the incomplete 2D information. Based on these priority principles, the following three steps are involved in our feature identification algorithm.

step1. Take circles as hints to identify revolutions in views.

step2. Take polygonal loops as hints to identify extrusions in views.

step3. The rest of single circles and polygonal loops with centerlines, which have no matching loop, are taken as hints to identify revolutions whose projection in the views are omitted.

The procedures of feature identification in the first two steps are similar. Take the first step as an example and the algorithm is as follows.

INPUT: set of the loops in each views.
OUTPUT: set of elemental object features $S_f$.

(1). Identify features.
for (each view $G_a$) do
  for (every loop $G_aL_i$ in $G_a$) do
    if ($G_aL_i$ is not a circle loop)
      continue;
    else for (each view $G_b$ ($a \neq b$)) do
      find the common coordinate axis $X$ between $G_a$ and $G_b$;
      for (every loop $G_bL_j$ in $G_b$) do
        if ($\text{project}_X(B(G_aL_i)) \neq \text{project}_X(B(G_bL_j))$)
          continue;
        else let the recognition mark $mFlag = 1$;
  for (every loop $G_aL_k$ in $G_a$) do
    if ($\text{project}_Y(B(G_aL_k)) \neq \text{project}_Y(B(G_aL_k))$)
      continue;
    else find the common axis $Z$ between $G_a$ and $G_c$;
    if ($\text{project}_Z(B(G_aL_k)) \neq \text{project}_Z(B(G_aL_k))$)
      continue;
    else if ($\text{project}_X(B(G_aL_k)) \neq \text{project}_X(B(G_aL_k))$)
      continue;
    else let $mFlag = 2$;

(2). Validate features.
for (each view $G_c$ ($c \neq a, c \neq b$)) do
  find the common axis $Y$ between $G_a$ and $G_c$;
  if ($X = Y$)
    continue;
  else for (every loop $G_cL_i$) do
    $l_i$ is added into $S_f$;

(3). Create new features.
if ($mFlag = 1$ and $mFlag = 2$) do
  create a new revolution feature $f_n^r$;
  $f_n^r$ is added into $S_f$;

The parameters of sweep operations can be obtained by 2D geometric information of a group of matching loops. It may be noted that there are at least two loops correspond to a feature generated in step 1 and step 2. However, in the case of step 3, an elemental object is identified by only one loop. Consequently, the information in one 2D view is insufficient to construct a 3D object. Considering the general rules in sectional views, especially in partial sections and offset sections, the holes features are reduplicative and uniform distributed. In other words, there are many same kinds of features contained in one 3D object. Moreover, a single circle indicates a through hole in views ordinarily. For example, Fig 4 shows a drawing of offset sectional view. The cutting line in view 2 crosses four circles which are marked with $F_1$, $F_2$, $F_3$ and $F_4$ respectively. Their profiles are drawn in view 1 with corresponding marks. However, the number of circles in view 2 is far more than four. Fortunately, it is found that the circles in view 2 can be classified into five kinds of features according to their formats (see Fig 5). The circles $C_1$, $C_2$ and $C_3$ shown in view 2 of Fig 4 are all single circles with no profile in view 1. Therefore, they are handled as through holes (see $F_3$ in Fig 5).

As mentioned above, the algorithm to construct objects which identified from step 3 is as follows:

(1). For a single circle $l_i$:
  if (in the case of partial sections and offset sections)
    for (each feature $f_j^l$ identified in step 1) do
      if ($\text{radius}(l_i) = \text{radius}(f_j^l)$ and $\text{center}(l_i) \neq \text{center}(f_j^l)$, where $l_i$ is a circle loop of $f_j^l$, $\text{radius}(l)$ and $\text{center}(l)$ are the radius and center of circle $l$)
        create a new revolution feature $f_j^r$;
        set $f_j^r$ as the same profile and angle of rotation with $l_i$;
        else create a new through holes feature $f_j^t$;
      else (in the other views)
        create a new through holes feature $f_j^t$;

(2). For a single polygonal loop $l_p$:
  if (there is a center line crosses $l_p$ and $l_p$ is symmetrical about the center line)
    create a new revolution feature $f_j^r$;
    set the angle of rotation of $f_j^r$ as $2\pi$;
  else the loop $l_p$ indicates no feature.

4. Implementation and discussion

4.1. Examples

Several examples are provided to demonstrate that various cases can be handled by our algorithm.

<table>
<thead>
<tr>
<th>Features</th>
<th>Circles</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td></td>
<td>4</td>
<td>concentric circles</td>
</tr>
<tr>
<td>$F_2$</td>
<td></td>
<td>3</td>
<td>grouping holes</td>
</tr>
<tr>
<td>$F_3$</td>
<td></td>
<td>1</td>
<td>middling single circle</td>
</tr>
<tr>
<td>$F_4$</td>
<td></td>
<td>10</td>
<td>biggest single circles</td>
</tr>
<tr>
<td>$F_5$</td>
<td></td>
<td>3</td>
<td>smallest single circles</td>
</tr>
</tbody>
</table>

Figure 4. an example of offset sectional views.

Figure 5. the classification of features in Fig 4.
Figure 6. Reconstructing 3D model from full sectional views. (a) the features identified in step 1 from Fig.3(a). (b) the features identified in step 2 from Fig.3(a). (c) the features identified in step 3 from Fig.3(a). (d) 3D solid.

Fig.6 shows the reconstruction of a pedestal with several polyhedrons and curved surfaces. The input drawing is shown in Fig.3, there is one full sectional view included in the three views. The procedure of feature identification is shown step by step. Fig.6(a) shows that there are four features formed in step 1, and one feature identified in step 2 (see Fig.6(b)). In step 3, two revolutions are identified by two polygonal loops (see Fig.6(c)). Fig.6(d) shows the reconstructed object which is obtained by combining these seven elemental objects.

Fig.7 shows the reconstruction of a solid of base plate. Fig.7(a) shows that there are two views in the input drawing, including one partial sectional view with plenty of lines missing. It is obvious that there are only 3 pair of loops marked with arrows in view 2 could find their matching loops in view 1. Fig.7(b) shows the final 3D model which is reconstructed by the existing algorithm. The final 3D model is reconstructed by our algorithm as shown in Fig.7(c). The features with incomplete projection in the sectional views are identified and recovered in the case. Four circles, which marked with '1' in Fig.7(a), are constructed as through holes. The other circles are succeed in finding their kinds.

The existing algorithms of reconstructing 3D object from sectional views allows only one sectional view contained in a drawing. However, in order to express complicated internal structures, several different sectional views are always appearing in one drawing at the same time.

Figure 7. Reconstructing 3D model from partial sections. (a) 2D views. (b) the 3D solid reconstructed by the existing algorithm. (c) the 3D solid reconstructed by our algorithm.

An example of a upper cover is provided in Fig.8 to illustrate the ability of our algorithm. Fig.8(a) shows that there are three views in the input drawing, including one offset section (view 1) and one partial section (view 2). Fig.8(b) is the intermediate solid generated by step 1 and step 2 of our algorithm. The circles indicated by arrows in Fig.8(a) failed to form any features. In step 3, the missing features are identified and the final 3D solid is obtained as shown in Fig.8(c).
4.2. Discussion

The existing algorithms are usually specific to fixed number of input views and the flexibility of the algorithms is limited. However, our algorithm has no limitation on the number of input views. In addition, although we mainly focus on sectional views, our algorithm is compatible with non-sectional views. An example of single view and non-sectional view is shown in Fig.9. It is noticed that the algorithm requires that the objects are revolutions in the case of single view.

Since 2D engineering drawings are often presented in flexible and diverse forms, and engineering semantics bear three notable features: implicit, empirical and customary. 2D engineering drawings imply complicated and changeable information, and it is difficult to use a mathematically strict way or form a strict formal description. Our algorithm mainly suffers from the following limitation:

1. The rules and priority principles described in this paper are obtained from generalized cases, such as the cases of uniform distribution and through holes in sectional views. The algorithm is imperfect to handle any realistic cases.

2. For the algorithm it can form all needful elemental objects but it is unsure that all the generated elemental objects are useful. At present we manually remove some redundant elemental objects instead of using multitudinous back-projection.

3. At present, we obtain elemental objects by extrusion and revolution operations. The axis of elemental objects should parallel to the coordinate axis. And the solids with interacting quadric surfaces are not settled.

5. Conclusion

In this paper, a hint-based feature identification algorithm for handling sectional views is proposed to reconstruct 3D objects from sectional views. The algorithm focuses on the problem of information missing in sectional views and improves the traditional volume-based reconstruction algorithm. The main advantages of the algorithm can be summarized as follows.

1. The algorithm can be adopted for more types of views. Both orthographic views and sectional views, such as full sections, offset sections and partial sections, are handled by the same procedure in the algorithm; moreover, there is no limitation on the number of views. It fits for single view as well as multiple views, and allows several sectional views appear in one drawing.

2. It can identify features in drawings efficiently, even the expression of the features is missing in 2D views.

We will organize our future work on the following aspects: the algorithm based on the other sectional views such as half sections, revolved sections; improvement of the range of 3D objects; utilizing artificial intelligence techniques to meet the requirements of real applications with incomplete information.

Acknowledgments

This work was supported by the 973 Program of China (Grant No. 2004CB719404), the 863 Program of China (Grant No. 2007AA040401) and the National Nature Science Foundation of China (Grant No. 60635020).

References


