



# Identification of Sections from Engineering Drawings Based on Evidence Theory

Jie-Hui Gong, Hui Zhang, Bin Jiang, Jia-Guang Sun

## ► To cite this version:

Jie-Hui Gong, Hui Zhang, Bin Jiang, Jia-Guang Sun. Identification of Sections from Engineering Drawings Based on Evidence Theory. ACM Solid and Physical Modeling Symposium (SPM 2008), Jun 2008, New York, United States. inria-00515523

**HAL Id: inria-00515523**

**<https://inria.hal.science/inria-00515523>**

Submitted on 7 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Identification of Sections from Engineering Drawings Based on Evidence Theory \*

Jie-Hui Gong<sup>†</sup>  
School of Software  
Tsinghua University  
Beijing 100084, P.R.China

Bin Jiang  
School of Software  
Tsinghua University  
Beijing 100084, P.R.China

Hui Zhang  
School of Software  
Tsinghua University  
Beijing 100084, P.R.China

Jia-Guang Sun  
Tsinghua University  
Beijing 100084, P.R.China

## ABSTRACT

View identification is the basal process for solid reconstruction from engineering drawings. A new method is presented to label various views from a section-involved drawing and identify geometric planes through the object at which the sections are to be located. In the approach, a graph representation is developed for describing multiple relationships among various views in the 2D drawing space, and a reasoning technique based on evidence theory is implemented to validate view relations that are used to fold views and sections in the 3D object space. This is the first automated approach which can handle multiple sections in diverse arrangements, especially accommodating the aligned section for the first time. Experimental results are given to show that the proposed solution makes a breakthrough in the field and builds a promising basis for further expansibility, although it is not a complete one.

## Categories and Subject Descriptors

J.6 [Computer-Aided Engineering]: Computer-Aided Design; G.3 [Probability and Statistics]: Probabilistic algorithms

## Keywords

Engineering drawing, Orthographic projection, Sectional view, Evidence theory, Solid reconstruction

## 1. INTRODUCTION

\*This work was supported by the 973 Program of China (Grant No. 2004CB719404) and the National Nature Science Foundation of China (Grant No. 60635020).

<sup>†</sup>gongjh03@gmail.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SPM '08 Stony Brook, New York, USA

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

Although solid models play a central role in modern mechanical design, 2D CAD systems are still commonly used for designing regular-shaped products [15]. Many product designs, including legacy data, are definitively expressed in the form of 2D engineering drawings. As yet, human interpretation is essential in order to retrieve the complete information contained in the drawings for downstream computer-aided manufacturing technologies. Under this situation, an automatic engine for solid reconstruction from multi-view drawings would be very helpful to shorten design cycle, improve production efficiency, and reduce manufacturing costs. The solution would have a wide range of practical applications, such as one-off converting 2D drawings into 3D CAD/CAM environment [5] and describing shapes in solid modeling systems as an alternate means of interaction [8].

In engineering practice, sections have been a staple mode to describe product designs, and most actual drawings contain sectional views [11]. Sections, or sectional views, are essential for presenting hidden details of objects without cluttering the drawings, and they are easy to draw and read under various rules and conventions.

A body of work related to solid reconstruction exists in the literature (refer to [14, 7] for a review of current literature). The current art only stresses geometric information embedded in regular views [16], while the problem of constructing solid models from sectional views has rarely been addressed in detail due to the intricacy and flexibility of conventional expressions of sections. Wesley et al. [17] first took sectional views into consideration in their B-rep oriented method. A full section with its cutting plane and viewing direction was used to define a half-space to determine candidate vertices of the object. However, the authors only discussed full sections and did not provide any results of implementation for sectional views. Aldefeld et al. [1] and Bin [2] also considered sectional views in their semi-automatic CSG oriented methods, respectively. But these methods rely on the guidance of user interpretation to generate primitives, which involve heavy manual operations. Geng et al. [6] presented a hybrid of B-rep and CSG oriented schemes for automatic solid reconstruction. Full sections were handled as an additional virtual partition of the 3D object space to help delete ambiguities that arise from the selection of base-faces and trajectories based on regular views. Nevertheless, their method can not directly handle a sectional view without the regu-

lar view in the same view direction, which excludes most of actual drawings. More recently, Dimri et al. [4] described a CSG oriented method to explicitly handle sectional views. They presented an algorithm to identify additional ontoformations from sectional views considering the changes in terms of edges and loops in sectional views as compared to regular views. Their method can handle the full section, half section, and offset section. But it is incompetent for multiple sections in various arrangements and just applied to the drawings with one sectional view in the standard configuration, which rigidly restricted its applications.

In any complete system of solid reconstruction, the initial process inevitably starts with the 2D drawing space. It is necessary to first determine which views and what sections are present before the 3D interpretation process can begin. Currently, view labelling techniques [10, 19] only apply to regular views. All the existing approaches that consider sectional views typically start from the assumption that the drawing layout and the view types must be known in advance. Manual interventions has to be introduced to identify, place, and align multiple views, which are normally troublesome, time-consuming, and error-prone due to the diversiform arrangement of various sections. This limitation compromises the efficiency and practicability of object recovery systems.

The purpose of this research is to develop a view identification method to accommodate the full section, offset section, aligned section, as well as the regular view in diverse configurations. To the best of our knowledge, there is no such algorithm in the literature so far. We overcome the limitation of the existing methods by employing a method based on evidence theory. The proposed method uses a graph formalism to collect geometric and symbolic relations of various views in the 2D drawing space as the evidence for view identification, and then uses evidence-based reasoning to determine view relations that are valid for transforming views and sections into the 3D object space. Using the method, we manage to extend the applicable range and improve the flexibility of view identification.

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 the graph representation defined. The details of the proposed method are described in Section 3, followed by experimental results of implementation in Section 4. Section 5 is the conclusion and future work.

## 2. CHARACTERISTICS OF ENGINEERING DRAWINGS WITH SECTIONAL VIEWS

### 2.1 Standard layout of orthogonal projections

In technical drafting practices, an engineering drawing normally consists of two or more orthographic projections of the represented object. Being imaginarily placed in a transparent box, as shown in Figure 1(a), the object can be projected into six orthogonal view-planes, i.e., the *front*, *top*, *left*, *bottom*, *right*, and *rear* projections. These standard projections are laid in the 2D *drawing-based coordinate system* on the  $XY$ -plane as shown in Figure 1(b). Each projected view is a separate line-drawing with the outline enclosing all the projected curves with respect to the corresponding *projection direction*. In any particular view, solid curves

represent the portions of the object that are directly visible from the projection direction, whilst dashed ones show the invisible portions.

The standard layout of orthographic views provides enough information to determine how the views are related to one another in the 3D *object-based coordinate system*. Each view in the drawing is aligned and adjacent to at least one other view, and the view planes of any two adjacent views must be perpendicular to each other and share their common Cartesian axis. The main view often corresponds to *front* view in a typical drawing layout and provides the basis for defining the 3D object space. Therefore, the projection plane of each view can be determined by its position in the drawing, as the *view-based coordinates* shown in Figure 1(b). For instance, front view corresponds to the  $zx$ -plane, top view to the  $xy$ -plane, and left view to the  $yz$ -plane. All the views can be accordingly transformed to their respective planes in the 3D object-based coordinate system.

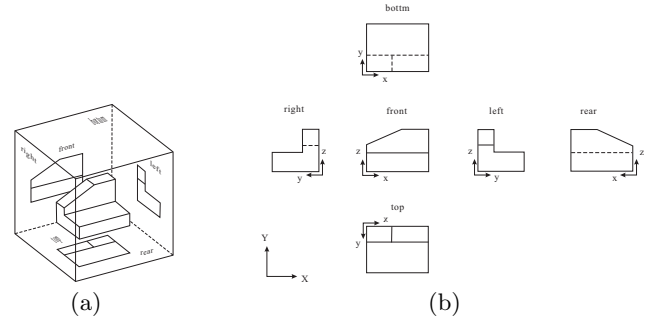


Figure 1: (a) Orthographic projections. (b) Standard layout of orthographic views.

### 2.2 Sectional views

Sections are commonly employed in engineering practice. Under the convention of orthographic projection, the hidden part from the view direction is represented as dashed lines. For an object whose interior details are intricate, the clutter of “hidden” lines in a regular view would be very difficult to understand. A sectional view can be applied to overcome this difficulty, by pretending to cut the object with one or more planes and remove the part nearer to the observer.

For presenting sections, the drawing contains symbolic entities interweaving the geometry. These section-related symbols include hatching, phantom lines, texts, and center lines. They provide the information regarding the position of sections in the 3D object space. For example in Figure 6, the presence of sectional views is indicated by cut surfaces filled with hatching in the drawing. Cutting planes are indicated in a view adjacent to the sectional view by phantom lines along with arrows that show the direction of sight. The arrows would be omitted if the collocation of the adjacent views satisfies the projection relation. Furthermore, the ends of cutting planes are labelled with texts, and the sectional view is named accordantly to match.

The following types of sections are appropriate to present an object with complex and asymmetric interior along with simple exterior [3].

- *Full section*

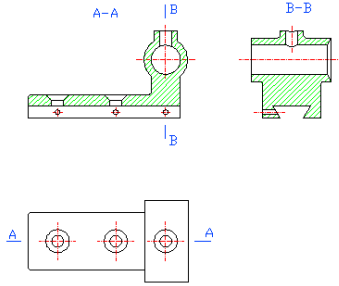
One cutting plane cuts entirely through an object, and hidden lines become solid that serve as boundary of section lines in the view. Figure 2 is an example with two full sections named “A–A” and “B–B”, respectively.

- *Offset section*

Several cutting planes offset to pass through staggered interior features of an object, and the direction changes of these planes are not represented in the view. Figure 3 shows an example with three offset sections.

- *Aligned section*

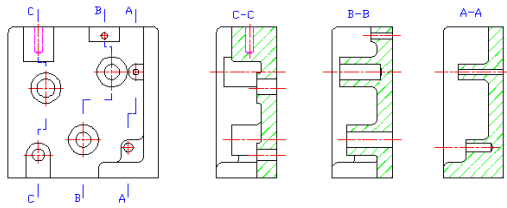
Two cutting planes intersect in the axis of circular features of an object, and the view is drawn as if the unaligned portion is revolved into the projection plane. An example with an aligned section is given in Figure 4.



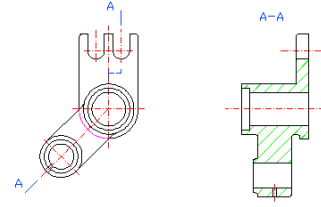
**Figure 2: Full section.**

For presenting different structural characteristics of the drawn object, as shown in Figures 2–4, several sections of various types may be neatly combined and variously arranged in the drawing. The sections may cause the changes of view relationships in terms of adjacency and alignment as compared with regular views.

On the one hand, the view adjacency of the standard layout is extended due to inducting sectional views in the drawing. A sectional view usually replaces a regular view in order to satisfy the standard layout of orthographic projections. If lack of space restricts the normal placement of a sectional view, the view may be placed in an alternate location. In this case, the sectional view should remain in the same orientation as if it were a direct projection from the cutting plane. Referring to Figure 3, the sectional views in the same projection direction are collocated to satisfy their projection relations with respect to the main view. Although the collocation of these views cannot fulfil the standard layout, the adjacent relations are indicated by the matching texts of the



**Figure 3: Offset section.**



**Figure 4: Aligned section.**

cutting lines in the main view and the names of the sectional views.

On the other hand, the view alignment of the standard layout may be compromised by replacing a regular view with a sectional view. Since a portion of the object is imaginarily removed for producing a full section or an offset section, the apparent size of the sectional view may be smaller than that of the regular view in the same projection direction. For an aligned section, the angled cutting plane is pretended to be revolved to the projection plane, thus the apparent sizes of the adjacent views may become unequal. For instance, in Figure 4, the adjacent views are not aligned since the apparent size of the aligned section is larger than that of the main view.

### 2.3 Graph representation of view relations

As described above, section-involved engineering drawings contain the geometric and symbolic information to fold views and sections in the 3D object space. We propose a representation, referred to as the *view relational graph* (VRG), to formalize all the information for describing view relations of a multi-view drawing with respect to the 2D drawing space. This representation can be regarded as an attributed directed multi-graph structure. Each node of the graph corresponds to a unique view that refers to an isolated projection image of the object relative to certain projection direction. Each directed arc of the graph, from one node to another, corresponds to a certain relation between the two views. We consider a set of view relations as follows. For the three kinds of sectional views that we are dealing with, these relations contain enough information for reasoning projection planes and identifying cutting planes.

- *Parallel-size correspondence* ( $E_1$ ): indicates that one view presents the same apparent size as the other view in the same coordinate-axis direction, without any other view between them.
- *Perpendicular-size correspondence* ( $E_2$ ): indicates that one view presents the same apparent size as the other view in perpendicular coordinate-axis directions, without any other view between them.
- *Name correspondence* ( $E_3$ ): indicates that one of the labelled cutting lines in one view corresponds to the name of the other view.
- *Center correspondence* ( $E_4$ ): indicates that the center of a circle in one view matches certain center line in the other view in the same axis direction.

Therefore, for a multi-view drawing, the VRG is defined as a tuple  $G = (N, A, T_n(N), T_a(A))$ , where  $N$  is the set of nodes, and  $A$  is the set of directed arcs. Each node or arc

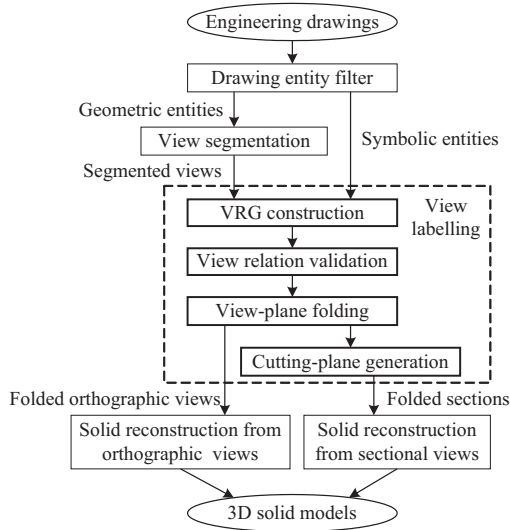
is assigned a vector of attributes. As summarized in Tab. 1,  $T_n(N) = \{t_n | t_n = (V(n), P(n), S(n), L(n)), n \in N\}$  and  $T_a(A) = \{t_a | t_a = (R(a), N_s(a), N_e(a), D(a)), a \in A\}$ , where  $L(n) = \{l_i^n | l_i^n = (L_n(n, i), L_a(n, i), L_v(n, i)), i = 1, \dots, k\}$  is the set of  $k$  cutting lines, and  $L_v(n, i) = \{(l_{js}, l_{je}) | j = 1, \dots, p\}$  is the set of  $p$  valid portions of the  $i$ th cutting line.

Lysak et al. [10] used a tree structure to describe the view adjacency of the standard layout, with each view represented as a node and each adjacency relationship as an arc. By allowing more than one view in the same projection direction, we extend the tree structure to present the view adjacency involving sections. Each branch between two nodes may describe adjacent and/or sectional relationships between the corresponding views.

### 3. VIEW LABELLING ALGORITHM

#### 3.1 Overview

The process of view labelling is illustrated in Figure 5, which is outlined within the pipeline of solid reconstruction from engineering drawings. The starting point is engineering drawings available as formatted data structures in CAD neutral format. After the drawing is imported, all the curves with solid/dashed line-types are first filtered as geometric data that is the projective images of the object. Symbolic entities relative to section expressions (i.e., hatching, phantom lines, texts, and center lines) also filter through for further processing.



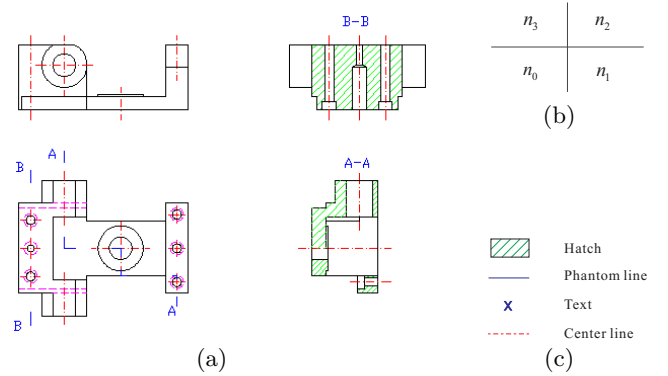
**Figure 5: View labelling in the pipeline of solid reconstruction from engineering drawings.**

In the stage of view segmentation, the geometric data are first checked and converted to a planar graph, as described by Liu et al. [9]. All intersection points between curves are calculated, and curves are partitioned at these intersection points. Redundant curves are removed, and co-linear curves are also combined. With that, curves are formed into a graph with distinct edges connecting only at their end vertices. And then the edges and vertices are grouped into views with respect to the drawing-based coordinate system, as proposed by Lysak et al. [10]. Connected sets of edges

are identified by travelling the graph. All the connected sets are checked pairwise for possible containment. Several segmented views are finally obtained by collecting the connected sets with contained relations into the same view.

The segmented views and the symbolic entities are subsequently input to the main algorithm of view labelling. The proposed method first generates the VRG by detecting the collocation and notation in the drawing, and then determines valid relations of the views by reasoning out the view tree based on evidence theory. After that, the view-planes are identified by travelling the branches in the tree, and all the views are accordingly transformed from the 2D drawing space to the 3D object space. The processing terminates hereto for the drawings with regular views. Based on the folded orthographic views, the solid model can be reconstructed by the method presented in authors' previous work [7]. Otherwise, for the drawings with sectional views, the cutting planes corresponding to sections are established according to the sectional relations among the folded views. Folded sections can be accordingly imported to the reconstruction process, and the object can be retrieved by the approaches that considers sectional views, such as [4].

The following Sections describe each procedure of view labelling in detail, illustrating with figures on a part with several features of revolution which are revealed by sectional views. Figure 6 is the segmented views of the part. Figure 7 illustrates the construction of VRG. Section 3.3.3 describes the validation of view relations, with the constructed view tree shown in Figure 8. In Figure 9, the projection planes of the views are sequentially identified, and the views are accordingly folded in the 3D object space as shown in Figure 10. Finally, Figure 11 shows all the cutting planes corresponding to the sections.



**Figure 6: Input drawing and segmented views. (a) Sectional views. (b) View index. (c) Legend.**

#### 3.2 VRG construction

As described above, drawings convey a lot of semantic information in a symbolic way, and a trained human expert can extract a lot of clues on expression modes from a single view and combine the clues to reveal the relations of multi views. In this step, we derive all the available information for representing the relations of the segmented views and construct the VRG of the drawing. This is undertaken in two stages: (1) Characterize nodes of the VRG; (2) Construct directed arcs of the VRG.

**Table 1: Various attributes in the VRG**

Elements	Attributes	Notations	Values
Node	View type	$V(n)$	$V(n) = \text{REGULAR}$ : regular view $V(n) = \text{SECTIONAL}$ : sectional view
	Projection plane	$P(n)$	$P(n) = \text{FRONT}$ : front projection $P(n) = \text{TOP}$ : top projection $P(n) = \text{LEFT}$ : left projection $P(n) = \text{BOTTOM}$ : bottom projection $P(n) = \text{RIGHT}$ : right projection $P(n) = \text{REAR}$ : rear projection
	Section name	$S(n)$	$S(n) = \text{null}$ : the view is regular. $S(n) = \text{"X"}$ : the view is labelled with "X-X".
	Cutting line $i$ ( $i = 1, \dots, k$ )	$L_n(n, i)$	$L_n(n, i) = \text{"X"}$ : the line is labelled with "X".
		$L_a(n, i)$	$L_a(n, i) = 0$ : the arrow is omitted. $L_a(n, i) = 1$ : the line is adjacent to an arrow.
		$L_v(n, i)$	$L_v(n, i) = \{(l_s, l_e)\}$ : the series of valid portions
Arc	Relation type	$R(a)$	$R(a) = E_1, E_2, E_3, \text{ or } E_4$
	Start node	$N_s(a)$	$N_s(a) = n_i$ : one view
	End node	$N_e(a)$	$N_e(a) = n_j$ : the other view
	Relative direction	$D(a)$	$D(a) = X$ : the horizontal axis $D(a) = Y$ : the vertical axis

In the first stage, nodes of the VRG are constructed one-to-one corresponding to the segmented views and assigned attribute vectors by parsing the symbolic information in the drawing. Section-related symbols (i.e., hatching, phantom lines, texts, and center lines) are the important clues for recognizing section expressions. By separating these symbols into respective views, the nodes are characterized as follows.

- (1) Hatching distribution. For each hatching  $h_i$ ,
  - (1.1) Find the node  $n_k$  that encircles  $h_i$ .
  - (1.2) Set the view type of  $n_k$  as *SECTIONAL*, and add  $h_i$  into the hatching list of  $n_k$ .
  - (1.3) Find the text in the form of "X-X" nearest to  $n_k$ , and then set the section name of  $n_k$  as "X".
- (2) Center line distribution. For each center line  $c_i$ ,
  - (2.1) Find the node  $n_k$  which the middle point of  $c_i$  is falling into.
  - (2.2) Add  $c_i$  into the center line list of  $n_k$ .
- (3) Phantom line distribution. For each phantom line  $l_i$ ,
  - (3.1) Find the node  $n_k$  which  $l_i$  is falling into or nearest to.
  - (3.2) Add  $l_i$  into the phantom line list of  $n_k$ .
- (4) Cutting line tracing. For each node  $n_k$  in  $N$ ,
  - (4.1) Find a phantom line  $l_s$  that is located outside  $n_k$  and parallel to certain axis of the drawing-based coordinate system.
  - (4.2) Construct a cutting line  $l_i^{n_k}$  of  $n_k$  beginning with  $l_s$ . Find the text in the form of "X" nearest to  $l_s$ , and then set the label of  $l_i^{n_k}$  as "X".
  - (4.3) Find the phantom line  $l_e$  on the prolongation of  $l_s$  and nearest to  $l_s$ .  $(l_s, l_e)$  is added into the series of valid portions of  $l_i^{n_k}$ .

(4.4) If  $l_e$  is located inside  $n_k$ , retrieve the phantom line  $l_t$  that is adjacent to  $l_e$ .

(4.4.1) If  $l_t$  is perpendicular to  $l_e$ , find the phantom line  $l_f$  on the prolongation of  $l_t$  and nearest to  $l_t$ .  $(l_t, l_f)$  is regarded as the bend of  $l_i^{n_k}$  and strode over. Then, retrieve the phantom line  $l_s$  that is adjacent to  $l_f$ , followed by the phantom line  $l_e$  on the prolongation of  $l_s$  and nearest to  $l_s$ .

(4.4.2) Otherwise, take  $l_t$  as  $l_s$ , and find the phantom line  $l_e$  on the prolongation of  $l_s$  and nearest to  $l_s$ .

(4.4.3)  $(l_s, l_e)$  is added into the series of valid portions of  $l_i^{n_k}$ .

(4.5) Repeat (4.4) until  $l_e$  is located outside  $n_k$ .

(4.6) Repeat (4.1)–(4.5) until all the phantom lines in the list are handled.

In the second stage, all the nodes are checked pairwise for revealing their relations according to their attribute vectors, thus the corresponding directed arcs are constructed and added into the VRG. For each pair of nodes,  $n_i$  and  $n_j$  ( $i \neq j$ ), the view relations are searched as follows, where  $projectA(l)$  is the projected inter-zone on  $A$ -axis of a line  $l$ ,  $projectA(p)$  for the projection of a point  $p$ , and  $projectA(B)$  for the projected inter-zone of a bounding box  $B$ .

- (1) Searching for parallel-size correspondence and perpendicular-size correspondence.
  - (1.1) Retrieve the bounding boxes  $B_i$  and  $B_j$  of  $n_i$  and  $n_j$ , respectively. Determine whether the line segment that connects the center points of  $B_i$  and  $B_j$  traverses the bounding boxes of any other nodes in  $N$ .
  - (1.2) If not,
    - (1.2.1) If  $projectX(B_i) = projectX(B_j)$ , construct an arc  $a$  of  $E_1$  with the start node  $n_i$  and the

end node  $n_j$ , and set the relative direction of  $a$  as  $X$ ;

(1.2.2) If  $projectY(B_i) = projectY(B_j)$ , construct an arc  $a$  of  $E_1$  with the start node  $n_i$  and the end node  $n_j$ , and set the relative direction of  $a$  as  $Y$ ;

(1.2.3) If  $projectX(B_i) = projectY(B_j)$ , construct an arc  $a$  of  $E_2$  with the start node  $n_i$  and the end node  $n_j$ , and set the relative direction of  $a$  as  $X$ ;

(1.2.4) If  $projectY(B_i) = projectX(B_j)$ , construct an arc  $a$  of  $E_2$  with the start node  $n_i$  and the end node  $n_j$ , and set the relative direction of  $a$  as  $Y$ ;

(2) Searching for name correspondence.

(2.1) If the view type of  $n_i$  is *SECTIONAL*, determine whether a cutting line  $l_k^{n_j}$  of  $n_j$  is labelled the same as the section name of  $n_i$ .

(2.2) If so, construct an arc  $a$  of  $E_3$  with the start node  $n_j$  and the end node  $n_i$ , and align the relative direction of  $a$  with the first element of  $l_k^{n_j}$ .

(3) Searching for center correspondence.

(3.1) For each circle (circular arc) in  $n_i$ , which is with center point  $p_k$ ,

(3.1.1) Determine whether a center line  $l_q$  in  $n_j$  satisfies  $projectX(l_q) = projectX(p_k)$ . If so, construct an arc  $a$  of  $E_4$  with the start node  $n_i$  and the end node  $n_j$ , and set the relative direction of  $a$  as  $X$ .

(3.1.2) Determine whether a center line  $l_q$  in  $n_j$  satisfies  $projectY(l_q) = projectY(p_k)$ . If so, construct an arc  $a$  of  $E_4$  with the start node  $n_i$  and the end node  $n_j$ , and set the relative direction of  $a$  as  $Y$ .

(3.1.3) If  $a$  is constructed, terminate the search circulation.

Figure 7 illustrates the construction of VRG according to the segmented views in Figure 6. All the nodes are assigned the attribute vectors as shown in Figure 7(a). The relation information of the nodes are collected in the set of directed arcs in Figure 7(b). Consequently, the VRG is constructed (Figure 7(c)).

### 3.3 View relation validation

Each directed arc in the constructed VRG is uncertain for identifying the spatial adjacency of views. On the one hand, any one of the arcs connecting two nodes, no matter what type it is, can not solely confirm that the nodes are adjacent to each other in the 3D object space. On the other hand, some arcs involved in the constructed VRG may be pathological for revealing the spatial adjacency. Therefore, the number of view relations required for identifying view adjacency is not known in advance. In this step, we introduce a reasoning technique to determine valid view relations by combining geometric and symbolic evidence and applying a minimum spanning tree.

#### 3.3.1 Principles of evidence theory

Evidence theory, also known as Dempster-Shafer (D-S) theory, offers the capability to manage the uncertainty of different subsets of assertions in a domain [13]. Its ability to narrow the set of promising hypotheses with the accumulation of evidence is particularly useful for our purposes.

In evidence theory, a problem domain is represented by a finite set  $\Theta$  of hypotheses called the *frame of discernment*, assuming that the hypotheses under consideration are mutually exclusive and exhaustive [18].

The basic propagation of uncertainty information is through a *basic probability assignment*, which is carried out by a function  $m : 2^\Theta \rightarrow [0, 1]$  over  $\Theta$ , verifying  $m(\emptyset) = 0$  and  $\sum_{A \in 2^\Theta} m(A) = 1$ . For a subset  $A \in 2^\Theta$ ,  $m(A)$  is the measure of the belief that is committed exactly to  $A$ . If  $m(A) > 0$ ,  $A$  is called a *focal element* of  $m$ .  $m(\Theta)$  denotes the quantity of belief that remains unassigned.

Function  $Bel : 2^\Theta \rightarrow [0, 1]$  is said to be a *belief function* over  $\Theta$  if there exists an assignment function  $m$  over  $\Theta$  with  $Bel(A) = \sum_{B \subseteq A} m(B)$  for all  $A \in 2^\Theta$ .  $Bel(A)$  is the degree of belief in  $A$  as well as all of its subsets. Therefore, for each singleton  $B$  in  $\Theta$ ,  $Bel(B) = m(B)$ .

Let  $m_1$  and  $m_2$  be the assignment functions over  $\Theta$  given by two different evidence sources. They can be used by *Dempster's rule of combination* [18] to generate a new assignment function defined as, for  $\emptyset \neq A \in 2^\Theta$ ,

$$m_1 \oplus m_2(A) = \frac{1}{1 - K} \sum_{B, C: B \cap C = A} m_1(B)m_2(C),$$

where normalization constant  $K$  is given by

$$K = \sum_{B, C: B \cap C = \emptyset} m_1(B)m_2(C).$$

The sum of all assignments by  $m_1 \oplus m_2$  adds up to 1, thus satisfying the definition of basic probability assignment.

#### 3.3.2 Reasoning technique

The reasoning technique is based on determining the subset of view pairs that should validate the most useful relations in the VRG for revealing the spatial adjacency of various views. The following four steps are involved in the method.

1. Form hypothesis set of all the node pairs in the VRG.
2. Gather evidence according to the arcs in the VRG.
3. Combine evidence for each view pair.
4. Construct the view tree based on the belief ranking of the view pairs.

In stage (1), the frame of discernment used here is  $\Theta = \{(n_i, n_j) | i < j, i, j = 1, \dots, k\}$ , where  $k$  is the number of nodes in the VRG. All the elements in  $\Theta$  are founded by enumerating the combination of the nodes in the VRG. Enumeration considers each possible pair of views individually, which is practical due to the limited sets of views involved in the drawing. Therefore,  $\Theta$  satisfies the assumption of evidence theory. Then, the branch between a view pair to be validated can be represented by a tuple  $(n_i, n_j)$ , and a subset  $A \in 2^\Theta$  is the hypothesis that suggests the set of branches should be validated.

In stage (2), all the directed arcs in the VRG, which indicate the relations of collocation and notation among views,

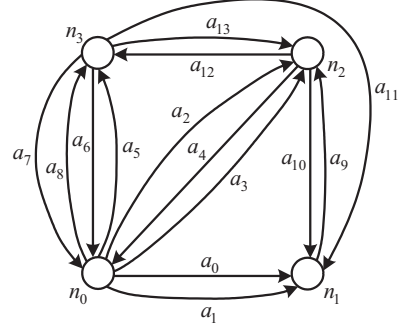


Node	$V(n)$	$S(n)$	$L(n)$			
			$i$	$L_n(n, i)$	$L_a(n, i)$	$L_v(n, i)$
$n_0$	REGULAR	null	0	"A"	1	3 portions
			1	"B"	0	1 portion
$n_1$	SECTIONAL	"A"	—			
$n_2$	SECTIONAL	"B"	—			
$n_3$	REGULAR	null	—			

(a)

Arc	$R(a)$	$N_s(a)$	$N_e(a)$	$D(a)$
$a_0$	$E_3$	$n_0$	$n_1$	Y
$a_1$	$E_4$	$n_0$	$n_1$	Y
$a_2$	$E_2$	$n_0$	$n_2$	Y
$a_3$	$E_3$	$n_0$	$n_2$	Y
$a_4$	$E_2$	$n_2$	$n_0$	X
$a_5$	$E_1$	$n_0$	$n_3$	X
$a_6$	$E_1$	$n_3$	$n_0$	X
$a_7$	$E_4$	$n_0$	$n_3$	X
$a_8$	$E_4$	$n_3$	$n_0$	X
$a_9$	$E_2$	$n_1$	$n_2$	Y
$a_{10}$	$E_2$	$n_2$	$n_1$	X
$a_{11}$	$E_2$	$n_3$	$n_1$	Y
$a_{12}$	$E_1$	$n_2$	$n_3$	Y
$a_{13}$	$E_1$	$n_3$	$n_2$	Y

(b)



(c)

**Figure 7: VRG construction. (a) Attributed nodes. (b) Attributed arcs. (c) Constructed VRG.**

form evidence helping support or helping exclude a one-element subset of  $\Theta$ . The impact of each evidence can be represented by a basic probability assignment. Since the conclusion as to what branches to be validated are based on the final *ranking* of the singletons in  $\Theta$  obtained from combined evidence. The exact value of the strength of evidence does not appear to be critical, as is typical for these applications [10, 12]. We employ the following assignments for the singleton subsets of  $\Theta$  based on the arcs in the VRG, which is chosen subjectively after examining a large number of drawings.

- If two views  $n_i$  and  $n_j$  are incident with an arc of parallel-size correspondence ( $E_1$ ), then  $m(\{(n_i, n_j)\}) = 0.3$  and  $m(\Theta) = 0.7$ .
- If two views  $n_i$  and  $n_j$  are incident with an arc of perpendicular-size correspondence ( $E_2$ ), then  $m(\{(n_i, n_j)\}) = 0.3$  and  $m(\Theta) = 0.7$ .
- If two views  $n_i$  and  $n_j$  are incident with an arc of name correspondence ( $E_3$ ), then  $m(\{(n_i, n_j)\}) = 0.8$  and  $m(\Theta) = 0.2$ .
- If two views  $n_i$  and  $n_j$  are incident with an arc of center correspondence ( $E_4$ ), then  $m(\{(n_i, n_j)\}) = 0.3$  and  $m(\Theta) = 0.7$ .

After making probability assignments corresponding to all the arcs in the VRG, we have a number of assignment functions each representing a separate piece of information. In stage (3), the combined assignment for each singleton of  $\Theta$  is calculated by repeatedly applying the combination rule on these assignment functions. The result of the combination is independent of the order in which the evidence are gathered and combined [12]. Therefore, a new assignment

for every singleton subset of  $\Theta$  are accumulated to represent the impact of the combined evidence, which is used to rank the singletons.

In stage (4), according to the ranking of the singletons of  $\Theta$ , we construct the view tree by *Kruskal's algorithm* for computing the minimum spanning tree to determine the branches to be validated. First, the view pair corresponding to highest ranked singleton is selected as the first branch of the view tree. Then, other views are sequentially spanned by adding the branch corresponding to the highest ranked singleton that has at least one of its terminal nodes outside the tree. Repeat until the tree is complete when all the nodes are spanned. The branches involved in the view tree are the hypotheses that correspond to the validated view pairs. Consequently, all the arcs that are overlaid by the validated branches are regarded as the most valuable view relations.

### 3.3.3 Example

This Section illustrate stepwise the above reasoning algorithm on the example in Figure 6. Considering the constructed VRG in Figure 7, the frame of discernment contains six pairs of nodes as follows.

$$\Theta = \{(n_0, n_1), (n_0, n_2), (n_0, n_3), (n_1, n_2), (n_1, n_3), (n_2, n_3)\}.$$

According to the arcs in the VRG, there are 14 pieces of evidence applicable to the hypothesis in  $\Theta$  with the following assignments.

$$a_0: m_0(\{(n_0, n_1)\}) = 0.8; m_0(\Theta) = 0.2.$$

$$a_1: m_1(\{(n_0, n_1)\}) = 0.3; m_1(\Theta) = 0.7.$$

$$a_2: m_2(\{(n_0, n_2)\}) = 0.3; m_2(\Theta) = 0.7.$$

$$a_3: m_3(\{(n_0, n_2)\}) = 0.8; m_3(\Theta) = 0.2.$$



$$a_4: m_4(\{(n_0, n_2)\}) = 0.3; m_4(\Theta) = 0.7.$$

$$a_5: m_5(\{(n_0, n_3)\}) = 0.3; m_5(\Theta) = 0.7.$$

$$a_6: m_6(\{(n_0, n_3)\}) = 0.3; m_6(\Theta) = 0.7.$$

$$a_7: m_7(\{(n_0, n_3)\}) = 0.3; m_7(\Theta) = 0.7.$$

$$a_8: m_8(\{(n_0, n_3)\}) = 0.3; m_8(\Theta) = 0.7.$$

$$a_9: m_9(\{(n_1, n_2)\}) = 0.3; m_9(\Theta) = 0.7.$$

$$a_{10}: m_{10}(\{(n_1, n_2)\}) = 0.3; m_{10}(\Theta) = 0.7.$$

$$a_{11}: m_{11}(\{(n_1, n_3)\}) = 0.3; m_{11}(\Theta) = 0.7.$$

$$a_{12}: m_{12}(\{(n_2, n_3)\}) = 0.3; m_{12}(\Theta) = 0.7.$$

$$a_{13}: m_{13}(\{(n_2, n_3)\}) = 0.3; m_{13}(\Theta) = 0.7.$$

After calculating  $m(A) = m_0 \oplus m_1 \oplus \dots \oplus m_{13}(A)$ , we obtain the following belief measure for the singleton subsets of  $\Theta$ , which represents the impact of the combined evidence.

$$m(\{(n_0, n_1)\}) = 0.278941; m(\{(n_0, n_2)\}) = 0.417948.$$

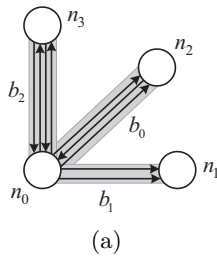
$$m(\{(n_0, n_3)\}) = 0.143716; m(\{(n_1, n_2)\}) = 0.047262.$$

$$m(\{(n_1, n_3)\}) = 0.019461; m(\{(n_2, n_3)\}) = 0.047262.$$

And then, we obtain the following ranking of the singleton subsets.

$$\{(n_0, n_2)\}, \{(n_0, n_1)\}, \{(n_0, n_3)\}, \{(n_1, n_2)\}, \{(n_2, n_3)\}, \{(n_1, n_3)\}.$$

The highest ranked singleton  $\{(n_0, n_2)\}$ , is selected to initialize the tree, and the corresponding branch validates all the arcs that connect the node pair, i.e.  $\{a_2, a_3, a_4\}$ . Until all the nodes are spanned, singletons  $\{(n_0, n_1)\}$  and  $\{(n_0, n_3)\}$  are sequentially added into the tree according to their higher ranking, and arc sets  $\{a_0, a_1\}$  and  $\{a_5, a_6, a_7, a_8\}$  are validated, respectively. The resulting view tree is shown in Figure 8(a), with the validated arcs listed in Figure 8(b). Therefore, the validated arcs are considered as the most valuable relations for identifying the spatial adjacency of the views.



Rank	Branch	Node pair	Validated arcs
1	$b_0$	$(n_0, n_2)$	$a_2, a_3, a_4$
2	$b_1$	$(n_0, n_1)$	$a_0, a_1$
3	$b_2$	$(n_0, n_3)$	$a_5, a_6, a_7, a_8$

(b)

**Figure 8: View relation validation. (a) View tree. (b) Validated view relations.**

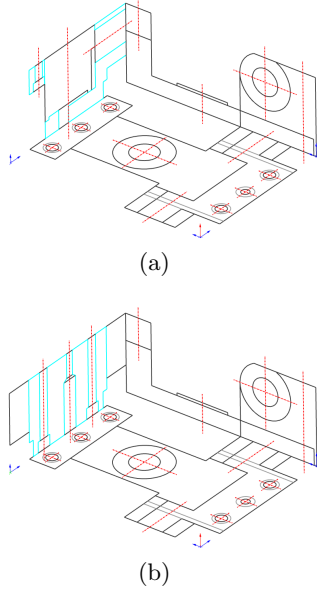
### 3.4 View-plane folding

In this step, we deduce the projection plane for each view by travelling the branches of the view tree. First, one view is selected as the root of the tree according to the typical layout of a drawing. We employ the following rules to find the root.

- (1) If there are only two views in the drawing, select the upper view as the root; otherwise, select the left one.
- (2) If there are more than two views in the drawing, the view with the greatest number of adjacent views is selected as the root view.

Then, the root is considered as the front view by defining  $xz$ -plane as its projection plane, and the projection planes of the other views are consequently confirmed by tracing down the view tree in breadth-first mode. After deciding the projection plane of a node  $n_h$ , the branches at  $n_h$  are considered in turn. For each branch, the valid arc that has the greatest probability assignment is selected as the current arc  $a_c$ , and the other end of  $a_c$  is considered as the current node  $n_c$ . Considering  $n_h$  and  $a_c$ , the projection plane of  $n_c$  is decided as follows, where the relative positions of the nodes are with respect to the 2D drawing-based coordinate system.

- (1) If the projection plane of  $n_h$  is *FRONT*,
  - (1.1) If the relative direction of  $a_c$  is  $X$ ,
    - (1.1.1) If  $n_c$  is on the bottom side of  $n_h$ , set the projection plane of  $n_c$  as *TOP*.
    - (1.1.2) Otherwise, set the projection plane of  $n_c$  as *BOTTOM*.
  - (1.2) If the relative direction of  $a_c$  is  $Y$ ,
    - (1.2.1) If  $n_c$  is on the right side of  $n_h$ , set the projection plane of  $n_c$  as *LEFT*.
    - (1.2.2) Otherwise, set the projection plane of  $n_c$  as *RIGHT*.
- (2) If the projection plane of  $n_h$  is *TOP*,
  - (2.1) If the relative direction of  $a_c$  is  $X$  and  $n_c$  is on the bottom side of  $n_h$ , set the projection plane of  $n_c$  as *REAR*.
  - (2.2) If the relative direction of  $a_c$  is  $Y$ ,
    - (2.2.1) If  $n_c$  is on the right side of  $n_h$ , set the projection plane of  $n_c$  as *LEFT*.
    - (2.2.2) Otherwise, set the projection plane of  $n_c$  as *RIGHT*.
- (3) If the projection plane of  $n_h$  is *LEFT*,
  - (3.1) If the relative direction of  $a_c$  is  $X$ ,
    - (3.1.1) If  $n_c$  is on the bottom side of  $n_h$ , set the projection plane of  $n_c$  as *TOP*.
    - (3.1.2) Otherwise, set the projection plane of  $n_c$  as *BOTTOM*.
  - (3.2) If the relative direction of  $a_c$  is  $Y$  and  $n_c$  is on the right side of  $n_h$ , set the projection plane of  $n_c$  as *REAR*.



**Figure 10: View-plane folding: The folded views on the same projection plane (i.e.,  $n_1$  and  $n_2$ ) are shown apart to avoid overlay. (a) Folded views  $n_0$ ,  $n_3$ , and  $n_1$  (offset section “A–A”). (b) Folded views  $n_0$ ,  $n_3$ , and  $n_2$  (full section “B–B”).**

According to respective projection planes, all the views are folded in the 3D object space. As specified in Figure 1(b), each view of the drawing has its own view-based origin and coordinate axes. The act of transforming views from the drawing-based coordinate system to the corresponding view-based coordinates involves translations and rotations. For each view, a transformation matrix is set corresponding to its origin and coordinate axes, and then the coordinate transformation can be achieved by matrix multiplication.

Figure 9 illustrates stepwise the decision of projection planes for the nodes in the constructed tree (Figure 8). Since  $n_0$  has the greatest number of adjacent nodes, it is first selected as the root, i.e. the front view (Figure 9(a)). With that, the branches at  $n_0$  are considered in turn. In Figure 9(b), the first branch  $b_0$  is considered, and the validated arc  $a_3$  is selected as the current arc. Since the relative direction of  $a_3$  is  $Y$ , the current node  $n_2$  is determined to be the left view. In the same manner, the projection planes of  $n_1$  and  $n_3$  are identified in Figure 9(c) and Figure 9(d), respectively. Finally, each view is transformed from the plane of drawing sheet to respective view-plane, and the folded views are shown in Figure 10.

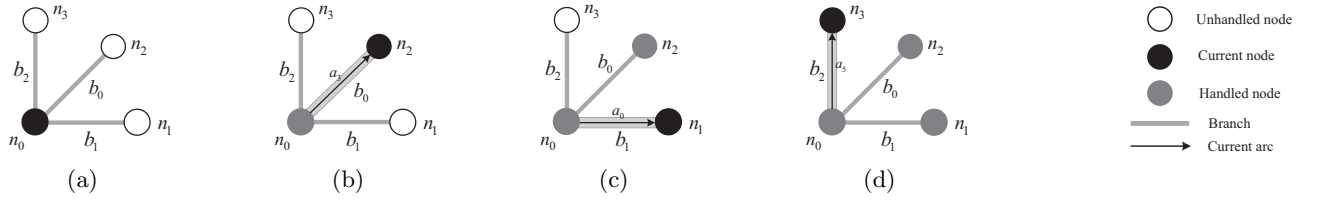
### 3.5 Cutting plane generation

Based on the folded views, this step is to generate the 3D information about the cutting planes of the object. In the drawing, a cutting line shows where the object was cut to obtain the section view, and hatching in the section view shows the solid surface of the object which were cut through to produce the section. Therefore, the validated arcs of *name correspondence* ( $E_3$ ) are important clues for filtering and transforming hatching loops of the sectional views into the 3D object space. The procedure is shown as follows, where  $projectA((l_s, l_e))$  is the projected inter-zone on  $A$ -axis of a

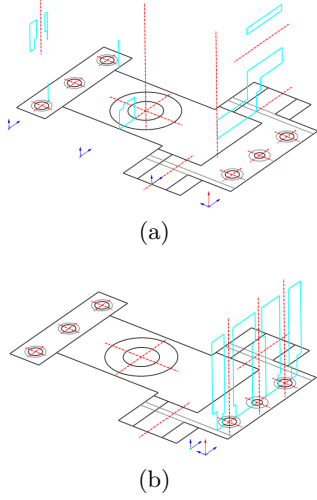
valid portion  $(l_s, l_e)$  of a cutting line,  $projectA(p)$  for the projection of a point  $p$ .

- (1) For each validated arc  $a_i$  of name correspondence,
  - (1.1) Retrieve the start node  $n_s$  and the end node  $n_e$  of  $a_i$ .
  - (1.2) Obtain the relative direction  $D$  of  $a_i$ , with the other axis  $A$  of the 2D drawing-based coordinate system recorded.
  - (1.3) Retrieve the origin  $p_o$  of  $n_e$ .
  - (1.4) Find the cutting line  $l_j^{n_s}$  of  $n_s$  that is labelled the same as the section name of  $n_e$ .
  - (1.5) For each valid portion  $(l_{ks}, l_{ke})$  of  $l_j^{n_s}$ ,
    - (1.5.1) If  $(l_{ks}, l_{ke})$  is parallel to  $D$ -axis,
      - (1.5.1.1) Calculate the signed distance  $d$  from  $projectA((l_{ks}, l_{ke}))$  to  $projectA(p_o)$ , and then translate  $p_o$  by  $d$  along  $A$ -axis.
      - (1.5.1.2) Set a transformation matrix  $M$  with the coordinate axes of  $n_e$  and the translated  $p_o$ .
      - (1.5.1.3) Set a filter  $f$  as  $projectD((l_{ks}, l_{ke}))$  on  $D$ -axis.
    - (1.5.2) Otherwise,
      - (1.5.2.1) Retrieve the terminal point  $p_c$  of  $(l_{ks}, l_{ke})$  that locates on the center of certain circle of  $n_s$ , with the other terminal point  $p_t$  of  $(l_{ks}, l_{ke})$  recorded.
      - (1.5.2.2) Calculate the signed distance  $d$  from  $projectA(p_c)$  to  $projectA(p_o)$ , and then translate  $p_o$  by  $d$  along  $A$ -axis.
      - (1.5.2.3) Calculate the angle  $\alpha$  between  $(l_{ks}, l_{ke})$  and  $D$ -axis.
      - (1.5.2.4) Set a transformation matrix  $M$  with the coordinate axes of  $n_e$ , the translated  $p_o$ , and the rotated angle  $\alpha$ .
      - (1.5.2.5) Calculate the signed distance  $d_s$  from  $p_t$  to  $p_c$ , and then set a filter  $f$  from  $projectD(p_c)$  extended  $d_s$  along  $D$ -axis.
  - (1.5.3) For each hatching loop  $l_h$  of  $n_e$ ,
    - (1.5.3.1) Determine whether some portion of  $l_h$  filters through  $f$ .
    - (1.5.3.2) If a portion or the whole of  $l_h$  is identified, multiply their coordinates by  $M$  to transform them into the object space.
  - (1.5.4) For each center line  $l_c$  of  $n_e$ ,
    - (1.5.4.1) Determine whether some portion of  $l_c$  filters through  $f$ .
    - (1.5.4.2) If  $l_c$  is identified, multiply the coordinates of its terminals by  $M$  to transform them into the object space.

Referring to Figure 8, the front view  $n_0$  is incident with two validated arcs of  $E_3$ , i.e.,  $a_0$  and  $a_3$ . In Figure 11(a), the first cutting line of  $n_0$  (Figure 7(a)), which corresponds to  $a_0$ , is handled to create three cutting planes, and the hatching loops of the adjacent node  $n_1$  (the offset section “A–A”) are filtered. Figure 11(b) shows the cutting plane identified in the same manner, according to the second cutting plane of  $n_0$ .



**Figure 9: View-plane decision.** (a)  $P(n_0) = \text{FRONT}$ . (b)  $P(n_2) = \text{LEFT}$ . (c)  $P(n_1) = \text{LEFT}$ . (d)  $P(n_3) = \text{BOTTOM}$ .



**Figure 11: Cutting plane generation.** (a) Three cutting planes are identified based on the front view and offset section “A–A”. (b) One cutting plane is identified based on the front view and full section “B–B”.

## 4. IMPLEMENTATION AND DISCUSSION

The implementation of the proposed algorithm was tested as an input module of the commercial CAD system TiGEMS<sup>®</sup> 6.0 that is owned by Tsinghua Software Information Technology Co. Ltd. The developed program accepts drawing files in CAD neutral format that consist of one or more sectional views, allowing that the number of views, the types of sections, as well as the view layout are unknown a priori.

### 4.1 Examples

Besides the example of Figure 6, two more examples (Figures 12–13) are provided to demonstrate various cases that can be handled by our method. They were tested on a personal computer platform with an Intel Pentium IV 2.40GHz CPU and 512MB RAM. The corresponding cutting planes were automatically identified within one second of CPU time.

Figure 12 shows the result of identifying several offset sections in the same direction. The input drawing is shown in Figure 3. The view in the left side of the drawing is defined as *FRONT*, which is with the greatest number of adjacent views in the constructed view tree. While the projection planes of the sections are all *LEFT*. Consider the arc of name correspondence that connects the front view and section “A–A”. There are two valid portions of the cutting line “A” in the front view, thus two cutting planes are one-to-one identified corresponding to section “A–A”, as shown in Figure 12(a).

In the same manner, three cutting planes are generated from folded section “B–B” (Figure 12(b)), and two cutting planes from folded section “C–C” (Figure 12(c)). All these cutting planes are parallel.

Figure 13 shows an example with all three types of sections. The input drawing that contains three views is given in Figure 13(a). After evidence-based reasoning, the view on the right-bottom of the drawing has the greatest number of adjacent views. It is regarded as the main view, i.e. on *FRONT* projection plane. And then, the projection planes of the other two views are accordingly defined: section “A–A” is defined as *BOTTOM* and section “B–B” is on *RIGHT* projection plane. Consequently, all the views are respectively transformed into the 3D coordinate system as shown in Figure 13(b). Two arcs of name correspondence ( $E_3$ ) are validated to connect the main view to the sectional views, respectively. According to the arc that is incident on section “A–A”, four cutting planes are generated: three are offset and the other one is angled, as shown in Figure 13(c). In the same manner, Figure 13(d) shows the plane that is generated to cut through the object, which corresponds to section “B–B”.

### 4.2 Discussion

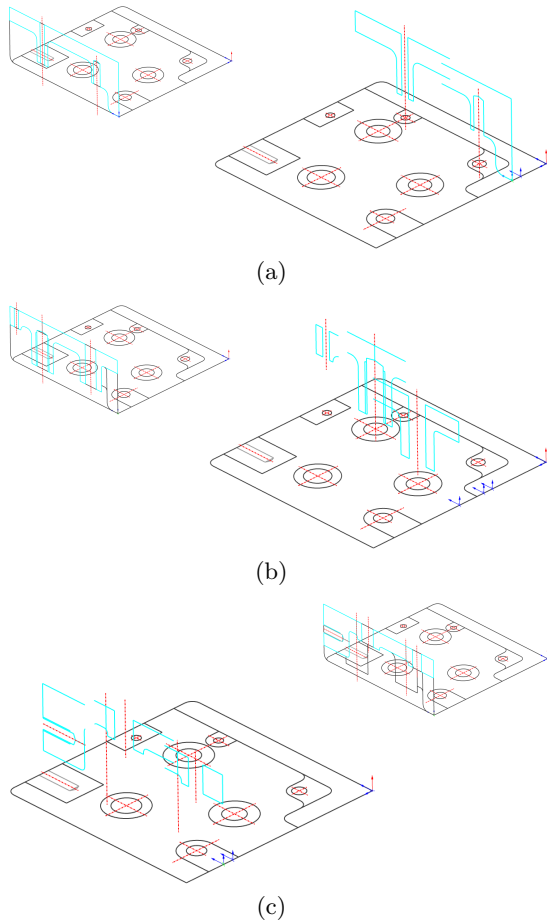
As can be seen, the proposed method is competent for engineering drawings that contain full sections, offset sections, aligned sections, and regular views. In all cases, the views and sections in diverse configurations have been successfully identified from the 2D drawing space and then accurately transformed into the 3D object space.

In addition to the three kinds of sections, there are other types of sectional views used in computer-aided design and drafting [11].

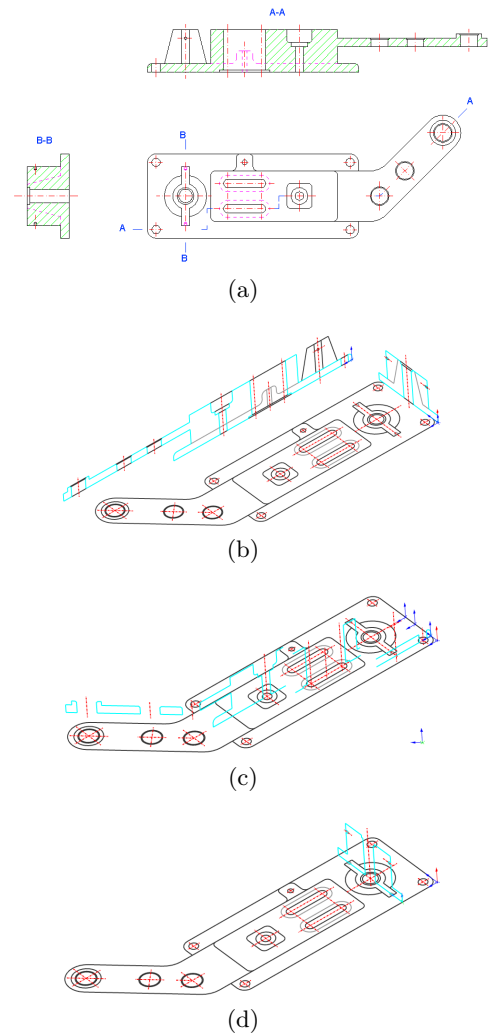
A *half section* is obtained when a cutting plane passes half-way through an object. The sectional view shows half of the object in section and the other half of the object as it would normally appear. The proposed method can be directly extended to identify the cutting plane of a half section by allowing a cutting line to terminate inside the view, but the usefulness of half sections is largely limited to symmetric objects.

A *broken-out section* is that only a partial section of a view is drawn to clarify an interior feature of an object as if a small portion of the object were broken away. In the view, the sectional portion is irregularly separated from the un-sectioned portion by a freehand break curve, and there is generally no cutting line used.

A *revolved section* is produced by cutting through an object and then revolving the cross-sectional shape by  $90^\circ$  onto a plane perpendicular to the line of sight. The revolved section may be drawn on the top of the existing view, or it may be broken away and isolated from the regular parts. *Re-*



**Figure 12: Identification of several offset sections in the same direction. (a) Two cutting planes corresponding to offset section “A–A” (left-top). (b) Three cutting planes corresponding to offset section “B–B” (left-top). (c) Two cutting planes corresponding to offset section “C–C” (right-top).**



**Figure 13: Identification of all three types of sections. (a) Input drawing. (b) Three folded views. (c) Four cutting planes corresponding to offset and aligned section “A–A”. (d) One cutting plane corresponding to full section “B–B”.**

*moved sections* are similar to revolved sections except that they are removed from the view. The removed section is generally not placed in direct alignment with the cutting line.

The broken-out section, revolved section, and removed section are not positioned in agreement with the direct projection from the cutting plane. The key task should be to use the knowledge of these kinds of sectional views to identify additional sections in the absence of projection relations.

## 5. CONCLUSION

A new method of labelling views from engineering drawings has been presented based on evidence theory. The main contribution of this work is the application to engineering drawings with sectional views to identify full sections, offset sections, and aligned sections. A graph representation (VRG) has been developed for describing the multiple relationships among various views, and a reasoning technique has been implemented for validating a set of view relations that are used to fold views and sections. The implementation of the proposed method has been tested on a variety of section-involved drawings and produced good results for multiple sections in different arrangements.

The presented method significantly extends the range of practical drawings that can be automatically identified due to its two advantages over other methods. Firstly, as a basis for view identification, the VRG makes the view relationships required for labelling sections much more accessible. Secondly, the evidence-based reasoning accommodates the diverse layout of views which arises from multiple sections involved in the drawing.

The current implementation, as the first solution to the problem of labelling various sections, does not fully validate some other kinds of sections. Future work will extend the approach to identify additional types of sectional views. The proposed framework based on evidence theory builds a promising basis for further expansibility of view identification, and it could be extended by expanding view relations of the VRG and correspondingly enhancing the reasoning technique.

## 6. REFERENCES

- [1] B. Aldefeld and H. Richter. Semiautomatic three-dimensional interpretation of line drawing. *Computers and Graphics*, 8(4):371–380, 1984.
- [2] H. Bin. Inputting constructive solid geometry representations directly from 2D orthographic engineering drawings. *Computer-Aided Design*, 18(3):147–155, 1986.
- [3] D. X. Chen. *Technical Drafting and CAD Elements*. Mechanical Industry Press, Beijing, 2004.
- [4] J. Dimri and B. Gurumoorthy. Handling sectional views in volume-based approach to automatically construct 3D solid from 2D views. *Computer-Aided Design*, 37(5):485–495, 2005.
- [5] D. Dori and K. Tombre. From engineering drawings to 3D CAD models: Are we ready now? *Computer-Aided Design*, 27(4):243–254, 1995.
- [6] W. D. Geng, J. B. Wang, and Y. Y. Zhang. Embedding visual cognition in 3D reconstruction from multi-view engineering drawings. *Computer-Aided Design*, 34(4):321–336, 2002.
- [7] J. H. Gong, H. Zhang, G. F. Zhang, and J. G. Sun. Solid reconstruction using recognition of quadric surfaces from orthographic views. *Computer-Aided Design*, 38(8):821–835, 2006.
- [8] C. Hubbard and Y. S. Kim. Geometric assistance for visual reasoning and construction of solids with curved surfaces from 2D orthographic views. *Advances in Engineering Software*, 32(1):21–35, 2001.
- [9] S. X. Liu, S. M. Hu, Y. J. Chen, and J. G. Sun. Reconstruction of curved solids from engineering drawings. *Computer-Aided Design*, 33(14):1059–1072, 2001.
- [10] D. B. Lysak, P. M. Devaux, and R. Kasturi. View labeling for automated interpretation of engineering drawings. *Pattern Recognition*, 28(3):393–407, 1995.
- [11] D. A. Madsen. *Engineering Drawing and Design, third edition*. Thomson Learning Inc., New York, 2004.
- [12] M. Marefat and R. L. Kashyap. Geometric reasoning for recognition of three-dimensional object features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):949–965, 1990.
- [13] G. Shafer. The Dempster-Shafer theory. In *Encyclopedia of Artificial Intelligence, 2nd edition*, pages 330–331, New York, 1992. John Wiley and Sons.
- [14] S. Soni and B. Gurumoorthy. Handling solids of revolution in volume-based construction of solid models from orthographic views. *Journal of Computing and Information Science in Engineering*, 3:250–259, 2003.
- [15] M. Tanaka, L. Anthony, T. Kaneeda, and J. Hirooka. A single solution method for converting 2D assembly drawings to 3D part drawings. *Computer-Aided Design*, 36(8):723–734, 2004.
- [16] K. Tombre. Analysis of engineering drawings: State of the art and challenges. In *Lecture Notes in Computer Science*, volume 1389, pages 257–264, 1997.
- [17] M. A. Wesley and G. Markowsky. Fleshing out projections. *IBM Journal of Research and Development*, 25(6):934–953, 1981.
- [18] N. Wilson. Algorithms for Dempster-Shafer theory. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*, pages 421–475, Boston, 2000. Kluwer Academic Publishers.
- [19] C. F. You and S. S. Yang. Reconstruction of curvilinear manifold objects from orthographic views. *Computers and Graphics*, 20(2):275–293, 1996.