www.cadanda.com Computer-Aided Design AND APPLICATIONS

Converting Sectional Views to Three Orthographic Views to Reconstruct 3D models

Fengqing Ding[1,3,4], Hui Zhang[1,3,4] and Yamei Wen[1,2,3,4]

[1]School of Software, Tsinghua University
[2]Tsinghua National Laboratory for Information Science and Technology
[3]Department of Computer Science and Technology, Tsinghua University
[4]Key Laboratory for Information System Security, Ministry of Education
dingfq@gmail.com, huizhang@tsinghua.edu.cn, wenym06@mails.tsinghua.edu.cn

ABSTRACT

Compared with the CSG-based approach, the Brep-based approach has several advantages to construct 3D models from 2D engineering drawings, such as the structure is simpler and the domain of objects that can be handled is wider. However, this approach cannot handle sectional views directly. In this paper, a new method of converting sectional views to three orthographic views is presented. Firstly, the views which have the same projection direction are merged into one view. If the number of views is two, then a new view will be added according to the coordinate relations. Secondly, elements which have been omitted in sectional views are recovered according to the matching information of the existing edges. Finally, the existing Brep-based approach is used to reconstruct the 3D models. The algorithm can handle full sections, broken-out sections, offset sections as well as two orthographic views. The algorithm has been validated by experiments.

Keywords: sectional views, brep-based approach, 3D reconstruction, view recovery.
DOI: 10.3722/cadaps.2011.571-582

1 INTRODUCTION

Since 1970s that Idesawa[1] proposed the first paper about 3D reconstruction of 2D engineering drawings, many researchers have studied on this topic and made significant progress during the past four decades. The existing reconstruction algorithms can be basically classified into two major categories [2, 3]: CSG-based approach and Brep-based approach. The CSG-based approach identifies volumes by matching patterns and then combines these volumes using Boolean operations to obtain the final objects. This approach is limited to volumes of extrusion [4,5] and revolution [6,7]. The Brep-based approach constructs a solid model by generating 3D vertices followed by 3D edges and then forming a wireframe model[8-11]. This approach was first formalized by Wesley and Markowsky[8]. Gong et. al [11] extended its domain of objects from polyhedral[8,9] to quadric surfaces by proposing a new concept of hybrid wire-frame model. And the processing time is drastically reduced. Compared with the CSG-based method, the Brep-based method has one important advantage in the wider object domain that can be reconstructed [11].

Sectional views are widely used in engineering practices. However, only a handful of studies are related to sectional views due to their flexible and diverse representations. Aldefeld and Richter [12] and Ho[13] proposed interactive methods to handle sectional views. Geng et. al [14] made a combination of the Brep-based approach and the CSG-based method to handle 2D views including full sections. Dimri and Gurumoorthy[15] used the matching relations between loops and edges to identify volumes if some edges are missing in sectional views. Wen et. al[16, 17] presented an algorithm to reconstruct 3D models from 2D sectional views using the CSG-based method. The domain of objects was extended to inclined and intersecting quadric surfaces. Moreover, the algorithm is able to capture the implicit information of 2D loops and recover repetitive features with incomplete projections in sectional views. Gong et. al[18] presented a preprocessing algorithm to label sectional views from engineering drawings based on evidence theory. However, the problem of reconstructing 3D models after preprocessing was not solved.

By far, most of the researches about reconstruction of sectional views are based on the CSG-based method and the domain of objects is limited to simple extrusions and revolutions. It is difficult to reconstruct 3D models from sectional views using the Brep-based approach, since there are many edges omitted and the number of views is not always equal to three. In this paper, we aim to overcome the difficulties of using the Brep-based approach to handle sectional views. A new algorithm is proposed to recover omitted edges and convert sectional views to three orthographic views. After recovering, three orthographic views are used to reconstruct 3D models by means of the existing Brep-based algorithms. The innovation point of this paper embodies that the types of views that can be handled by the Brep-based algorithms is extended to sectional views and two views.

The rest of the paper is organized as follows. A brief overview of the characteristics of engineering drawings with sectional views is presented in Section 2, with some definitions that are used in our algorithm. The view recovery algorithm is introduced in Section 3. Experimental results and the discussions are presented in Section 4. Section 5 is the conclusion and future work.

2 CHARACTERISTICS OF ENGINEERING DRAWINGS WITH SECTIONAL VIEWS

2.1 Definitions

Definition 1 (*Cutting line*): In the drawing, a cutting line shows where the object was cut to obtain a section view [18].

Definition 2 (*Cutting direction*): Cutting direction means the normal vector of a sectional view in 3D object based coordinates.

Definition 3 (*Incomplete edge*): In sectional views, an edge is defined as an incomplete edge if it intersects with a cutting line.

In half sections and partial sections, the edges in both sides of cutting lines are incomplete edges and represent the projections at different depths. For example, in VIEW1 of Fig.1, the bold lines a, b, d and arc c are incomplete edges. The edge <A,C> is broken into two edges a and b by a cutting line. However, there is no edge in VIEW2 that can be matched with a and b. Such incomplete edges should be repaired for edge matching.

Incomplete edges can be lines and arcs. Arc c in VIEW1 of Fig.1 is an incomplete arc. For an incomplete arc, its center point and radius are used to recover a complete circle. In other cases, incomplete edges need to be recovered by using the projection matching relations. For convenience, the intersection point of an incomplete edge and a cutting line is defined as the *Ending Point*, and the other endpoint is defined as the *Starting Point*. Based on the definition of Ending Point and Starting Point, the rule of matching incomplete edges is described as follows.

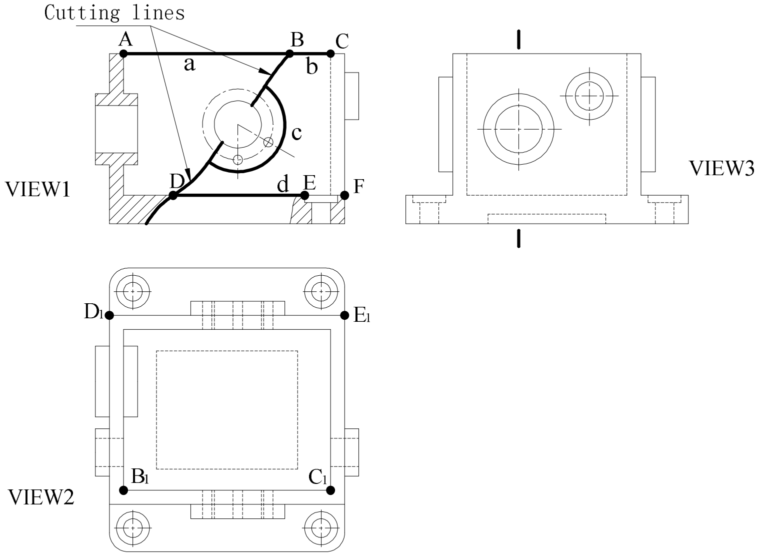


Fig. 1: An example of incomplete edges.

2.2 Characteristics of Sectional Views

The following characteristics of sectional views have been identified by analyzing a large number of engineering drawings

- There are many incomplete edges in half sections and partial sections.
- If projections in one or two views can clearly represent a volume, then the projections in other views can be omitted. Revolutions such as cylinders and cones are the most common cases. Usually, in the case that their axes are paralleled to coordinate planes, two of their projections are same and one of them can be omitted. Moreover, the projections in only one view are enough for through holes and through grooves.
- For holes which are distributed on a circle, it is often that only one or two of them are drawn. The number of these holes can be labeled if need be.
- The projection planes are often parallel or vertical to the axes of mechanical parts to describe the parts more simply. Therefore, most of the lines in 2D views are horizontal or vertical to coordinate axes, and the projections of many edges will be degenerated to 2D points in some views.

3 SECTIONAL VIEWS RECOVERING ALGORITHM

3.1 Overview

In this paper, our purpose is to convert sectional views to three orthographic views based on the projection-missing characteristics of sectional views.

The input is 2D vector engineering drawings in CAD neutral format, such as .DFX and .DWG. After the drawing is imported, all the geometric entities and symbolic entities relative to section expressions (i.e., hatching, cutting lines, and center lines) are filtered. In the preprocessing phase, the 2D views have been separated and converted from view-based coordinate system to object based coordinates according to the bounding box of view and the equal relationship among views, and then the view relations are labeled, as described by Liu et. al[10] and Gong et. al[18].

In the stage of view recovery, there are four main steps, as shown in Fig.2:

(1) Merge and add views;

(2) Repair incomplete edges;

(3) Recover omitted projections of revolutions;

(4) Recover vertical edges and horizontal edges.
The details of each stage are described in the following subsections.

Finally, 3D wireframe models are reconstructed from the three views via the existing Brep-based algorithms[11,19].

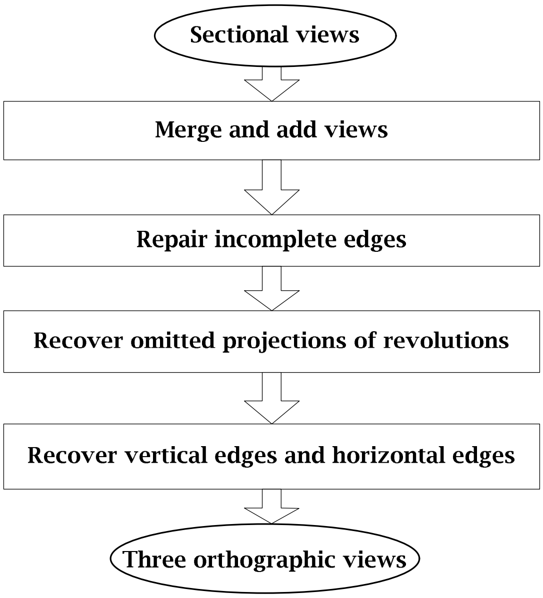


Fig. 2: The flow chart of the view recovery algorithm.

3.2 Merge and add Views

In view of the situation that there might be several views that in the same cutting direction which has been computed according to the location of the views in the preprocessing phase, as described by Liu[10] and Gong[18]. For example, in Fig.3, view A-A, B-B and C-C have the same cutting direction. These views need to be merged into one view. Views' merging is union of their sets of edges and points. Then, the overlap elements in merged view are removed. Meanwhile, an empty view need to be added if the number of views is less than three. The location of the third view is determined according to the equal relationship among the bounding boxes of the existing views. The pseudo-code of merging and adding views is shown as Algorithm 1.

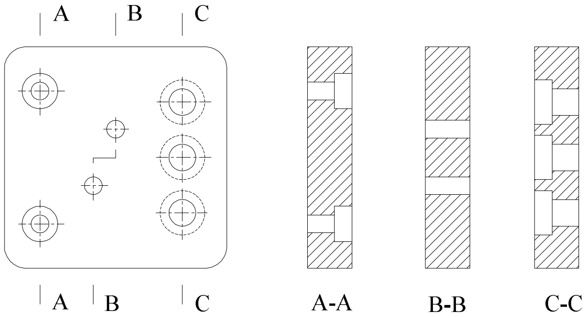


Fig. 3: An example of four views.

Algorithm 1. *Merge and add views*
Input: GS: set of views.
Output: GS': new set of views.

```
Let GS' = Ø;
For (each view G^i in GS) do
   Compute the cutting direction V^i of G^i;
   For (each view G^j in GS (i≠j)) do
      Compute the cutting direction V^j of G^j;
      If (V^i == V^j) then
         G^i = G^i ∪ G^j;
         Delete G^j;
GS' = GS' ∪ {G^i};
If (the number of views in GS' is two) then
   Add a new empty view to GS' according to the location of other views;
```

For example, views A-A, B-B and C-C in Fig.3 have the same cutting direction. According to Algorithm 1, view B-B is merged to view A-A firstly, and then view C-C is merged to view A-A. As shown in Fig.4, view A-A is the result of merging views, which contains original views A-A, B-B and C-C. After merging views, an empty view would be added because the number of views is 2.

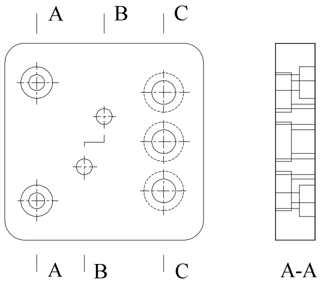


Fig. 4: An example of merging views.

3.3 Repair Incomplete Edges

There are many incomplete edges in sectional views. According to Definition 3, incomplete edges can be identified from sectional views, and then they need to be recovered. The difficulty of this step is finding the relationship between incomplete edges and complete edges.

Rule 1 (*Match incomplete edges*): An edge $a \in G^i$ is said to be a matching edge of an incomplete edge $b \in G^j$, if $P^X(a^P) = P^X(b^S)$, $P^X(a^P) < P^X(b^E) < P^X(a^Q)$ or $P^X(a^Q) < P^X(b^E) < P^X(a^P)$, where X is the common coordinate axis of these two different views G^i and G^j, $P^X(n)$ returns the projected coordinate value of point n along X-axis, b^S and b^E represent the Starting Point and Ending Point of b, and a^P and a^Q are the endpoints of a.

For example, in Fig.1, edge b in VIEW1 is an incomplete edge. Its Starting Point is C, and its Ending Point is B. According to the Rule 1, edge $<B_1,C_1>$ in VIEW 2 can be matched with b. The pseudo-code of repairing incomplete edges is shown as Algorithm 2.

Algorithm 2. *Repair incomplete edges*
Input: CE: set of complete edges;
IE: set of incomplete edges.
Output: CE': new set of complete edges.

```
Let CE' = CE;
For (each ie^i in IE^i of view G^i) do
   For (each ce^j in CE^j of view G^j (i≠j)) do
      Judge whether ie^i and ce^j are matching edges or not using Rule 1;
```

If (ie^i and ce^j are matching edges) then
Extend ie^i to convert a complete edge ce^i according to the matching information;
$CE' = CE' \cup \{ ce^i \}$;

3.4 Recover Omitted Projections of Revolutions

There are a great number of circles and arcs in engineering drawings, such as the projections of cylinders, cones, chamfers, etc. According to different types of projections of revolutions, the process is divided into the following three steps:
Step1: Add the omitted circles which are uniformly distributed along a circumference;
Step2: Recover the omitted projections of chamfers;
Step3: Recover the omitted projections of cylinders and cones.

In half sections and partial sections, circles which are uniformly distributed along a circumference are usually partially omitted because of their regular distribution. Circles in VIEW1 of Fig.1 are such a case. In this step, we proposed an algorithm to deal with the omitted circles by extracting their regularities of distribution. The pseudo-code is shown as Algorithm 3.

Algorithm 3. *Add the omitted circles*
Input: CL: set of circles whose line types are center line in view G;
C: set of circles in view G.
Output: G': recovered view of G
Let $G' = \emptyset$;
For (each cl^i in CL) do
For (each c^j in C) do
If(the center point of c^j is not located on cl^i) then
Continue;
Let r^j = the radius of c^j;
For (each c^k in C, ($j \neq k$)) do
If(the center point of c^k is not located on cl^i) then
Continue;
Let r^k = the radius of c^k;
If ($r^k == r^j$) then
Compute the angle a^{jk} between the centers of c^k, cl^i and c^j;
Let $n = 360/ a^{jk}$, where n is the number of the uniformly distributed circles;
If (the number of circles in C which is located on cl^i is not equal to n) then
Compute the positions which are uniformly distributed on cl^i;
Add the omitted circles to view G' according to the positions and radius;
$G' = G' \cup G$;

The result of recovering circles with uniform distribution in VIEW1 of Fig.1 is shown in Fig.5. Four circles are added to the view.

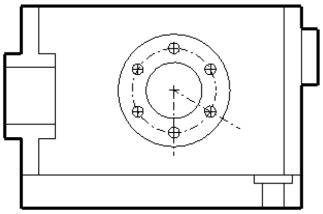


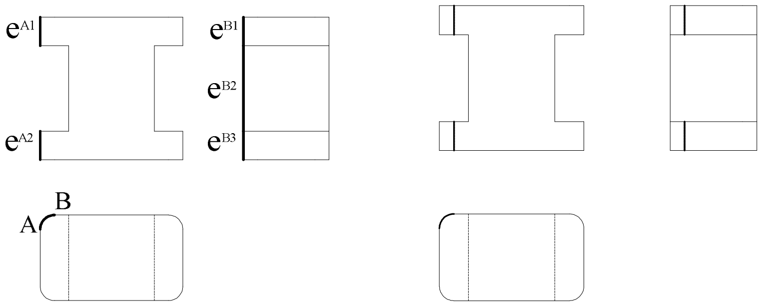
Fig. 5: The result of adding omitted circles.

The projections of chamfers are always incomplete in both sectional views and orthographic views because tangent edges have no projection. However, the projections of chamfers must be recovered when reconstructing 3D models. Here, we use the projection relations to recover desired edges. The pseudo-code of adding projections of chamfers is shown as Algorithm 4.

Algorithm 4. *Recover the projections of chamfers*

```
Input :  A : set of arcs;
         L : set of lines.
Output: GS': set of recovered views
For (each a^i in A of view G^i) do
   If (not both endpoints of a^i are tangent with their adjacent lines) then
      Continue;
   Let sum^j = 0;
   For (each l^j in G^j (i≠j)) do
      Compute the common coordinate axis X between G^i and G^j;
      If (l^j is vertical to X-axis) then
         If (P^X(a^ip) == P^X(l^j), where a^ip is an endpoint of a^i) then
            Let sum^j = sum^j + the length of l^j;
            L^j = L^j∪{l^j};
   Let sum^k = 0;
   For (each l^k in G^k (k≠i , k≠j)) do
      Compute the common coordinate axis Y between G^k and G^i;
      If (l^k is vertical to Y-axis) then
         If (P^X(a^ip) == P^X(l^k)) then
            Let sum^k = sum^k + the length of l^k;
            L^k = L^k∪{l^k};
   If (sum^j < sum^k) then
      Add rectangles in views G^j and G^k which have the same projections with a^i and l^j in L^j;
   Else
      Add rectangles in views G^j and G^k which have the same projections with a^i and l^k in L^k;
GS'={G^i, G^j, G^k}
```

For example, in Fig.6(a), an endpoint A of arc <A, B> has the same projection with edges e^{A1} and e^{A2} of the upper left view, and the other endpoint B has the same projection with edges e^{B1}, e^{B2} and e^{B3} of the right view. Because the sum of lengths of e^{A1} and e^{A2} is shorter than that of e^{B1}, e^{B2} and e^{B3}, we add rectangles in these two views according to the projections of arc <A, B> and edges e^{A1} and e^{A2}. The added lines are shown in Fig. 6(b).



(a) (b)

Fig. 6: An example of recovering projections of a chamfer. (a) original views, (b) recovered views.

A typical characteristic of projections of cylinders and cones is that two of their projections are the same. Thus, it is not difficult to recover their omitted projections. The recovering algorithm is as follows.

Algorithm 5. Recover the omitted projection of cylinders and cones

Input: GS: set of views;
C: set of circles and arcs;
L: set of lines.
Output: GS': set of recovered views
For (each c^i in C of view G^i in GS) do
  Let $L^j = \emptyset$;
  For (each l^j in L of view $G^j (i \neq j)$ in GS) do
    If (l^j is vertical line or horizontal line) then
      If (l^j has the same projection with c^i) then
        $L^j = L^j \cup \{ l^j \}$;
  If (there are four lines in L^j which can form a rectangle R^j) then
    If (there is no rectangle R^k in view $G^k (k \neq i, k \neq j)$ which can match c^i and R^j)then
      Add a new rectangle which has the same projection with c^i and R^j to view G^k, $(k \neq i, k \neq j)$;
For (each c^i in C of view G^i in GS) do
  For (each c^k in C of view G^i in GS) do
    If (c^i and c^k are concentric circles) then
      Let $L^j = \emptyset$;
      For (each l^j in L of view $G^j (i \neq j)$ in GS) do
        If (l^j has the same projection with c^i) then
          $L^j = L^j \cup \{ l^j \}$;
      If (there are four lines in L^j which can form an isosceles trapezoid T^j) then
        Add a new trapezoid which has the same projection with c^i and T^j in G^k, $(k \neq i, k \neq j)$;
GS' = GS;

Fig.7 shows the result of recovering the drawing in Fig.1 after using Algorithm 3, Algorithm 4 and Algorithm 5.

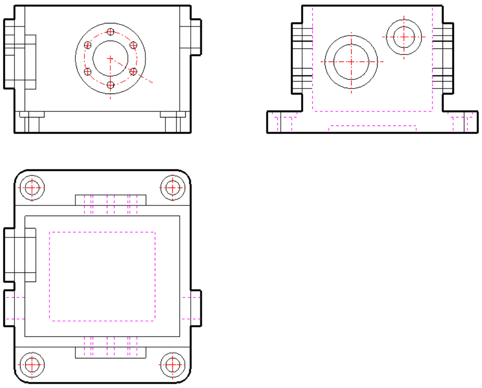


Fig. 7: The result of recovering omitted projections of revolutions.

3.5 Recover Vertical Edges and Horizontal Edges

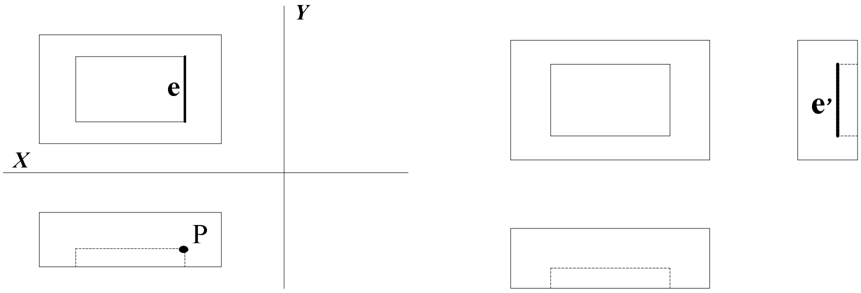
In sectional views, there are lots of horizontal edges and vertical edges. In most cases, this situation will lead to projection degeneration. The main task in this step is to recover edges in such a case by matching the projection relations. The pseudo-code of recovering vertical and horizontal edges is shown as Algorithm 6.

Algorithm 6. ***Recover vertical edges and horizontal edges***
Input: GS: set of views
E: set of 2D edges
P: set of 2D points

Output: GS': set of recovered views

For (each e^i in E^i of view G^i in GS) do
  If (e^i is neither a vertical line nor a horizontal line) then
    Continue;
  Find a view G^j ($i \neq j$)) which has the common coordinate axis X with G^i and e^i is vertical to X-axis;
  For (each p^j in P^j of view G^j) do
    If ($P^X(e^i) \neq P^X(p^j)$) then
      Continue;
    If (there is no edge which can be matched with e^i and p^j in view G^k ($k \neq i$, $k \neq j$)) then
      Add an edge in G^k according to e^i and p^j;
GS' = GS;

For example, in Fig.8(a), edge e is vertical to X-axis. Meanwhile, there is a point P in the bottom view. According to algorithm 6, e has the same projection with P along X-axis. Because there is no edge in the side view which is matched with e and P, an edge would be added in the side view according to their projections. The added edge e' is shown in the side view of Fig. 8(b).

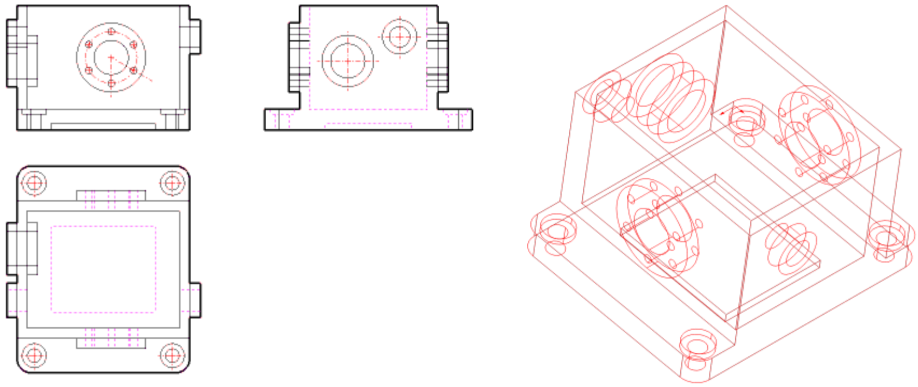


(a) (b)

Fig. 8: Recovering vertical edges and horizontal edges. (a) original views, (b) recovered views.

4 IMPLEMENTATION AND DISCUSSION

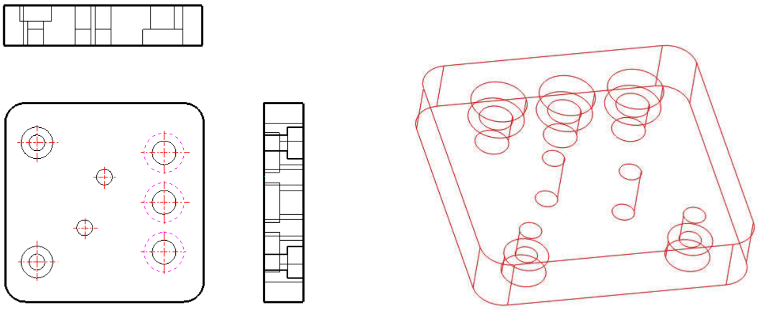
Our algorithm is implemented and integrated into the platform of a geometric modeling system. The developed program accepts drawing files in CAD neutral format (.dwg or .dxf). Several examples are provided to demonstrate the practicability of our algorithm.



(a) (b)

Fig. 9: Box. (a) recovered three orthographic views, (b) 3D wireframe model.

Fig.9(a) shows the recovered three orthographic views from the views in Fig.1. The repairing operations in this case are listed as follows: recovering incomplete edges, recovering the volumes that are uniformly distributed in the object, recovering the profiles of revolutions and chamfers, and recovering vertical edges and horizontal edges. The final 3D wireframe model is shown in Fig.9(b).

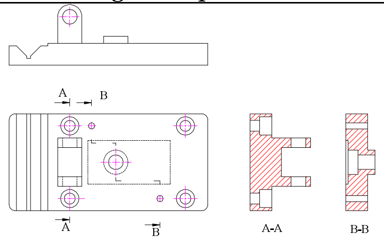
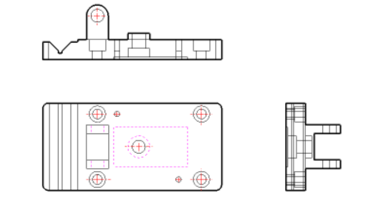
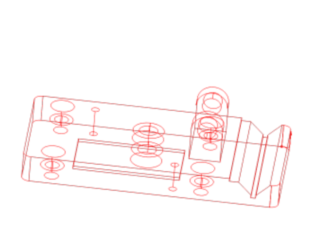
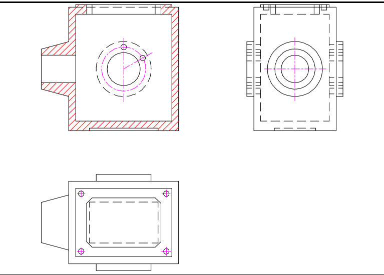
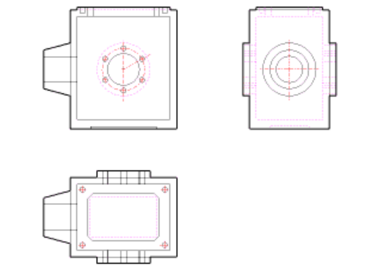
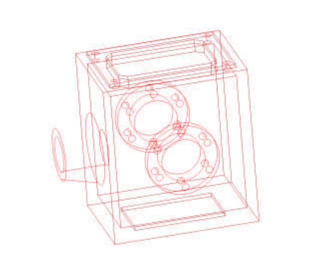
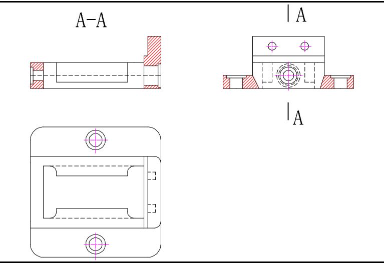
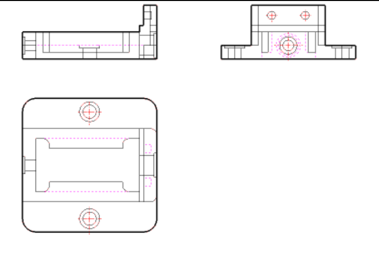
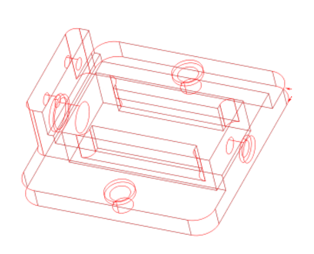


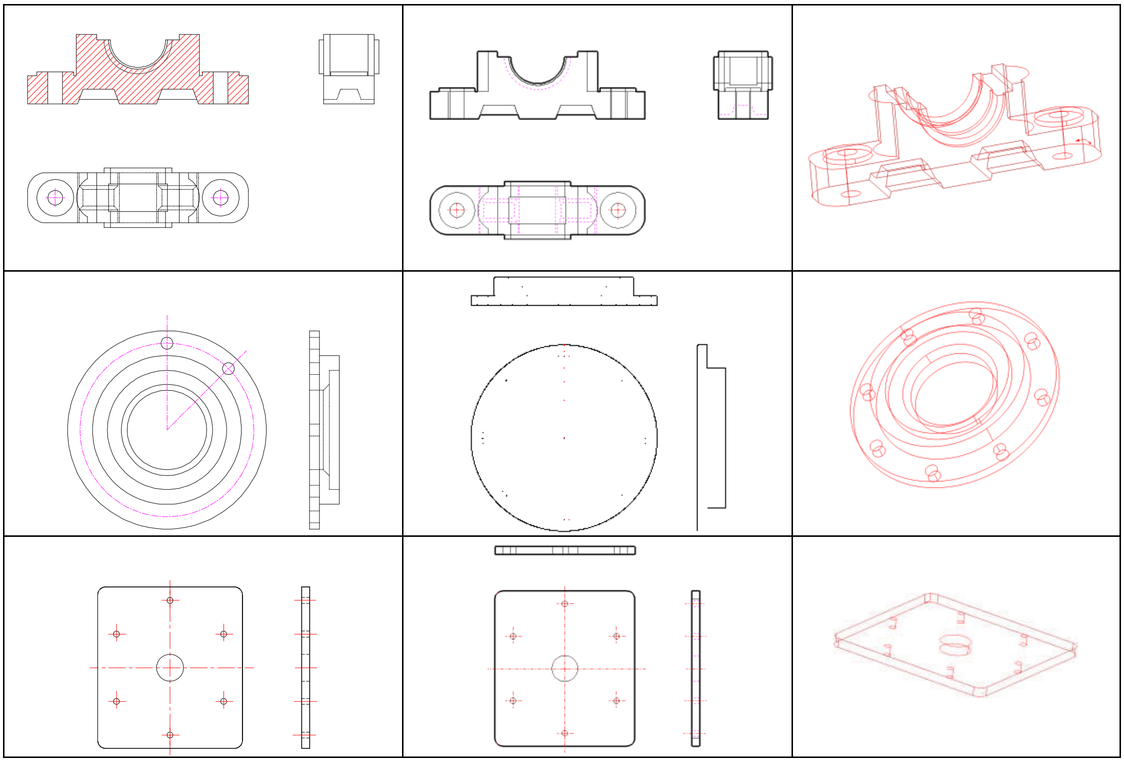
(a) (b)

Fig. 10: Flat plate. (a) recovered three orthographic views, (b) 3D wireframe model.

Fig. 10(a) shows the recovered three orthographic views from the sectional views in Fig.2. First, the views with the same projection direction are merged into one view. Then, a new empty view is added since there are only two views. Finally, the missing edges are added and the complete three views are recovered. The final 3D wireframe model is shown in Fig. 10(b).

Many more examples are listed in Tab. 1 to demonstrate that various cases can be handled by our algorithm.

Original input views	Recovered three views	3D wireframe model
		
		
		



Tab. 1: Several examples.

5 CONCLUSION

Sectional views are widely used in engineering practices. However, it is difficult to reconstruct 3D objects from sectional views since there are many omitted edges. In this paper, we propose a new algorithm to transform sectional views to three orthographic views for the further process of 3D reconstruction. By analyzing the rules of projection-missing in sectional views, the omitted edges can be recovered. The recovered complete three orthographic views can be used to reconstruct 3D solid models using the existing Brep-based method. Full sections, offset sections and partial sections are handled in this paper. However, this algorithm is limited to the views with three view projection directions. In the future, the cases which have more than three view projection directions will be considered. Also, we will extend the work to other sections, such as revolve sections, cross sections as well as oblique sections. Moreover, we will study on errors correction of 2D views so that the algorithm can handle imprecise drawings.

ACKNOWLEGMANTS

This work was supported by the 973 Program of China (Grant No. 2010CB328001) and the National Nature Science Foundation of China (Grant No.60903106 and No.61035002).

REFERENCES

[1] Idesawa, M.: A system to generate a solid figure from three view. Bulletin of the Japan Society of Mechanical Engineering, 16, 1973, 216-225.

[2] Nagendra, I.V.; Gujar, U.G.: 3-D objects from 2-D orthographic views a survey. Computers and Graphics, 12(1), 1988, 111-114. doi:10.1016/0097-8493(88)90015-5

[3] Wang, W.; Grinstein, G.G.: A survey of 3D solid reconstruction from 2D projection line drawings. Computer Graphics Forum, 12(2), 1993, 137-158. doi:10.1111/1467-8659.1220137
[4] Chen, Z.; Perng, D.B.: Automatic reconstruction of 3D solid objects from 2D orthographic views. Pattern Recognition,21(5), 1988, 439-449. doi:10.1016/0031-3203(88)90003-9
[5] Shum, S.S.P.; Lau, W.S.; Yuen, M.M.F.; Yu, K.M.: Solid reconstruction from orthographic views using 2-stage extrusion. Computer-Aided Design, 33(1), 2001, 91-102. doi:10.1016/S0010-4485(00)00079-8
[6] Soni, S.; Gurumoorthy, B.: Handling solids of revolution in volume-based construction of solid models from orthographic views. Journal of Computing and Information Science in Engineering, 3, 2003, 250-259. doi:10.1115/1.1604831
[7] Lee, H.; Han, S.: Reconstruction of 3D interacting solids of revolution from 2D orthographic views. Computer-Aided Design,37(13), 2005, 1388-1398. doi:10.1016/j.cad.2005.01.007
[8] Wesley, M.A.; Markowsky, G.: Fleshing out projections. IBM Journal of Research and Development, 25(6), 1981, 934-953. doi:10.1147/rd.256.0934
[9] Yan, Q.W.; Chen, C.L.P.; Tang, Z.S.: Efficient algorithm for the reconstruction of 3D objects from orthographic projections. Computer-Aided Design, 26(9), 1994, 699-717. doi:10.1016/0010-4485(94)90020-5
[10] Liu, S.X.; Hu, S.M.; Chen, Y.J.; Sun, J.G.: Reconstruction of curved solids from engineering drawings. Computer-Aided Design, 33(14), 2001, 1059-1072. doi:10.1016/S0010-4485(00)00143-3
[11] Gong, J.H.; Zhang, H.; Zhang, G.F.; Sun, J.G.: Solid reconstruction using recognition of quadric surfaces from orthographic views. Computer-Aided Design, 38(8), 2006, 821-835. doi:10.1016/j.cad.2006.04.009
[12] Aldefeld, B.; Richter, H.: Semiautomatic three-dimensional interpretation of line drawing. Computers and Graphics, 8(4), 1984, 371-380. doi:10.1016/0097-8493(84)90035-9
[13] Ho, B.: Inputting constructive solid geometry representations directly from 2D orthographic engineering drawings. Computer-Aided Design, 18(3), 1986, 147-155. doi:10.1016/0010-4485(86)90325-8
[14] Geng, W.D.; Wang, J.B.; Zhang, Y.Y.: Embedding visual cognition in 3D reconstruction from mutil-view engineering drawings. Computer-Aided Design, 34, 2002, 321-336. doi:10.1016/S0010-4485(01)00092-6
[15] Jitendra, D.; Gurumoorthy, B.: Handling sectional views in volume-based approach to automatically construct 3D solid from 2D views. Computer-Aided Design, 37, 2005, 485-495. doi:10.1016/j.cad.2004.10.007
[16] Wen, Y.M.; Zhang, H.: Identifying features in reconstructing 3D solids from sectional views. Proceedings 2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, 2009, 377-382.
[17] Wen, Y.M.; Zhang, H.; Yu, Z.M.; Sun, J.G.; Paul, J.C.: Reconstructing 3D objects from 2D sectional views of engineering drawings using volume-based method. In: proceeding of Shape Modeling International 2010, 13-24. doi:10.1109/SMI.2010.16
[18] Gong, J.H.; Zhang, H.; Jiang, B.; Sun, J.G.: Identification of Sections from Engineering Drawings Based on Evidence Theory. Computer-Aided Design, 42(2), 2010, 139-150. doi:10.1016/j.cad.2009.04.015
[19] Gong, J.H., Zhang, H, Zhang, Y.W., Sun, J.G.: Converting hybrid wire-frames to B-rep models. Proceedings of ACM Symposium on Solid and Physical Modeling (SPM2007), Beijing, China, 2007. 283-289.