What the Back of the Object Looks Like: 3D Reconstruction from Line Drawings Without Hidden Lines

Liangliang Cao, Jianzhuang Liu, *Senior Member, IEEE,* and Xiaoou Tang, *Senior Member, IEEE*

***Abstract*— The human vision system can interpret a single 2D line drawing as a 3D object without much difficulty even if the hidden lines of the object are invisible. Many reconstruction methods have been proposed to emulate this ability, but they cannot recover the complete object if the hidden lines of the object are not shown. This paper proposes a novel approach to reconstructing a complete 3D object, including the shape of the back of the object, from a line drawing without hidden lines. First, we develop theoretical constraints and an algorithm for the inference of the topology of the invisible edges and vertices of an object. Then we present a reconstruction method based on perceptual symmetry and planarity of the object. We show a number of examples to demonstrate the success of our approach.**

***Index Terms*— 3D reconstruction, hidden topology, line drawings, visual perception.**

I. Introduction

A line drawing is defined as a 2D projection of the edges and vertices of a 3D object in a generic view, with or without hidden lines visible. The human vision system has the ability to interpret 2D line drawings as 3D objects without difficulty. Emulating this ability is an important research topic in computer vision and is finding applications in a diverse range of areas, such as providing 2D sketch query interface for 3D object retrieval from large databases or from the web [1], [2], [3], interactive generation of 3D models from images [4], [5], [6], providing rich databases to object recognition systems and reverse engineering algorithms for shape reasoning [7], [8], [9], and flexible sketching interface for designers to sketch ideas in object design [10], [11], [12].

There has been much effort devoted to line drawing interpretation. However, few study the inference of the hidden topology and the reconstruction of the complete 3D objects from line drawings with only visible edges and vertices. This is a challenging task since we have to infer the most plausible shape of the back of the object without looking at it. This paper proposes a novel approach to tackle this hard problem. Compared with the line drawings with hidden lines visible, these line drawings are easier and more natural to draw. More importantly, most vision systems do not provide the hidden lines, and thus our work is more suitable for general vision applications. For example, if we want to recover the complete 3D shape of an object in an image, we do not have the invisible edges to work with. Based on the visual perception of such line drawings, our algorithm is designed in two steps: inference of the topology of the hidden part of an object and reconstruction of the geometrical shape of the complete object. Fig. 1 shows the two steps with an example.

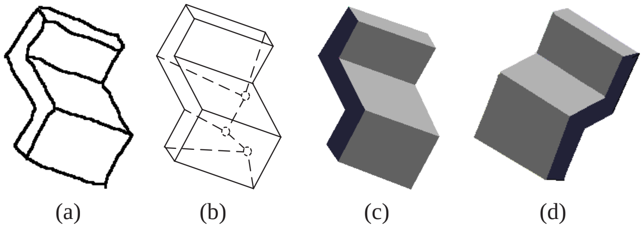

Fig. 1. (a) A line drawing without hidden lines. (b) The inferred hidden topological structure (dotted lines and circles) before reconstruction. (c) Reconstructed complete 3D object. (d) The 3D object in another view.

The rest of this paper is organized as follows. Section II reviews related work. Section III gives assumptions for our work and terms that are often used in the paper. We present theoretical constraints for the inference of the hidden structure from a line drawing in Section IV. The algorithmic implementation of the inference is discussed in Section V. Section VI proposes an optimization-based method for the complete 3D object reconstruction. Experimental results are given in Section VII. Finally, Section VIII concludes this paper.

II. Related Work

Since the early stage of computer vision, a large amount of work called line labeling has been carried out for line drawing interpretation [13], [14], [15], [16], [17], [18], [19]. Line labeling focuses on finding a set of consistent labels from a line drawing without hidden lines, and does not explicitly give the 3D shape represented by a line drawing.

There is a body of work that discusses 3D reconstruction from multiple views of a line drawing. Its target is to reconstruct a 3D CAD model from its multiple (three, in general) orthographic projections [20], [21], [22]. More information can be found from three orthographic views for the reconstruction task than from a single projection. In this paper, we concern the 3D reconstruction from only one view of an object.

Some related work is about testing the correctness of a line drawing based on algebraic or geometric tests [23], [24], [25], [26], [27] that apply to a line-labelled version of the line drawing. The algorithm presented in [28] tries to overcome the superstrictness problem in this line drawing interpretation.

Manuscript received ???, 2006; revised ???, 2007.

L. Cao is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. Email: cao4@uiuc.edu.

J. Liu is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. Email: jzliu@ie.cuhk.edu.hk.

X. Tang is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. Email: xtang@ie.cuhk.edu.hk.

Most 3D reconstruction methods from a line drawing (with or without hidden lines and vertices) assume that the face topology of the line drawing has been known in advance. This information can greatly reduce the complexity of the reconstruction. Face identification from line drawings is not a trivial problem, and many algorithms have been proposed for it [29], [30], [31], [32], [33]. The algorithm in [29] can handle only very simple objects, and the one in [32] focuses on manifolds with all hidden lines visible. The three algorithms in [30], [31], and [33] deal with general objects, but the one in [33] is most efficient. In this work, we use it to find the face cycles from a line drawing.[1]

The ultimate target of line drawing interpretation is to reconstruct the 3D objects from 2D line drawings. To this end, many researchers formulated the problem as an optimization problem based on different objective functions. Marill proposed a criterion, minimizing the standard deviation of the angles between all pairs of lines meeting at vertices in a reconstructed object, to emulate human 3D perception of 2D line drawings [34]. This idea is followed by many researchers [5], [10], [12], [29], [35], [36], [37]. With the help of the shading information in images, Sugihara [38], Shimshoni and Ponce [39], and Shimodaira [40] tried to recover 3D polyhedra in the images from the edges (line drawings) of the polyhedra. These reconstruction methods [34], [5], [10], [12], [29], [35], [36], [37], [38], [39], [40] cannot recover complete 3D objects if their hidden lines are not given. Varley and Martin [41] attempted to find the hidden topology of a line drawing representing a manifold polyhedron. However, they had to assume that the 3D geometry of the visible part of the polyhedron has been obtained, and the polyhedron has been assigned to one of the several regular categories, which can be very difficult to determine when only the visible part of the object is given.

III. Assumptions and Terminology

Although the 2D projection of a 3D object has lost the depth information of the 3D vertices and edges, human beings still have the ability to perceive the complete 3D shape from the line drawing even if its hidden lines are invisible. Emulating this function presents a very hard problem. One reason is that there are infinite possible structures being the hidden part of the object. Therefore, we have to impose reasonable constraints for the inference of the hidden structure such that the final recovered 3D object is in accordance with our perception from the line drawing. In this work, we focus on a class of solids that covers a large set of common objects.

Assumption 1: The 3D objects are polyhedra with each vertex met by three edges and each edge passed through by two faces, and without through holes.

Assumption 2: A line drawing is the parallel or near-parallel projection of the visible edges and vertices of a single polyhedron defined above in a generic view.

Assumption 3: Every hidden vertex is connected with at least one visible vertex.

In Assumption 2, a line drawing is said to be the projection of a polyhedron in a *generic* view if the topology of the line drawing is preserved under slight variations of the viewpoint. The reason to have Assumption 3 is that if all the three invisible edges meeting at a hidden vertex are allowed to connect to other hidden vertices, there will be infinite possible structures being the hidden part. This assumption restricts the infinite structures to limited simpler cases. Gestalt psychology, one of the most influential theories with a long history, asserts that human beings are innately driven to perceive things as good a whole as possible. Here *good* can mean many things such as simplicity, symmetry, and regularity [42], [43]. Assumption 3 reflects the perception of simplicity by human beings.

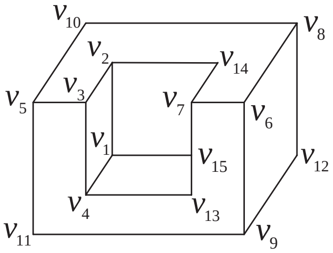

Fig. 2. Illustration of some terms. Here v_{1-9} are complete vertices, v_{10-14} are incomplete vertices, and v_{15} is a broken vertex. Edge (v_{11}, v_9) is a boundary edge, and (v_1, v_{15}) is both a broken edge and a zero edge. Cycle $(v_1, v_2, v_3, v_4, v_1)$ is a visible face, and cycle $(v_{10}, v_5, v_{11}, v_9, v_{12}, v_8, v_{10})$ is a boundary cycle.

For easier understanding of the technical content of our approach, we summarize the terms that will be used in the rest of the paper. Some of these terms are illustrated with Fig. 2.

- *Edge.* An edge of a line drawing is the intersection of two non-coplanar planes.
- *Degree.* The degree $d(v)$ of a vertex v is the number of edges meeting at v in a line drawing.
- *Incomplete vertex.* An incomplete vertex v is a vertex of $d(v) = 2$.
- *Complete vertex.* A complete vertex v is a vertex of $d(v) = 3$.
- *Broken vertex.* A broken vertex v is defined as one formed when part of an edge is blocked by a face. It is not a real vertex of the object the line drawing represents, and its degree is defined as $d(v) = 1$.
- *Broken edge.* A broken edge is a visible edge connected to a broken vertex.
- *Cycle.* A cycle is a closed trail in a line drawing where all its vertices except the end vertices are distinct.
- *Visible face.* A visible face is a face bounded by a cycle where all its edges are visible.
- *Rank.* The rank $R(e)$ of an edge e is the number of visible faces with edges passing through e.
- *Zero edge.* A zero edge e is an edge of $R(e) = 0$. A broken edge is also a zero edge.
- *Boundary edge.* A boundary edge e is an edge of $R(e) = 1$.
- *Boundary cycle.* A boundary cycle is a cycle where all its edges are boundary edges.
- *Hidden cycle.* A hidden cycle is a cycle where all its edges and vertices are invisible.
- N_H. N_H denotes the number of hidden vertices.

IV. Constraints for Inferring the Hidden Topological Structure

At first, we compute the degrees of vertices and the ranks of edges from a given line drawing. There are three types of vertices

[1] For the line drawings discussed in this paper, since the hidden lines are invisible, the task of face identification becomes easier, and it is possible to develop simpler algorithms for it.

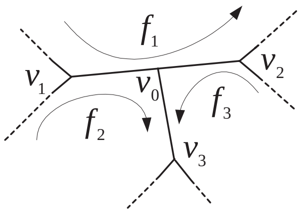

Fig. 3. Part of a line drawing where v_1, v_0 and v_2 are collinear.

as defined in Section III. Incomplete vertices are easy to find, but to distinguish broken vertices from complete vertices is not obvious. The following theorem allows us to identify the broken vertices.

Theorem 1: If a vertex v_0 touches a straight line in a line drawing as shown in Fig. 3, then v_0 is a broken vertex.

Proof: Suppose, to the contrary, that v_0 is not a broken vertex. Then it is a complete vertex. With the assumption that every edge of the object is passed through by two faces, there are three planar faces passing through v_0. Let them be $f_1 = (v_1, v_0, v_2, ..., v_1), f_2 = (v_1, v_0, v_3, ...v_1)$ and $f_3 = (v_2, v_0, v_3, ..., v_2)$, as shown in Fig. 3. According to the assumption that the line drawing is the projection of a polyhedron in a generic view, the three vertices v_1, v_0, v_2 are also collinear in 3D space. Thus, the straight line (v_1, v_2) and the vertex v_3 that is not on this line define a plane in 3D space, implying that the two faces f_2 and f_3 are coplanar, which contradicts the definition that an edge ((v_3, v_0) here) is the intersection of two non-coplanar faces. Therefore, if edge (v_3, v_0) is a visible edge of the object, v_0 is a broken vertex. ■

Knowing the types of all the vertices, we obtain their degrees immediately. To find the ranks of the edges, we have to find the visible faces first. The algorithm published in [33] is used for this purpose. For example, the four visible faces found from the line drawing shown in Fig. 2 are $(v_1, v_2, v_3, v_4, v_1)$, $(v_6, v_9, v_{12}, v_8, v_6)$, $(v_2, v_3, v_5, v_{10}, v_8, v_6, v_7, v_{14}, v_2)$, and $(v_4, v_3, v_5, v_{11}, v_9, v_6, v_7, v_{13}, v_4)$. Then we have $R(v_2, v_3) = R(v_3, v_4) = R(v_3, v_5) = R(v_6, v_7) = R(v_6, v_8) = R(v_6, v_9) = 2$, and $R(v_1, v_2) = R(v_1, v_4) = R(v_2, v_{14}) = R(v_7, v_{14}) = R(v_5, v_{10}) = R(v_{10}, v_8) = R(v_8, v_{12}) = R(v_{12}, v_9) = R(v_9, v_{11}) = R(v_{11}, v_5) = R(v_7, v_{13}) = 1$. Since v_{15} is a broken vertex, we cannot find the visible faces passing through (v_1, v_{15}) at this stage. Thus, $R(v_1, v_{15}) = 0$. Note that when v_{15} is detected to be a broken vertex, we have $d(v_{15}) = 1$ and the line from v_7 to v_{13} is one edge but not two.

The key to the inference of the invisible vertices and edges is to determine the number of hidden vertices N_H and the connections among invisible, incomplete, and broken vertices. The following theorems give constraints useful for the inference.

Theorem 2: Let $\mathcal{V_I}$ and $\mathcal{V_B}$ be the sets of incomplete and broken vertices of a line drawing, respectively. Then we have

$$N_H \leq |\mathcal{V_I}| + |\mathcal{V_B}|, \tag{1}$$

where $|\cdot|$ denotes the number of elements in a set.

Proof: A complete vertex does not connect to any hidden vertex. An incomplete vertex connects to one hidden vertex. A broken vertex connects to one hidden vertex too. From Assumption 3, the largest value of N_H appears when all the hidden vertices connect to different visible vertices, which implies the inequality in (1). ■

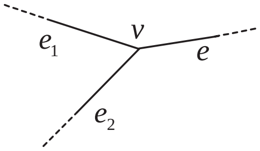

Fig. 4. Part of a line drawing where e is a boundary edge and v is a complete vertex.

The following Lemma [44] is used to prove Theorem 3.

Lemma 1: Let $\mathcal{G}$ be any graph, $\mathcal{E}$ be the set of edges and $\mathcal{V}$ be the set of vertices in $\mathcal{G}$. It holds that

$$\sum_{v \in \mathcal{V}} d(v) = 2|\mathcal{E}|. \tag{2}$$

Theorem 3: Let $\mathcal{V_I}$ and $\mathcal{V_B}$ be the sets of incomplete and broken vertices of a line drawing, respectively. If $|\mathcal{V_I}| + |\mathcal{V_B}|$ is even (odd), then N_H of the line drawing must be even (odd).

Proof: Suppose that there are N_H hidden vertices and l hidden edges in the object the line drawing represents. If we construct a graph using all the hidden edges, hidden vertices, incomplete vertices, and broken vertices (without all the visible edges), from Lemma 1 we have

$$\sum_{v \in \mathcal{V_I}} 1 + \sum_{v \in \mathcal{V_B}} 1 + \sum_{v \in \mathcal{V_H}} 3 = 2l, \tag{3}$$

where $\mathcal{V_H}$ is the set of hidden vertices and $|\mathcal{V_H}| = N_H$. The above equation can be rewritten as $|\mathcal{V_I}| + |\mathcal{V_B}| + N_H = 2(l - N_H)$. Therefore, if $|\mathcal{V_I}| + |\mathcal{V_B}|$ is even (odd), N_H must be even (odd) too. ■

Theorem 4: Given a line drawing, we have

$$R(e_i) < d(v), \tag{4}$$

and

$$\sum_{i=1}^{d(v)} R(e_i) = 2F, \tag{5}$$

where e_i, $i = 1, 2, ..., d(v)$, are the edges adjacent to vertex v, and F is the number of visible faces passing through v.

Proof: If v is a broken vertex, then e_i is a zero edge and $R(e_i) = 0$. Thus $R(e_i) < d(v) = 1$. If v is an incomplete vertex, then $R(e_i) \leq 1$ and $R(e_i) < d(v) = 2$. If v is a complete vertex, then $R(e_i) \leq 2$ and $R(e_i) < d(v) = 3$. Thus the inequality in (4) holds.

Since a visible face passing through v passes through two of the $e_i, i = 1, 2, ..., d(v)$, and $R(e_i)$ is the number of visible faces passing through e_i, we have the equation in (5) immediately. ■

Theorem 5: If a line drawing has no zero edges, all the boundary edges must be on one of the boundary cycles.

Proof: We divide the proof into three parts.

(**i**) We first show that every incomplete vertex is connected with two boundary edges. For an incomplete vertex v with its two adjacent edges e_1 and e_2, we have $d(v) = 2$. From (4) in Theorem 4, we obtain $R(e_1) < 2$ and $R(e_2) < 2$. Since there are no zero edges, it follows that $R(e_1) = R(e_2) = 1$. Thus the two edges are boundary edges.

(**ii**) We then prove that each of the two vertices of a boundary edge e is connected with one and only one other boundary edge.

- When a vertex v of the boundary edge e is an incomplete vertex, the statement is true by (i).

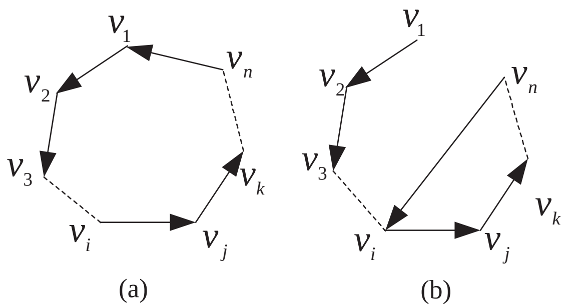

Fig. 5. Walking along boundary edges. (a) A boundary cycle. (b) An impossible path.

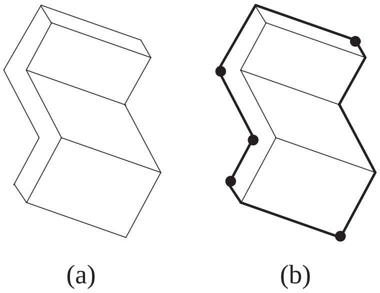

Fig. 6. (a) A line drawing without zero edges. (b) The boundary cycle (bold lines) on which all the boundary edges and incomplete vertices (filled-in circles) reside.

- When a vertex v of the boundary edge e is a complete vertex, let the other two edges connecting to v be e_1 and e_2, as shown in Fig. 4. From (5) in Theorem 4, we know that $R(e) + R(e_1) + R(e_2)$ is even. Since e is a boundary edge and there are no zero edges in the line drawing, we have $R(e) = 1, R(e_1) \neq 0, R(e_2) \neq 0, R(e_1) \leq 2$, and $R(e_2) \leq 2$. Therefore there are only two cases for $R(e_1)$ and $R(e_2)$: (a) $R(e_1) = 1$ and $R(e_2) = 2$, and (b) $R(e_1) = 2$ and $R(e_2) = 1$. Either case (a) or case (b) indicates that one and only one of the two edges e_1 and e_2 is a boundary edge.

(**iii**) By the statement in (ii), we can always walk from a boundary edge to another boundary edge (see Fig. 5). Since the line drawing has finite vertices, we must meet one of the vertices on the path we have passed through. However, the case in Fig. 5(b) cannot happen because the vertex v_i is met by three boundary edges, which contradicts the statement in (ii). Thus we can only have the case shown in Fig. 5(a), which forms a boundary cycle. ■

Corollary 1: If a line drawing has no zero edges, an incomplete vertex must be on a boundary cycle.

Proof: Since every incomplete vertex is connected with two boundary edges (see (i) in the proof of Theorem 5), it must be on a boundary cycle by Theorem 5. ■

From the next section, it can be seen that boundary cycles are important for the inference of the hidden structure of a line drawing. Theorem 5 and Corollary 1 show that boundary edges and incomplete vertices can only appear on boundary cycles in a line drawing without zero edges. Fig. 6 gives such an example.

When a line drawing has zero edges, we remove these edges and thus reduce the line drawing to a new one without zero edges. Then Theorem 5 and Corollary 1 apply to this new line drawing.

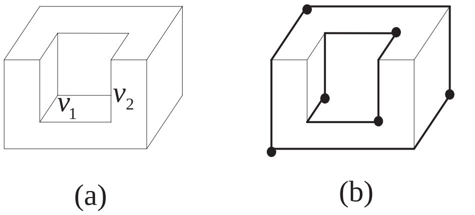

Fig. 7. (a) A line drawing with an zero edge (v_1, v_2). (b) The line drawing after removing the zero edge, where all the boundary edges and incomplete vertices are on the boundary cycles (bold lines).

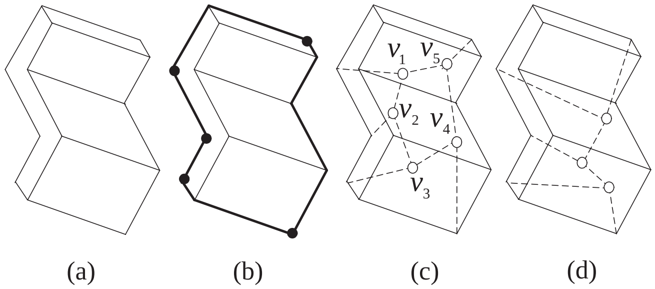

Fig. 8. An example of inferring the hidden topology of a line drawing. (a) The line drawing of an object. (b) Finding the boundary edges (the bold cycle) and the incomplete vertices (filled-in circles). (c) Initial hidden vertices (blank circles). (d) The most plausible result after the reduction of the initial hidden structure.

Fig. 7 shows an example. In Fig. 7(a), the edge (v_1, v_2) is a zero edge since v_2 is a broken vertex. After removing (v_1, v_2), the reduced line drawing is shown in Fig. 7(b), where the bold lines illustrate two boundary cycles. We can also see that all the incomplete vertices are on the cycles.

V. Recovering the Hidden Topological Structure

This section discusses how to recover the hidden structure of a line drawing, based on the assumptions and properties stated in the previous section. We first present an algorithm for this propose and then explain the implementation of the steps. We consider line drawings without zero edges first. How to handle line drawings with zero edges is discussed in Section V-E.

A. Outline of the Algorithm

Step 1. Compute the degrees of all the vertices and the
ranks of all the edges from a line drawing.
Step 2. Find boundary cycles and incomplete vertices.
Step 3. Construct an initial hidden structure.
Step 4. Reduce the initial hidden structure to the most
plausible one according to human visual
perception of the 3D object.

Fig. 8 demonstrates how the algorithm works by an example. In the algorithm, Step 1 has been described in the first part of Section IV; Step 2 can be done according to Theorem 5, Corollary 1, and the definitions of boundary cycles and incomplete vertices; Step 3 will be discussed in Section V-B; Step 4 will be explained in Sections V-C and V-D.

B. Constructing an Initial Hidden Structure

Theorem 2 indicates that the largest $N_H = |\mathcal{V}_\mathcal{I}|$ since we now consider a line drawing without zero edges (thus $\mathcal{V}_\mathcal{B} = \emptyset$).

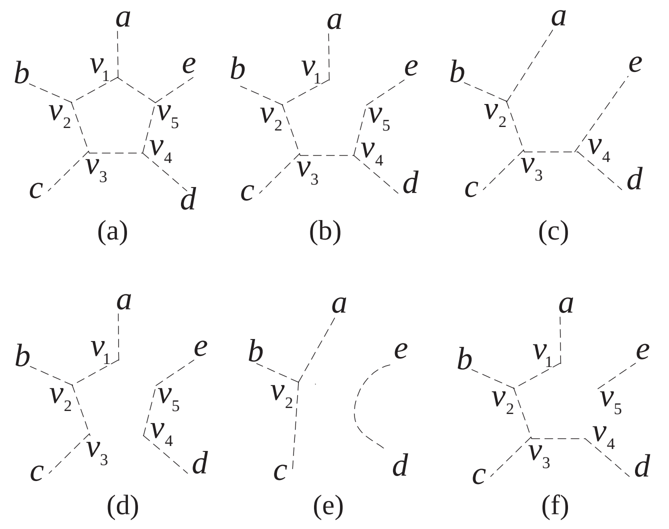

Fig. 9. Illustration of the cutting-and-merging procedure where a–e are incomplete vertices, v_{1-5} are hidden vertices, and $(v_1, v_2, v_3, v_4, v_5, v_1)$ is a hidden cycle. (a) The initial hidden structure. (b) Result after cutting one edge (v_1, v_5) from (a). (c) Result after merging v_1 to v_2 and v_5 to v_4 from (b). (d) Result after cutting two edges (v_1, v_5) and (v_3, v_4) from (a). (e) Result after merging v_1 and v_3 to v_2 and merging v_5 and v_4 from (d), where v_5 and v_4 disappear after being merged. (f) Result after cutting two edges (v_1, v_5) and (v_4, v_5) from (a), where v_5 has no place to be merged with.

We set $|\mathcal{V}_\mathcal{I}|$ hidden vertices and connect each incomplete vertex to a different hidden vertex. Two hidden vertices are connected if their corresponding incomplete vertices are closest on the boundary cycle. One example is given in Fig. 8(c), where the cycle $(v_1, v_2, v_3, v_4, v_5, v_1)$ is a hidden cycle. Note that the result of this initialization meets the constraints presented in Section IV.

C. Reducing the Initial Hidden Structure

Beginning with the initial hidden structure, we design a procedure to search for other possible hidden structures with fewer hidden vertices. The procedure uses a strategy of cutting-and-merging of edges and vertices. Cutting one edge on a hidden cycle removes this edge from the cycle while keeping the two vertices of the edge. After the cutting, the two hidden vertices of the edge are met by only two hidden edges (see Fig. 9(b)). To maintain that every vertex is met by three edges, we merge the two vertices to their adjacent hidden vertices (see Fig. 9(c)). Since each cutting-and-merging reduces two of the hidden vertices, the resulting number of hidden vertices is even (odd) if the initial N_H is even (odd), which satisfies the constraint imposed by Theorem 3.

At first, the cutting is applied to one hidden edge on the hidden cycle each time, resulting in different hidden structures. The cutting is also used to cut two or more such edges each time. It should be emphasized that not every cutting of two or more edges each time is valid. Figs. 9(d) and (f) show two examples of cutting two edges each time from Fig. 9(a). The former is valid but the latter is not.

The cutting-and-merging procedure is always applied to the initial hidden structure, cutting one edge or multiple edges each time. The maximum number of edges that can be cut each time is the largest integer $\leq N_H/2$ because removing one edge reduces two hidden vertices. All the hidden structures obtained from the procedure plus the initial structure are kept for the selection of the most plausible one. Fig. 10 shows all the hidden structure obtained from Fig. 8(c).

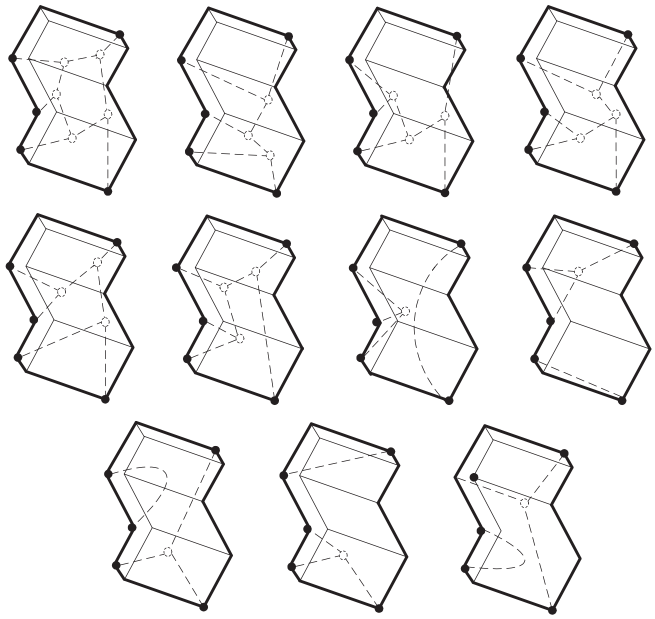

Fig. 10. The initial hidden structure and all others derived by the cutting-and-merging procedure.

Now we analyze the complexity of the cutting-and-merging procedure. Since the maximum number of edges that can be cut each time is the largest integer $l \leq N_H/2 = |\mathcal{V}_\mathcal{I}|/2$, the number of all the hidden structures (valid or invalid) is equal to $C^0_{|\mathcal{V}_\mathcal{I}|} + C^1_{|\mathcal{V}_\mathcal{I}|} + C^2_{|\mathcal{V}_\mathcal{I}|} + ... + C^l_{|\mathcal{V}_\mathcal{I}|}$, where $C^0_{|\mathcal{V}_\mathcal{I}|} = 1$ denotes the initial structure. Thus the complexity of the procedure is exponential in the number of the incomplete vertices in a line drawing. Although it is exponential, the computational time for recovering a hidden structure usually takes negligible time when the number of incomplete vertices is small (say, less than 15). For each of the objects in the experiments, the time spent to recover the hidden structure is less than 0.02 second.

D. Selecting the Most Plausible Structure

Given a set of possible hidden structures, the selection of the most plausible one is based on the visual psychological properties from Gestalt psychology. The law of symmetry is one of the most important Gestalt laws, which reveals that the human visual system is overwhelmed by symmetry and tends to interpret a figure in such a way as to produce an object that is as symmetrical as possible. When this law is applied to the inference of the hidden structure of a line drawing, each hidden face is related to a similar visible face. At this stage, since we discuss topological structures only, we say that two faces are similar if they have the same number of edges. These two similar faces are also called topologically same. Motivated by this law, we have the following Rule 1 for selecting the most plausible hidden structure:

Rule 1: Given a set of hidden structures from a line drawing, select the one having as many hidden faces similar to the visible faces as possible.

We define a non-symmetry measure NSM for the selection. Suppose that there are n hidden structures. Initially, we set $NSM_i = 0$, $1 \leq i \leq n$. For every hidden face in the ith structure, we check if there is a corresponding topologically same visible

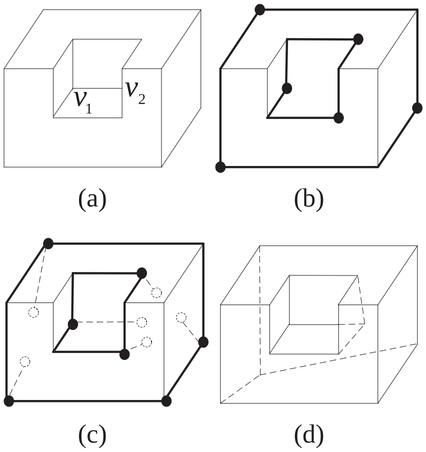

Fig. 11. (a) A line drawing with only one zero edge. (b) Two found boundary cycles (bold lines) with incomplete vertices (filled-in cycles) after removing the zero edge from (a). (c) Each incomplete vertex connected to a hidden vertex. (d) The final selected result.

face. If no, increase NSM_i by 1. For each structure, one visible face cannot be used more than once in the checking. The most plausible is the jth structure with $NSM_j = \min_{1 \leq i \leq n} \{NSM_i\}$.

Rule 2: When there are two or more plausible structures with the same value of non-symmetry measure, choose the one with the fewest hidden vertices.

Rule 2 is based on the property of simplicity in Gestalt psychology. If finally there are still more than one candidate after applying Rules 1 and 2, all of them are reconstructed and the user can select one. In most cases, only one most plausible hidden structure in a line drawing is obtained using Rules 1 and 2. For the example shown in Fig. 10, the second hidden structure is the most plausible one and selected.

E. Handling Line Drawings with Zero Edges

For a line drawing with zero edges, we remove these edges and thus reduce the line drawing to a new one without zero edges. Then Theorem 5 guarantees that the boundary edges of the new line drawing form boundary cycles. Based on these boundary cycles, we treat the zero edges as part of the initial hidden topology and then perform the inference procedure described in Sections V-C and V-D. The only difference is that we do not cut and merge the zero edges. An example is shown in Fig. 11. From the original line drawing (Fig. 11(a)), we remove the zero edge (v_1, v_2) and find the two boundary cycles (Fig. 11(b)). Then each incomplete vertex is connected to a hidden vertex, as shown in Fig. 11(c). Note that the zero edge is now treated as a “hidden” edge. The final selected hidden structure is given in Fig. 11(d).

Another line drawing with four zero edges is shown in Fig. 12(a). After removing the zero edges, we obtain a new line drawing (Fig. 12(b)), from which one boundary cycle and incomplete vertices are found. In the initialization, each incomplete vertex is connected to a hidden vertex (Fig. 12(c)), and the hidden vertices are connected by a hidden cycle (Fig. 12(d)). Then the inference procedure reduces the initial hidden structure, and the most plausible structure is shown in Fig. 12(e).

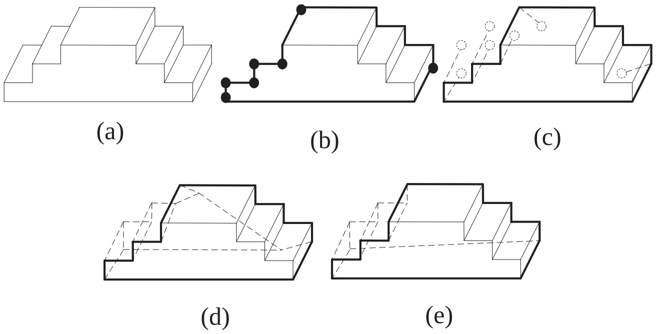

Fig. 12. (a) A line drawing with four zero edges. (b) One boundary cycle (bold lines) with incomplete vertices (filled-in circles) after removing all the zero edges from (a). (c) Each incomplete vertex connected to a hidden vertex. (d) The initial hidden structure (dotted lines). (e) The final selected result.

It should be emphasized that the found hidden vertices and edges, except zero edges, have no geometrical constraints in the 2D plane. They provide only the topological structures. The geometrical reconstruction of a complete 3D object from a line drawing is the purpose of the next section.

VI. 3D Reconstruction

In what follows, we call a line drawing with its recovered hidden structure a complete line drawing. After obtaining a complete line drawing, the next task is to reconstruct its 3D shape. Since we already know the face topology of the complete line drawing, we only need to derive the 3D coordinates of all the visible and hidden vertices. We consider a line drawing as a parallel (or near parallel) projection of a 3D object. The x- and y-coordinates of each visible vertex are thus already known, and only the z-coordinate (depth) has to be derived. However, all the x-, y-, and z-coordinates have to be found for hidden vertices. This reconstruction problem is more difficult than those in the previous work where the 2D positions of all the vertices and edges of a line drawing are known.

In the following, we use an optimization-based approach to tackle the reconstruction problem. It inflates a flat line drawing into a 3D object by assigning depths to all the vertices and x- and y-coordinates to all the hidden vertices. The recovered object should be in accordance with our visual perception. Note that in Section V-E, we treat zero edges as part of the hidden structure of a line drawing. In 3D reconstruction, the x- and y-coordinates of the visible vertices on zero edges are already available (see Fig. 12(e)). Besides, the x- and y-coordinates of a hidden vertex connected to a broken vertex are not independent; this hidden vertex is confined on the line passing through the visible broken edge corresponding to the broken vertex. An example is given in Fig. 13, where v_2 is a broken vertex and not a real vertex of the 3D object. We do not need to derive the depth of v_2. Instead, we need to find the x-, y-, and z-coordinates of the vertex v_3. The x- and y-coordinates of v_3 are confined on the line passing through the broken edge (v_1, v_2). More precisely, v_3 is located somewhere by stretching the broken edge (v_1, v_2). Next we develop an objective function first and then discuss how to obtain the 3D reconstruction by optimization.

The objective function consists of three components. The first one is a symmetry measure. Based on the spirit of the law of symmetry from Gestalt psychology, we consider a symmetry measure S for a closed planar figure. It is defined as

$$S = \frac{A}{P^2}, \tag{6}$$

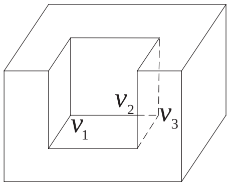

Fig. 13. The 2D position of v_3 obtained by stretching the visible broken edge (v_1, v_2).

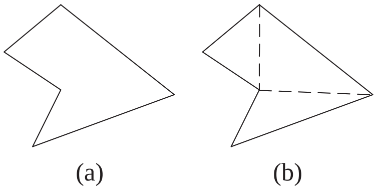

Fig. 14. (a) A face of an object which may not be planar in 3D space. (b) Triangulation of the face, where the area of the face is denoted by the sum of the areas of the three triangles.

where A and P are the area and perimeter of the figure, respectively.

It holds that $S \leq \frac{1}{4\pi}$ for any closed planar figure [45]. A circle is the most symmetrical planar figure with $S = \frac{1}{4\pi}$. For a polygon with m vertices, its symmetry measure $S \leq 4m\tan(\frac{\pi}{m})$ [45]. The maximum is achieved if and only if the polygon is the most symmetrical with m equal-length sides. These facts indicate that (6) is a rather reasonable measure of symmetry.

A polyhedron consists of more than three faces, each being a polygon. We consider the recovered object as the integration of all its planar faces in 3D space. Thus the whole symmetry measure of a polyhedron with n faces is defined as

$$WS = \sum_{i=1}^{n} \frac{A_i}{P_i^2} \tag{7}$$

where A_i and P_i, $1 \leq i \leq n$, are the area and perimeter of face i, respectively. We expect that given a line drawing, maximizing WS combined with other two criteria would provide us with the most plausible recovered 3D object. The intuition behind it is that if we force the faces of the reconstructed object to be as symmetrical as possible, then the flat 2D line drawing will be inflated into a 3D object. More explanation of this constraint and its comparison with the second constraint described next can be found in [46].

It should be mentioned that the faces of the 3D object may not be strictly planar. In this case, the area of a face is denoted by the sum of the areas of the triangles obtained by the triangulation of the face (see Fig. 14). An algorithm for the triangulation of a polygon can be found in [47].

Marill [34] presented his approach to 3D reconstruction based on a criterion: minimizing the standard deviation of all the angles (SDA) formed by every two adjoining lines in the reconstructed object. SDA is computed by

$$SDA = \mathrm{Var}(\theta_1, \theta_2, ...\theta_k) \tag{8}$$

where $\theta_1, \theta_2, ..., \theta_k$ are all these k angles and Var denotes standard deviation. This criterion can be regarded as another representation of the symmetry constraint. Minimizing SDA is the second component of the objective function.

The third component is about planarity. When we observe a line drawing representing a 3D polyhedron, we can clearly identify the cycles representing faces. This face information is very useful in helping our perception of the shape of the object. We also enforce this planarity constraint in the 3D reconstruction.

Let the plane passing through face i be represented by a vector $\mathbf{f}_i = (a_i, b_i, c_i)^T$. In what follows, we also represent 3D vertices in vector form. Then a vertex $\mathbf{v} = (x, y, z)^T$ on the plane $\mathbf{f}_i$ satisfies the linear equation

$$a_i x + b_i y + c_i z - 1 = 0, \tag{9}$$

or

$$\mathbf{v}^T \mathbf{f}_i = 1. \tag{10}$$

Suppose that face i has m vertices $\mathbf{v}_{ij} = (x_{ij}, y_{ij}, z_{ij})^T$, $1 \leq j \leq m$. Then we have

$$\mathbf{V}_i \mathbf{f}_i = \mathbf{1}, \tag{11}$$

where $\mathbf{V}_i = (\mathbf{v}_{i1}, \mathbf{v}_{i2}, .., \mathbf{v}_{im})^T$ and $\mathbf{1}$ is an m-dimensional column vector with all its elements being 1.

However, for a face with more than three vertices, it is not likely for all the vertices to be located exactly on a plane in 3D space, especially during the first stage of the reconstruction. To compute the deviation from planarity for the vertices, the least square fitting technique is employed. Instead of solving the over-constrained equation in (11), we seek $\mathbf{f}_i$ that minimizes the square error $(\mathbf{V}_i\mathbf{f}_i - \mathbf{1})^T(\mathbf{V}_i\mathbf{f}_i - \mathbf{1})$. In other words, $\mathbf{f}_i$ should satisfy

$$\frac{\partial}{\partial \mathbf{f}_i}(\mathbf{V}_i\mathbf{f}_i - \mathbf{1})^T(\mathbf{V}_i\mathbf{f}_i - \mathbf{1}) = 0, \tag{12}$$

from which we have

$$\mathbf{f}_i = (\mathbf{V}_i^T\mathbf{V}_i)^{-1}\mathbf{V}^T\mathbf{1}. \tag{13}$$

Thus, we evaluate the deviation from planarity DP_i for the vertices on face i by the sum of all the squared distances of the m vertices to the plane, i.e.,

$$DP_i = \frac{1}{a_i^2 + b_i^2 + c_i^2} \sum_{j=1}^{m} (a_i x_{ij} + b_i y_{ij} + c_i z_{ij} - 1)^2. \tag{14}$$

For an object with n faces, the weighted total deviation from planarity DP is defined as

$$DP = w \sum_{i=1}^{n} DP_i, \tag{15}$$

where the weight w is a balance parameter and is chosen to be 0.01. Because DP is usually much larger than WS' in (16) and SDA if $w = 1$, we set w a small value to balance the three constraints.

Based on the above analysis, the objective of reconstruction is to maximize WS while minimizing DP and SDA. In order to combine the three targets into one objective function, WS is replaced with WS' that takes the form

$$WS' = \sum_{i=1}^{n} \frac{P_i^2}{A_i}. \tag{16}$$

Finally, the objective function to be minimized is defined as

$$\begin{aligned} f(z_1, z_2, ..., z_v, &d_1, d_2, ..., d_b, x_{h1}, y_{h1}, x_{h2}, y_{h2}, ..., x_{hu}, y_{hu}) \\ &= \lambda(WS' + SDA) + (1 - \lambda)DP, \end{aligned} \tag{17}$$

where $0 \leq \lambda \leq 1$ is a weighting factors; $z_1, z_2, ..., z_v$ are the depths of all the v visible and hidden vertices; $d_1, d_2, ..., d_b$ are the lengths of all the b broken vertices stretched out along the directions of their broken edges (see Fig. 13); $(x_{h1}, y_{h1}), (x_{h2}, y_{h2}), ..., (x_{hu}, y_{hu})$ are the x- and y-coordinates of all the u hidden vertices, except the b hidden vertices stretched out from the broken vertices. Minimizing f expresses our aim to construct a 3D object as symmetrical as possible with the constraint of planarity.

With a fixed λ in (17), many optimization algorithms can be used to minimize the objective function, such as hill-climbing, genetic algorithms, simplex search, and simulated annealing. However, we have found that Leclerc and Fischler's continuation method [29] with varying λ is more effective to obtain good results. Leclerc and Fischler used this method to minimize an objective function that is in a similar form to (17). We also use this method for our 3D reconstruction.

In the continuation method, λ is a sequence of descent steps applied to f for decreasing values of λ. The sequence begins with some initialization of the variables of f (e.g., assigning random values to them) and with some relatively large $\lambda \leq 1$, and the hill-climbing algorithm presented in [34] is employed to minimize f. Then λ is reduced by a given amount and the hill-climbing algorithm is applied again, starting from the solution obtained by the hill-climbing algorithm for the previous value of λ. This procedure is repeated until λ reaches a predefined small value. This strategy favors the symmetry at the beginning of the optimization, and then the constraint of planarity becomes more dominating. More details about this method can be found in [29]. In our experiments, the sequence of λ is: $1, 1/2, 1/4, 1/8, 1/16$. Many other descent schemes for λ are possible which can lead to similar results.

VII. Experimental Results

In this section, we present a number of examples to illustrate the proposed approach to the inference and reconstruction of complete 3D objects from line drawings without hidden edges and vertices. Fig. 15 shows 24 line drawings and their reconstruction results. The results of the inference of the hidden structures (dotted lines) are also illustrated. Note again that the hidden structures in these illustrations are topological but not geometrical; the 2D positions of the hidden vertices cannot be located in the inference process. They are recovered together with the z-coordinates of all the vertices in the reconstruction stage. From Fig. 15, we can see that the reconstructed objects accord with our visual perception very well.

The algorithm is implemented in C++ running on a Pentium 4 PC with a 3.2 GHz CPU. Table I gives the computational time for each object in Fig. 15, together with the numbers of the visible and invisible vertices and edges of each line drawing. The time includes both the inference of the hidden structure and the 3D reconstruction. The time for the inference can be negligible because the longest time is less than 0.02 second. From the table, we see that each of the objects can be handled within 5 seconds.

We have obtained a large set of successful examples, in addition to those in Fig. 15. In general, however, our current approach can only deal with line drawings satisfying the three assumptions made in Section III. Fig. 16 shows an example our algorithm cannot handle. The hidden structure is illustrated by dotted lines. The 3D object has a through hole. The complex hidden structure makes our algorithm fail. However, our current work with the encouraging successful results is an important step towards the research on handling more complex objects. We believe that this can be achieved by exploring more geometrical constraints and using higher-level information from human visual perception.

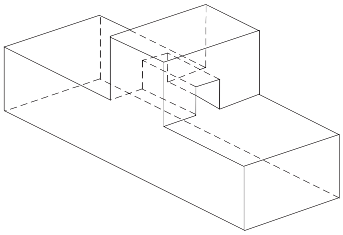

Fig. 16. A line drawing with a through hole which our algorithm fails to handle.

VIII. Conclusions

We have proposed a novel approach to 3D reconstruction from single 2D line drawings without hidden lines. We first infer the hidden structure of a line drawing with the help of the constraints given in several theorems. Then we present an optimization-based method to recover the 3D shape of the complete object. The objective function is developed based on perceptual symmetry and planarity. A number of encouraging results have been obtained, which demonstrate the success of the proposed approach. The future work includes handling curved objects and more complex polyhedra by exploring more geometrical constraints from line drawings and using higher-level information from human visual perception.

Acknowledgements

This work was supported by the Research Grants Council of the Hong Kong SAR (Project No. CUHK 414306). The preliminary version of this paper was published in ICCV 2005 [48].

References

[1] S. Ortiz Jr., "3D searching starts to take shape," *Computer*, Aug., pp. 24–26, 2004.

[2] P. Min, J. Chen, and T. Funkhouser, "A 2D sketch interface for a 3D model search engine," *ACM SIGGRAPH'02, Technical Sketches*, p. 138, 2002.

[3] M. Yu, I. Atmosukarto, W.K. Leow, Z. Huang, and R. Xu, "3D model retrieval with morphing-based geometric and topological feature maps," *IEEE Proc. Computer Vision and Pattern Recognition*, vol. 2, pp. 656–661, 2003.

[4] P. Debevec, C.J. Yaylor, and J. Malik, "Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach," *ACM Proc. SIGGRAPH'96*, pp. 11–20, 1996.

[5] A. Turner, D. Chapman, and A. Penn, "Sketching space," *Computers & Graphics*, vol. 24, pp. 869–879, 2000.

[6] P. Sturm and S. Maybank, "A method for interactive 3d reconstruction of piecewise planar objects from single images," *British Machine Vision Conference*, pp. 265–274, 1999.

[7] S. Bagali and J. Waggenspack, "A shortest path approach to wireframe to solid model conversion," *Proc. 3rd Symp. Solid Modeling and Applications*, pp. 339–349, 1995.

[8] T. Syeda-Mahmood, "Indexing of technical line drawing databases," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 737–751, 1999.

[9] M.Q.J. Marefat, "Hierarchical bayesian methods for recognition and extraction of 3-D shape features from CAD solid models," *IEEE Trans. Systems, Man and Cybernetics, Part A*, vol. 27, no. 6, pp. 705–727, 1997.

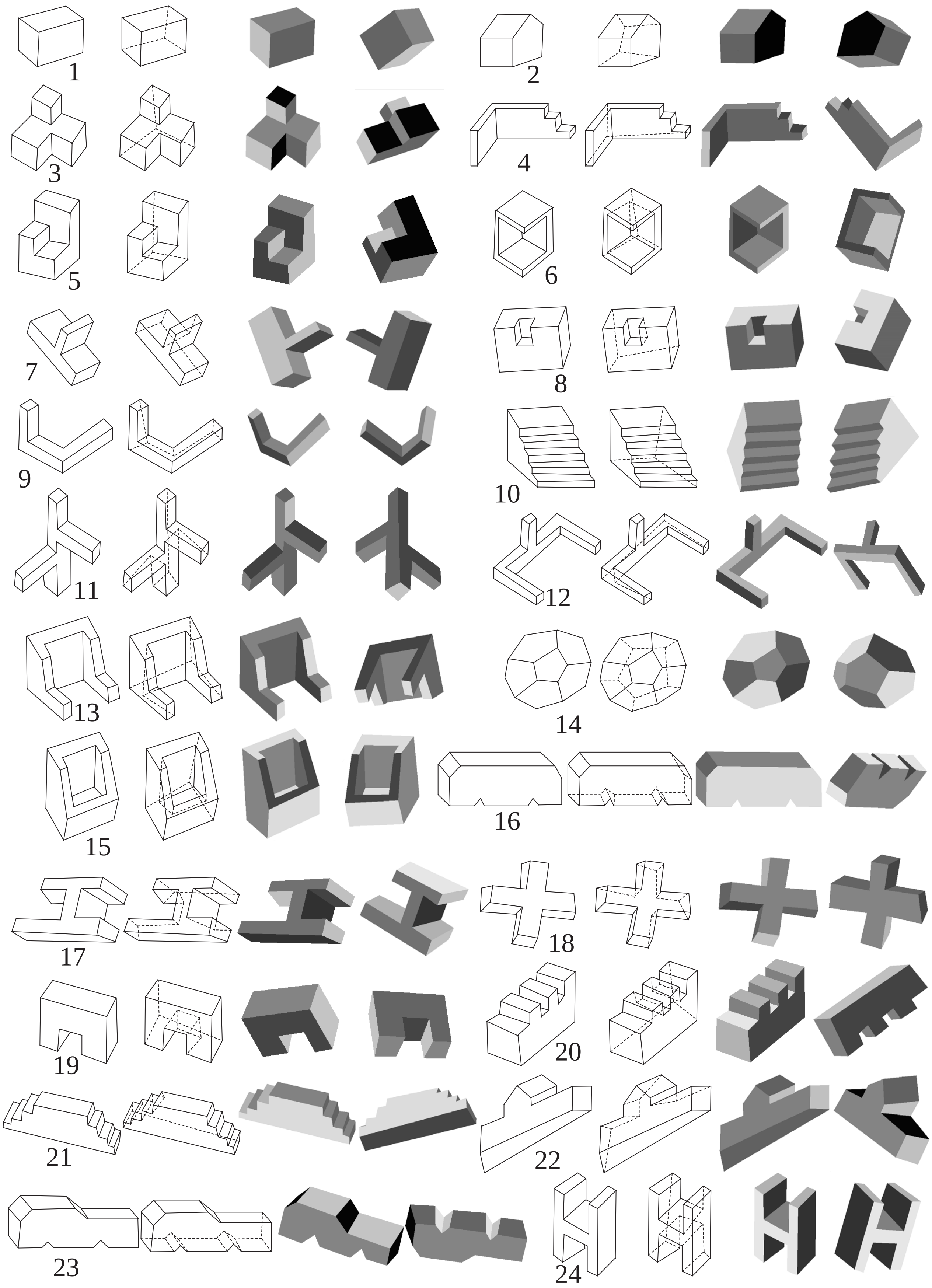

Fig. 15. Results of the inference of the hidden structures (dotted lines) and the final 3D reconstruction. Each object is shown in two views and its faces are denoted by different grey levels.

TABLE I
THE NUMBERS OF THE VISIBLE AND INVISIBLE VERTICES AND EDGES, AND THE TIME TAKEN BY OUR ALGORITHM TO HANDLE EACH LINE DRAWING IN FIG. 15.

Line drawing	Visible vertices/edges	Invisible vertices/edges	Time (seconds)	Line drawing	Visible vertices/edges	Invisible vertices/edges	Time (seconds)
1	7/9	1/3	0.05	13	23/31	2/6	2.45
2	8/10	2/5	0.02	14	15/20	5/10	1.89
3	17/24	1/3	0.56	15	18/22	4/10	2.55
4	19/27	1/3	0.94	16	16/19	8/17	4.03
5	15/21	1/3	0.24	17	21/25	5/13	3.60
6	15/19	2/6	0.61	18	21/25	5/13	1.72
7	15/19	2/6	0.50	19	12/14	5/11	0.76
8	15/19	2/6	0.42	20	27/33	5/13	4.68
9	13/17	3/7	3.02	21	31/39	4/12	4.87
10	23/33	1/3	3.47	22	15/18	4/10	1.42
11	23/29	3/9	2.39	23	19/24	7/15	4.98
12	23/29	2/6	2.81	24	21/25	5/13	4.19

[10] H. Lipson and M. Shpitalni, “Optimization-based reconstruction of a 3D object from a single freehand line drawing,” *Computer-Aided Design*, vol. 28, no. 8, pp. 651–663, 1996.

[11] D. Dori and L. Wenyin, “Automated CAD conversion with the machine drawing understanding system: concepts, algorithms, and performance,” *IEEE Trans. Systems, Man and Cybernetics, Part A*, vol. 29, no. 4, pp. 411–416, 1999.

[12] A. Shesh and B. Chen, “Smartpaper: An interactive and user friendly sketching system,” *Computer Graphics Forum*, vol. 23, no. 3, pp. 301–310, 2004.

[13] M. Clowes, “On seeing things,” *Artificial Intelligence*, vol. 2, pp. 79–116, 1971.

[14] D. Huffman, “Impossible objects as nonsense sentences,” *Machine Intelligence*, vol. 6, pp. 295–323, 1971.

[15] R. Haralick and L. Shapira, “The consistent labeling problem: Part 1,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 173–184, 1979.

[16] J. Malik, “Interpreting line drawings of curved objects,” *Int’l Journal of Computer Vision*, vol. 1, pp. 73–103, 1987.

[17] M.C. Cooper, “Interpretation of line drawings of complex objects,” *Image and Vision Computing*, vol. 11, no. 2, pp. 82–90, 1993.

[18] M.C. Cooper, “The interpretation of line drawings with contrast failure and shadows,” *Int’l Journal of Computer Vision*, vol. 43, no. 2, pp. 75–97, 2001.

[19] L.G. Shapiro and G.C. Stockman , *Computer Vision*. Upper Saddle River, N.J.: Prentice Hall, 2001.

[20] R. Lequette, “Automatic construction of Curvilinear Solid from Wireframe Views,” *Computer-Aided Design*, vol. 20, no. 4, pp. 171–180, 1988.

[21] S. Ablameyko, V. Bereishik, A. Gorelik, and S. Medvedev, “3D Object Reconstruction from Engineering Drawing Projections,” *Computing & Control Engineering Journal*, vol. 10, no. 6, pp. 277–284, 1999.

[22] M.H. Kuo, “Reconstruction of Quadric Surface Solid from Three-View Engineering Drawings,” *Computer-Aided Design*, vol. 30, no. 7, pp. 517–527, 1998.

[23] K. Sugihara, “Mathematical structures of line drawings of polyhedrons: toward man-machine communication by means of line drawings,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no.5, pp. 458–469, 1982.

[24] K. Sugihara, “A necessary and sufficient condition for a picture to represent a polyhedral scene,” *IEEE Trans.Pattern Analysis and Machine Intelligence*, vol. 6, no. 5, pp. 578–586, 1984.

[25] W. Whiteley, “A matroid on hypergraphs with applications in scene analysis and geometry,” *Discrete and Computational Geometry*, vol. 4, pp. 75–95, 1989.

[26] W. Whiteley, “Weavings, sections and projections of spherical polyhedra,” *Discrete and Computational Geometry*, vol. 32, pp. 275–294, 1991.

[27] L. Ros and F. Thomas, “Geometric methods for shape recovery from line drawings of polyhedra,” *Journal of Mathematical Imaging and Vision*, vol. 22, no. 1, pp. 5–18, 2005.

[28] L. Ros and F. Thomas, “Overcoming superstrictness in line drawing interpretation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 456–466, 2002.

[29] Y. Leclerc and M. Fischler, “An optimization-based approach to the interpretation of single line drawings as 3D wire frames,” *Int’l Journal of Computer Vision*, vol. 9, no. 2, pp. 113–136, 1992.

[30] M. Shpitalni and H. Lipson, “Identification of Faces in a 2D Line Drawing Projection of a Wireframe Object,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1000–1012, 1996.

[31] J. Liu and Y. Lee, “A graph-based method for face identification from a single 2D line drawing,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1106–1119, 2001.

[32] J. Liu, Y. Lee, and W.-K. Cham, “Identifying faces in a 2D line drawing representing a manifold object,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1579–1593, 2002.

[33] J. Liu and X. Tang, “Evolutionary search for faces from line drawings,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 861–872, 2005.

[34] T. Marill, “Emulating the human interpretation of line-drawings as three-dimensional objects,” *Int’l Journal of Computer Vision*, vol. 6, no. 2, pp. 147–161, 1991.

[35] L. Baird and P. Wang, “3D object perception using gradient descent,” *Int’l Journal of Mathematical Imaging and Vision*, vol. 5, pp. 111–117, 1995.

[36] P.A.C. Varley and R.R. Martin, “Estimating depth from line drawings” *Proc. 7th ACM Symposium on Solid Modeling and Applications*, pp. 180–191, 2002.

[37] K. Shoji, K. Kato, and F. Toyama, “3-D interpretation of single line drawings based on entropy minimization principle,” *IEEE Proc. Computer Vision and Pattern Recognition*, vol. 2, pp. 90–95, 2001.

[38] K. Sugihara, “An algebraic approach to shape-from-image problem,” *Artificial Intelligence*, vol. 23, pp. 59–95, 1984.

[39] I. Shimshoni and J. Ponce, “Recovering the shape of polyhedra using line-drawing analysis and complex reflectance models,” *Computer Vision and Image Processing*, vol. 65, no. 2, pp. 296–310, 1997.

[40] H. Shimodaira, “A shape-from-shading method of polyhedral objects using prior information,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 612–624, 2006.

[41] P.A.C. Varley and R.R. Martin, “A system for constructing boundary representation solid models from a two-dimensional sketch — topology of hidden parts,” *Proc. First UK-Korea Workshop on Geometric Modeling and Computer Graphics*, pp. 129–144, 2000.

[42] S. Palmer, *Vision Science: Photons to Phenomenology*. The MIT Press, 1999.

[43] K. Koffka, *Principles of Gestalt Psychology*. Routledge & K. Paul, 1963.

[44] D. West, *Introduction to Graph Theory*. Prentice Hall, 1996.

[45] M. Berger, *Geometrie*. Paris: CEDIC/Fernand Nathan, 1977.

[46] L. Cao, *3D Object Reconstruction from Line Drawings*. Master Thesis, the Department of Information Engineering, the Chinese University of Hong Kong, 2005.

[47] M.D. Berg, M.V. Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Springer, 2000.

[48] L. Cao, J. Liu, and X. Tang, “3D object reconstruction from a single 2D line drawing without hidden lines,” *IEEE Proc. Int’l Conf. Computer Vision*, vol.1, pp. 272–277, 2005.

PLACE
PHOTO
HERE

Liangliang Cao received the B.E. degree from the University of Science and Technology of China, Hefei, in 2003, and the M.Phil. degree from the Chinese University of Hong Kong, Hong Kong, in 2005. After spending one year as a research assistant in the Department of Information Engineering, the Chinese University of Hong Kong, he is now a Ph.D. student in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. His research interests include computer vision and machine learning.

PLACE
PHOTO
HERE

Jianzhuang Liu (M’02-SM’02) received the B.E. degree from Nanjing University of Posts and Telecommunications, P. R. China, in 1983, the M.E. degree from Beijing University of Posts and Telecommunications, P. R. China, in 1987, and the Ph.D. degree from the Chinese University of Hong Kong, in 1997. From 1987 to 1994, he was a faculty member in the Department of Electronic Engineering, Xidian University, P. R. China. From August 1998 to August 2000, he was a research fellow at the School of Mechanical and Production Engineering, Nanyang Technological University, Singapore. Then he was a postdoctoral fellow in the Chinese University of Hong Kong for several years. He is now an assistant professor in the Department of Information Engineering, the Chinese University of Hong Kong. His research interests include image processing, computer vision, pattern recognition, and graphics.

PLACE
PHOTO
HERE

Xiaoou Tang (S’93-M’96-SM’02) received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, and the M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a professor and the director of Multimedia Lab in the Department of Information Engineering, the Chinese University of Hong Kong. He is also the group manager of the Visual Computing Group at the Microsoft Research Asia. He is a local chair of the IEEE International Conference on Computer Vision (ICCV) 2005, an area chair of CVPR 2007, a program chair of ICCV 2009, and a general chair of the ICCV International Workshop on Analysis and Modeling of Faces and Gestures 2005. He is a guest editor of the Special Issue on Underwater Image and Video Processing for IEEE Journal of Oceanic Engineering and the Special Issue on Image- and Video-based Biometrics for IEEE Transactions on Circuits and Systems for Video Technology. He is an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). His research interests include computer vision, pattern recognition, and video processing.