

# Optimization-Based Reconstruction of a 3D Object From a Single Freehand Line Drawing

H. Lipson and M. Shpitalni

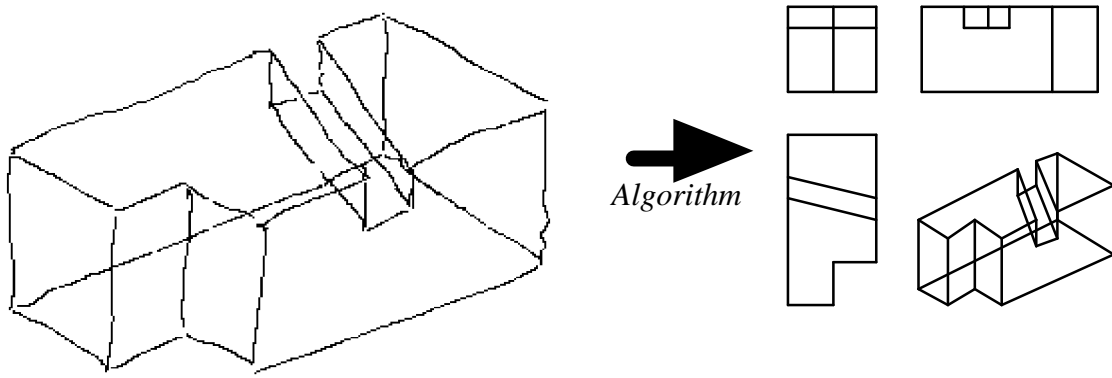
*Laboratory for Computer Graphics and CAD, Faculty of Mechanical Engineering,  
Technion, Haifa 32000, Israel*

**Abstract** - This paper describes an optimization-based algorithm for reconstructing a 3D model from a single, inaccurate, 2D edge-vertex graph. The graph, which serves as input for the reconstruction process, is obtained from an inaccurate freehand sketch of a 3D wireframe object. Compared with traditional reconstruction methods based on line labeling, the proposed approach is more tolerant of faults in handling both inaccurate vertex positioning and sketches with missing entities. Furthermore, the proposed reconstruction method supports a wide scope of general (manifold and non-manifold) objects containing flat and cylindrical faces. Sketches of wireframe models usually include enough information to reconstruct the complete body. The optimization algorithm is discussed, and examples from a working implementation are given.

## Introduction

Despite significant developments in the field of computer aided design, conceptual designers still tend to prefer pencil and paper to CAD systems. This tendency is especially noticeable in the preliminary stages of product formation, when a design is nothing more than a collection of abstract ideas. The reason most often given for not using a computer at this stage of conceptual design is that the interface is not suitable for sketching very basic ideas. Although the interface is user-friendly, it still seems to disturb the designer's flow of ideas and to interfere with creativity and concentration.

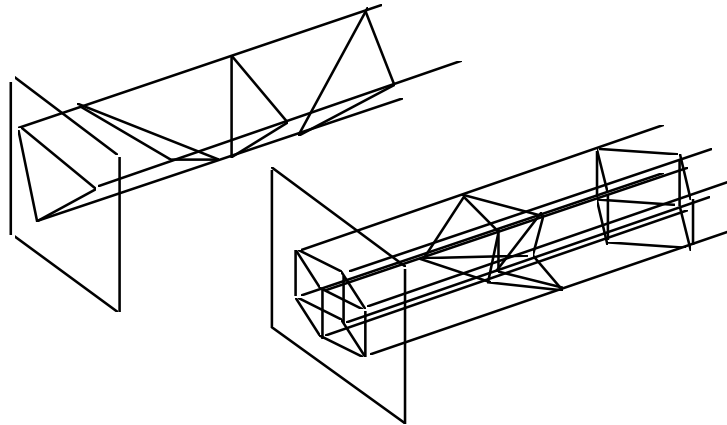
A new interface for conceptual design based on 3D object reconstruction from a single 2D sketch was introduced by the authors [1]. The system concept is illustrated in Fig. 1. The system analyzes and interprets single 2D inaccurate line drawing (projection) input and reconstructs the 3D object most likely to have been selected by a human observer. Once the 3D model is obtained, it can be manipulated or modified, and further detail can be sketched in, perhaps from a more convenient point of view. The output of the system is an abstract three-dimensional model that was constructed without any commands or keyboard input. Although this model does not have exact dimensions, it is sufficient to convey the design concept to other designers or to a conventional CAD system. The proposed approach is intended to provide a designer with the means to convey his ideas to a CAD system in a way similar to the way in which designers communicate among themselves.



**Fig 1.** Reconstruction of a 3D object from a single 2D line drawing

Much work has been devoted to reconstructing 3D information from a 2D source. The research presented in this paper investigates the ability to reconstruct the 3D model of an object depicted by a 2D freehand sketch.

The line drawing is assumed to represent a projection of a general object, in wireframe representation, as seen from some arbitrary point of view. Since the 2D line drawing lacks depth information, the reconstruction process is non-deterministic. A simple line drawing can represent the projections of an infinite number of possible 3D objects (Fig. 2). Interpretation heuristics regarding regularities in the image, however, allow additional implicit information to be extracted from the line drawing and make it possible to reconstruct the object most likely represented by the input drawing.



**Fig 2.** A single 2D sketch can be a projection of an infinite number of 3D objects.

The reconstruction algorithm proposed and discussed in this paper is executed after several preprocessing stages in which the initial 2D rough sketch is transformed into a 2D *line* and *junction* graph. In this graph, each line is assumed to correspond to the projection of exactly one *edge* of the depicted 3D object and each junction of lines to one *vertex* of the object. The graph is 2D and contains no information regarding the 3D object represented, its type or its position with respect to the viewpoint.

The preprocessing stage is described in detail in [1] and [2]. In brief, the initial (raw) sketch strokes are smoothed, classified into geometrical entities and then linked together at their endpoints to form a projected topological edge-vertex graph. The original sketch

is obtained from on-line sketching interface, and it is assumed that each edge is drawn as a continuous sketch stroke. This assumption facilitates the automatic distinction between real vertices and accidental crossing of lines. Then, faces of the depicted 3D wireframe object are determined in the 2D topological graph. Because general objects are considered (both manifold and non-manifold and possibly including cylindrical faces), the face identification process is complex. A separate paper by the authors [2] dedicated to this subject has been submitted.

In this paper, the motivation for this work is discussed in the introduction. Next, related work is reviewed. Then, the requirements are set, and the reconstruction problem and underlying assumptions are discussed. The reconstruction method is then developed, and examples from a working implementation are given.

## **Problem definition and assumptions**

Input to the reconstruction algorithm is a single 2D projection of some 3D object in the form of a **2D line-junction** graph. The graph originated from a previous rough freehand sketch and may still contain inaccuracies both in position of vertices and in connectivity of edges. The goal of the algorithm is to restore the original **rough** 3D object using information derived from the projection alone. It should be noted again that at this stage the graph is still **two-dimensional** and represents a projection of a 3D object. Furthermore, no additional information is available regarding the three dimensional object itself, its type or its position with respect to the viewpoint.

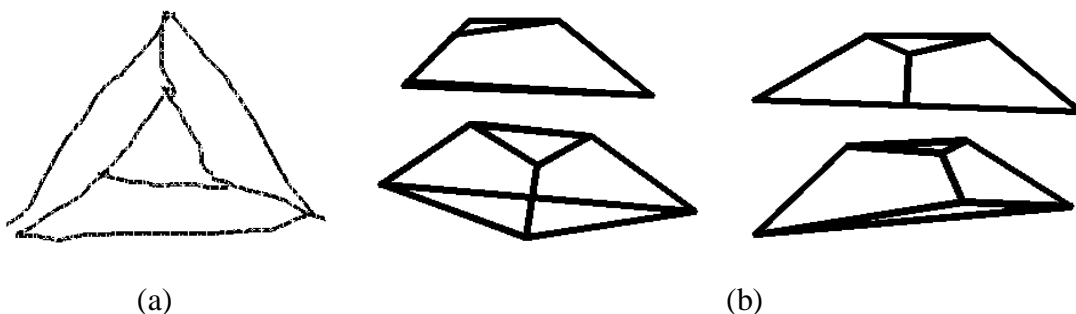
The problem can be defined more precisely by setting out a number of assumptions and requirements.

### **Assumptions:**

- \* The input to the system consists of a single 2D line drawing only which is given as a graph of connected entities. (This graph may be the result of previous processing of a raw input sketch.)
- \* The input projection represents a wireframe model of a general object that may be manifold, non-manifold or an assembly of such objects. However, no information is provided to the system about the 3D object itself, its type or its position relative to the viewpoint.
- \* The projection is drawn from a general - non-accidental - viewpoint that reveals all edges and vertices. That is, none of the edges or vertices of the object coincide accidentally, and none of them accidentally appear to be joined in the projection. This assumption also requires that the topology of the projected edge-vertex graph will not change if the viewpoint is perturbed slightly within its neighborhood. (Although this requirement seems limiting, it complies with regular practice in engineering communication.)
- \* All drawn lines and curves in the projection represent real edges, silhouette curves or intersections of faces in the 3D object. No shadow lines or surface marks are allowed.
- \* The sketch is assumed to depict the object in a parallel (or nearly parallel) projection. Although a perspective projection could convey more spatial information, this source relies too heavily on the user's sketching skills and is therefore prone to inaccuracies.

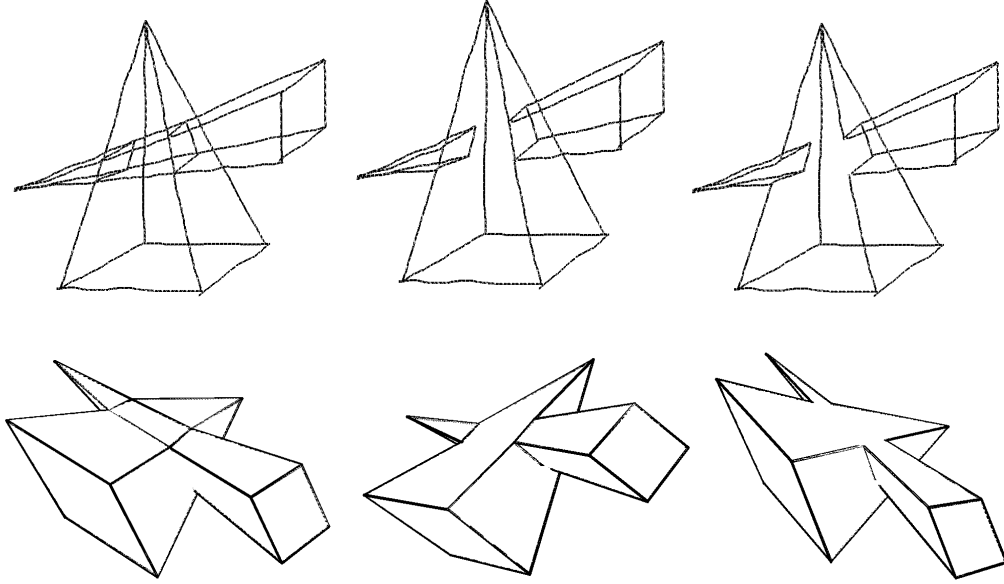
## Requirements:

- \* The algorithm must arrive at the most plausible reconstruction. A 2D sketch might be a projection of an infinite number of possible 3D objects, as is illustrated in Fig. 2. It is therefore impossible to uniquely identify the source object. Human observers, however, seem to have quite definite ideas regarding the 3D object that gave rise to a particular projection. The reconstruction, then, must arrive at the *most plausible* 3D object described by a given projection, that is, the object that human observers are most likely to select.
- \* The algorithm may not consult the user at any point during the reconstruction (non-interactive execution).
- \* The algorithm is required to arrive at a *rough* 3D reconstruction. For example, if a sketch of a cube is given, the algorithm is required to produce a rough 3D cube where: (1) dimensions bear a rough resemblance to those of the sketch; (2) lines are approximately perpendicular; and (3) faces are approximately planar (see examples in Fig. 13). Although this requirement is somewhat "loose", a user can easily determine if the computer has come up with the "correct" solution, even if it is rough. To be more specific, the resulting reconstruction must exhibit correct three-dimensional topology (i.e., relative position of faces, lines and vertices, such as above, below, between, adjacent, intersecting, etc.) and approximate geometrical qualities (parallelism, perpendicularity, proportional lengths, symmetry, etc.). Currently, there is no accurate or absolute measure of the correctness of a reconstruction. A classical example is the question of *generic reconstructibility* addressed by Sugihara [8]. The sketch shown in Fig. 3 (a) below is *not* mathematically reconstructable with planar faces, since the extended three lines do not meet at the same point. However, observers of the sketch immediately understand it to represent a truncated pyramid with planar faces. **It is exactly this capability that we wish to emulate.** Indeed, the reconstruction shown in Fig. 3 (b) does reflect the correct notion of the object, though not strictly accurately. For this reason, the reconstruction technique does not rely, at any point, on **exact** solution of equations, as in previous approaches. In fact, this tolerance is what makes the reconstruction possible. Some tests showed that forcing **strict planarity** on a model originating in an **inaccurate** sketch produces reconstructions which are not plausible. For conveying qualitative conceptual information, this kind of reconstruction will suffice. Clearly, for the reconstructed object to be of further use in detail design, it must be transformed into an accurate model by means of further information, such as explicit dimensions and a variational geometry process. This stage is currently under research.



**Fig 3.** (a) Original sketch of an object unable to be reconstructed generically;  
(b) views of the reconstructed 3D object (2 secs.).

- \* The procedure for identifying the most probable solution must be able to tolerate faults due to (a) inaccurate positioning of vertices, (b) inaccurate connectivity between entities, and (c) missing or excess entities. In fact, this flexibility is a key principle in determining whether the proposed reconstruction algorithm can be used in practice. Freehand sketches are often rough and inaccurate; it is actually the inaccurate nature of the sketch that makes it a fast, comfortable and natural communication language for conveying conceptual ideas. In order to preserve this important quality, fault tolerance is required. Similarly, engineers may express a geometrical concept by occasionally eliminating or adding some lines beyond the correct technical drawing regulations. Although these deviations may render the drawing topologically incorrect, they often do not impede an observer's ability to understand them, as is illustrated in the two incorrect drawings in Fig. 4. We wish to emulate this tolerance.



**Fig 4.** Different (incorrect) versions of sketches of an intersection of two pyramids, and (correct) 3D reconstructions.

## Related work

Much work has been done on reconstruction of a 3D object from three (and sometimes two) orthogonal projections; some is surveyed in [3]. Most of these methods rely on correlating between different views while relying on their mutual orthogonality for deriving spatial information. Since a single projection is given, these works are not applicable here. A more elaborate survey of 3D reconstruction methods from both single views and multiple views is provided in [4].

Single view scenes with hidden lines *removed* have been studied qualitatively by Huffman [5] and Clowes [6] and quantitatively by others, e.g., Kanade [7] and Sugihara [8], resulting in various line labeling schemes based on junction libraries. According to these techniques, all line segments in a drawing are labeled either as a concave or convex junction of faces or as an edge of an occluding face. A set of consistent line labels is searched for in a library of possible junction configurations. Several such sets may be generated. In general, however, line labeling procedures serve for drawing *interpretation* rather than actual *reconstruction* of the depicted 3D scene. Kanade [7] and Sugihara [8] perform 3D reconstruction using additional information derived from image intensities or

geometrical regularities consistent with a chosen labeling set. Line labeling methods, however, are not suitable for handling inaccurate drawings and possible missing entities. A missing line may render a complete drawing insoluble. It should be noted that these works are confined to scenes with hidden lines removed (thus simplifying the face identification and line labeling processes), and polyhedral object domain (thus limiting the possible edge and face configurations).

An optimization-based approach for reconstruction from an accurate view has been suggested by Marill [9]. In his work, the depth of vertices of a body represented in a line drawing projection is optimized until the minimum-standard-deviation-of-angles (MSDA) between line pairs at junctions is reached. This approach has been amended by Leclerc [10] to consider face planarity as well; however, these two criteria alone appear to be sufficient for reconstructing only a limited set of object types. An interactive reconstruction system based on line labeling has been implemented by Lamb et al [11]. Their system tries to identify the principal axis, and uses symmetry detection and user intervention in cases of ambiguity. Marti et al [12] have dealt with interpretation of sketches of origami-world objects with hidden lines visible. The interpretation is performed by preprocessing a scanned paper sketch and then using line labeling procedures. A recent work by Grimstead et al [13] constructs a trihedral object from an orthographic view with hidden lines removed. The reconstruction is based on line labeling and iterative least squares solution of a system of equations derived from line labels and image regularities. Work on the general application of machine interpretation of line drawings including non-geometrical data is given by Dori [14].

This work attempts to provide a basis for reconstructing a larger variety of object types depicted in inaccurate sketches with hidden lines visible, as is often the case in mechanical engineering drawings.

## **The proposed approach**

The key principle in the reconstruction process is the perception that although the input edge-vertex graph itself is merely two dimensional, it does hold some implicit spatial information. This implicit information enables a human observer to have a "feel" for the 3D object depicted by the graph. The implicit 3D information originates from three sources:

- Image regularities
- Face topology
- Statistical configuration of entities

In the following sections, image regularity and statistical configuration sources of information are described. For a detailed description of face topology, refer to [2].

## **Image Regularities**

Image regularities are special geometrical relationships between individual entities or within groups of entities. The heuristic rule is that the image regularities do not appear in the drawing accidentally, but rather correspond to some real geometrical regularities existing in the 3D object [7]. An example of a typical regularity is *parallelism*. The

heuristic rule for the *parallelism* regularity is that if two lines are parallel in the sketch plane, they probably represent parallel lines in the 3D object, although mathematically this is not necessarily the case. This heuristic rule has a sound statistical basis: two crossing lines in space will appear parallel only when viewed from a limited scope of viewpoints. Parallel lines in space, however, will appear parallel when viewed from *any* viewpoint. Hence, if two lines are parallel in the sketch plane, it is *more likely* that they represent true spatially parallel lines. Most of the following image regularities are based on similar grounds. The notion of image regularities is so deeply rooted in the human visual system that an image failing to comply with them often perplexes the viewer. An excellent example is found in M. C. Escher's puzzling drawings, where some of the scenes contain parallel lines that do not correspond to parallel lines in the three dimensional scene.

A 3D reconstruction is now sought by moving the vertices of the graph in the depth direction ( $z$ ) perpendicular to the drawing plane while retaining their projected position ( $x, y$ ). The vertices are moved to conform with image regularities detected in the given 2D edge-vertex graph.

Some image regularities have been used (e.g. by Kanade [7]) to create a set of equations, which are then solved to obtain the 3D object from an accurate drawing. This approach, however, has been found inappropriate for practical use when dealing with inaccurate drawing sources. The basic problems with regularities in such cases are: (a) they do not *necessarily* represent a 3D relationship because they might be accidental, and (b) in an inaccurate drawing where parallel lines, for instance, are not exactly parallel in the drawing, it may be difficult to detect these regularities with certainty. As a result of these issues, the information available in image regularities is used as soft constraints which do not necessarily require full compliance. In the proposed algorithm, a set of soft constraints has been constructed by checking all regularities over all the combinations of entities in the topological graph. Each constraint is weighted according to the probability that the regularity detected in the sketch indeed represents a 3D geometrical relationship. For a proposed 3D reconstruction, the regularities are evaluated using mathematical terms, multiplied by their weight-coefficient and summed to produce an overall compliance function. This compliance function estimates how well the specific 3D reconstruction conforms with the regularities identified in its 2D projection. A 3D configuration of vertices is then searched for by optimizing the compliance function. This approach tolerates inaccuracies while arriving at a reconstruction that will approximately satisfy *most* of the regularities. All applications seek to bring the compliance function to an extremum; however, different weighted compositions of regularities in the compliance function may be necessary for different types of sketches. The composition used here is suitable for general purposes (all the examples in Fig. 13 have been computed with the same setting). Better compositions for a more specific object domains can be found. The success of the optimization procedure depends on the variety of regularities used, their mathematical formulation, and the optimization algorithm itself. After a discussion of consistency of interpretation, the regularities are described and discussed.

## Consistency of interpretation

An important principle in the reconstruction process is that interpretation remains invariant despite small changes in the input sketch. An infinitesimal change in a sketch will not make a human observer interpret the sketch differently. Consider the following example: one of the important geometrical relationships between two entities in a sketch is parallelism. Some sketchers in existing CAD systems use  $7^\circ$  of angular difference as a

threshold for determining parallelism, implying that an angular difference of  $7.0001^\circ$  would be interpreted as *not* parallel. Such behavior directly conflicts with the consistency of interpretation principle. To avoid this, a *continuous* compliance factor  $\mu(a)$  is defined, where  $a$  is the angular difference between the two lines in question. The compliance factor  $\mu$  ranges from 1.0 for exact parallelism ( $a=0$ ) and descends to zero like a standard normal distribution curve with  $\sigma = 7^\circ$ , as  $a$  approaches  $90^\circ$ . In computations of the target function, the parallelism criterion would be weighted according to its compliance factor. Intuitively, it could be said that "the more the lines are parallel in the sketch plane, the more they are required to be parallel in space." Thus, by avoiding a clear cut threshold, a small change in the angle of a line would not cause a step difference in interpreting parallelism. A more general formulation of the function  $\mu(x)$  is given by:

$$\mu_{a,b}(x) = e^{-\left(\frac{x-a}{b}\right)^2} \quad (1)$$

where  $x$  represents the value to be checked,  $a$  is a nominal value (e.g.,  $90^\circ$  for perpendicularity), and  $b$  is a reasonable deviation (e.g.,  $7^\circ$ ).

Here  $\mu(x)$  has been defined so that it evaluates to 1.0 when  $x=a$  exactly, and degenerates to 0.0 like a standard distribution curve with  $\sigma = b$  as  $x$  retreats from  $a$ . For practical purposes, equation (1) has been modified to eliminate values close to zero that may otherwise be weak regularities. That is:

$$\mu_{a,b}(x) = \max\left[0, 1.1 \cdot e^{-\left(\frac{x-a}{b}\right)^2} - 0.1\right] \quad (2)$$

The inclusion of the consistency principle is a significant contribution to the robustness of the interpretation.

## Formulation of image regularities

Image regularities are heuristic in their nature and can be classified into three groups:

1. Regularities reflecting some spatial relationship among individual entities. For example, parallelism between two entities is likely to reflect parallelism in 3D.
2. Regularities reflecting some spatial relationship among a selected group of entities, such as entities in a contour or a chain of entities. For example, skewed symmetry of a projected contour is likely to represent a symmetrical contour in 3D.
3. Regularities affecting all the entities in the drawing. For example, the isometry regularity requires that the 3D length of entities will be uniformly proportional to the length of their projection.

Each regularity is expressed by a mathematical term and is included in the compliance function. The term evaluates to a value denoting how well a specific image regularity agrees with its associated 3D configuration. Since the compliance function serves as a target for minimization, the terms included in it must be formulated to exhibit uniform mathematical behavior. Consequently, three guiding principles have been set:

1. All terms must vanish for complete compliance and must evaluate to approximately  $n$  for clear non-compliance, where  $n$  is the number of vertices participating in the evaluation of the term. Consequently, whenever a vertex is repositioned, its

influence is proportional to the number of constraints it affects. It is not mathematically possible to make values of different units comparable. For example, units of length can take arbitrary magnitudes compared with units of angular deviation. In our case, however, the sketch dimensions, in pixels, are limited and are typically around half of the screen width. When such limitations exist, all units can be normalized to become dimensionless and can be successfully weighted and compared.

2. The terms must evaluate the square of the linear deviation from compliance. To achieve linearity, inverse trigonometric functions are often used to compensate for non-linearity of vector operations. Although these functions may slow the computation, they ensure that changes in deviation uniformly affect the model *compliance-function*.
3. The terms must include a weighting factor to accommodate uncertainty in identifying regularity due to sketch inaccuracies.

The following notation is used:

$a$  represents the criteria value. These criteria are summed up and used as a minimization target.

$w$  represents the weight given to a constraint, based on the dominance of its associated regularity.

$v, v'$  are vectors that respectively represent a vertex belonging to a 3D object and its 2D projection in the sketch projection.

$l, l'$  are unit vectors that respectively represent a direction in 3D and in the sketch plane.

Following is a list of image regularities. Each regularity includes an observable geometrical relationship in 2D and the associated configuration presumed in 3D. The mathematical term used to evaluate the regularity and its weighting coefficient are given.

**Face planarity.** A face contour consisting entirely of straight lines reflects a planar face in 3D. The evaluation of this condition is performed in two stages: first, the best-fit surface for the contour vertices is found; then, the deviation of each vertex from that surface is computed, squared, and summed.

The best-fit plane is assumed to be in the form:

$$ax + by + cz + d = 0 \quad (3)$$

The plane coefficients  $a, b, c$  are computed by solving the linear system using the given list of points  $(x_i; y_i; z_i \ i=1..n)$  lying on the plane, and arbitrarily assuming  $d=1$ .

$$\sum \begin{bmatrix} x_i^2 & x_i y_i & x_i z_i \\ x_i y_i & y_i^2 & y_i z_i \\ x_i z_i & y_i z_i & z_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \sum \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (4)$$

The coefficients are then normalized by having  $\sqrt{a^2 + b^2 + c^2} = 1$  with  $d$  scaled appropriately and the deviation of a point from the plane taken as the absolute value  $|ax_i + by_i + cz_i + d|$ .

Note that in order to use this regularity, a preprocessing stage is necessary in which edge-circuits corresponding to the faces of the depicted 3D object are identified *in the 2D graph*. This process is described in detail for general (non-manifold) objects in [2].

Other methods exist for evaluating the planarity of a set of points in space. However, methods relying on checking the planarity of every four-point sequence around the contour (e.g., Leclerc [10]) are not sufficient because local planarity of contour points does not ensure its global planarity. Such a situation may occur whenever there is a sequence of three collinear points.

**Line parallelism.** A parallel pair of lines in the sketch plane reflects parallelism in space. The term used to evaluate the parallelism of a line pair is

$$\alpha_{parallel} = w_{1,2} \left[ \cos^{-1}(\hat{l}_1 \cdot \hat{l}_2) \right] \quad w_{1,2} = \mu_{0^\circ, 7^\circ} \left( \cos^{-1}(\hat{l}_1 \cdot \hat{l}_2') \right) \quad (5)$$

where  $\hat{l}_1$  and  $\hat{l}_2$  are the unit direction vectors of the first and second lines, respectively. The weight  $w_{1,2}$  given to this regularity for a specific pair of lines depends on how the two lines are parallel in the sketch plane.

**Line verticality.** A line that is vertical in the sketch plane (parallel to the y-axis of the drawing page) is "vertical in space," i.e., its two endpoints have similar depth (z) coordinates. The term used to evaluate the verticality of a line is

$$\alpha_{vertical} = w_l \left[ \cos^{-1}(\hat{l}_y) \right] \quad w_l = \mu_{0^\circ, 7^\circ} \left( \cos^{-1}(\hat{l}_y') \right) \quad (6)$$

where  $\hat{l}_y$  is the *vertical* component of the line's direction vector and  $\hat{l}_y'$  is its *vertical* component in the sketch plane.

**Isometry.** Lengths of entities in the 3D model are uniformly proportional to their lengths in the sketch plane. The term to account for non-uniformity corresponds to the standard-deviation of scales.

$$\alpha_{isometry} = n \cdot \sigma^2(r_i) \quad r_i = \frac{\text{length}(\text{entity}_i)}{\text{length}'(\text{entity}_i)} \quad (7)$$

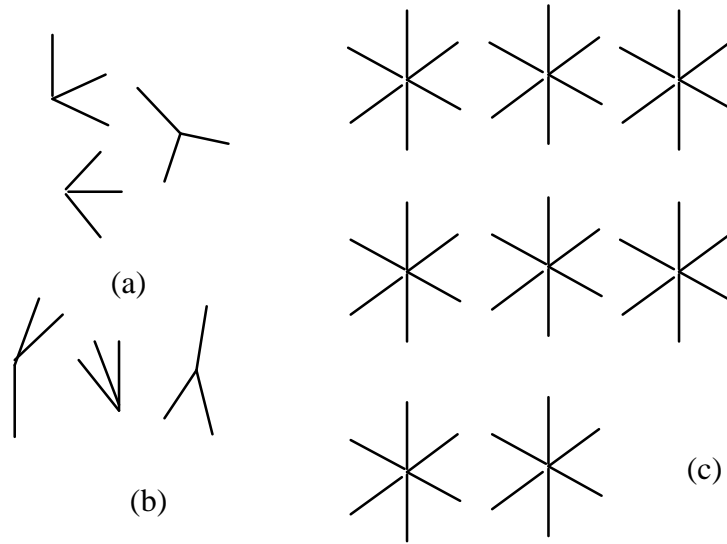
where  $n$  is the number of entities,  $r_i$  is the ratio between the current length of entity  $i$  and its length in the sketch plane, and  $s$  is the standard deviation of the series of  $r_i$ .

**Corner orthogonality.** A junction of three lines that mathematically qualifies as a projection of a 3D orthogonal corner *is* orthogonal in space. To determine whether a junction of three lines in a plane qualifies as a projection of an orthogonal corner, the following test is applied, based on the fact that the projection of an orthogonal corner spans at least  $90^\circ$ . A junction of three lines has eight variants, created by flipping the direction of each line and considering the eight resulting permutations (see Fig. 5). The eight variants of the junction *in the sketch plane* are tested; for each variant, three lines exist  $l'_{i=1..3}$ , forming three pairs between themselves,  $l'_{i=1,2}, l'_{i=2,3}, l'_{i=3,1}$ . Each line is described by a 2D unit direction vector  $\hat{l}_i'$  in the sketch plane, pointing from the junction outwards. If a junction variant spans less than  $90^\circ$  (i.e., is not a projection of an orthogonal corner), all of the three dot-products of its direction-vector pairs will be positive. If a three-line junction is a projection of an orthogonal corner, all of its eight

variants must span at least  $90^\circ$ . Thus, if any one of the eight variants appears to span less than  $90^\circ$ , (shows such an "all-positive" condition), the tested junction is unlikely to represent an orthogonal corner. Consequently, the term used to evaluate the corner orthogonality condition is

$$\alpha_{corner} = w_{corner} \sum_{Pair=1}^3 \left[ \sin^{-1}(\hat{l}_1 \cdot \hat{l}_2) \right]^2 \quad (8)$$

$$w_{corner} = \begin{cases} 1 & \text{if } \beta \leq 0 \\ \mu_{0,0.1} & \text{if } \beta > 0 \end{cases} \quad \beta = \max_{8 \text{ variants}} \left[ \min_{3 \text{ pairs}} (\hat{l}'_a \cdot \hat{l}'_b) \right] \quad (9)$$



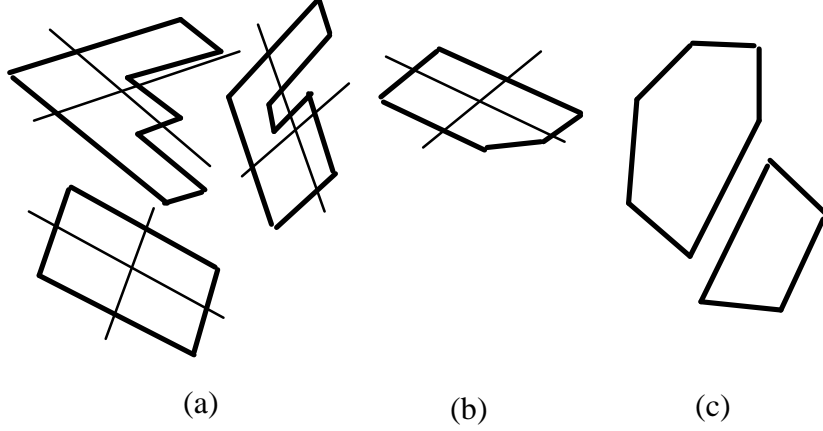
**Fig 5.** (a) Some three-line junctions may appear to form orthogonal corners, (b) some not. (c) A three-line junction has eight variants.

**Skewed facial orthogonality.** A face contour that shows skewed orthogonality is probably orthogonal in space. If entities on the contour of a planar face join only at right angles, then the contour can be said to be orthogonal. If this contour is viewed from an arbitrary viewpoint, it will exhibit skewed orthogonality, as is illustrated in Fig. 6 (a). Faces or entity chains showing skewed orthogonality are easily detected by alternating their boundary lines between two main directions which correspond to the main axis directions of the original shape (see Fig. 6 (a)). The statistical behavior of the alternating values produced by multiplying the scalar-product and the cross-product of adjacent 2D lines in the sketch plane is used for detection. Consistent behavior is likely to represent skewed orthogonality. The amount by which the face is considered to have skewed orthogonality is represented by the value of the weighting coefficient  $w_{skewed \ orthogonality}$ . The terms to evaluate the above for a face are

$$\alpha_{skewed \ orthogonality} = w_{skewed \ orthogonality} \sum_{i=1}^n \left[ \sin^{-1}(\hat{l}_i \cdot \hat{l}_{i+1}) \right]^2 \quad (10)$$

$$w_{\substack{\text{skewed} \\ \text{orthogonality}}} = \mu_{0,0.2}(\sigma(\beta_{i=1\dots n})) \quad \beta_i = (-1)^i \cdot [\hat{l}'_i \cdot \hat{l}'_{i+1}] \cdot [\hat{l}'_i \times \hat{l}'_{i+1}] \quad (11)$$

where  $n$  represents the number of lines along the face contour. Faces in which only part of their contour shows skewed orthogonality, such as in Fig. 6(b), can also be accepted, with the line list  $\hat{l}'_{i=1\dots n}$  reduced to a sub-section along the contour path. This requires detection of sub-sections with consistent  $b_i$  (low  $\bullet$ ).



**Fig. 6** (a) Faces showing skewed orthogonality with their skewed principal axis, (b) partial orthogonality and (c) none.

**Skewed facial symmetry.** A face showing skewed symmetry in 2D denotes a truly symmetrical face in 3D. Algorithms for the detection of skewed symmetry have been the subject of extensive research (see, for example, [15,16]). A simplification used here is that if skewed symmetry exists in a polygonal shape, its axis intersects the contour at two points, each either a vertex or midpoint of an entity. Assuming also that the number of entities on both sides of the symmetry axis in a truly symmetrical shape is equal, the number of possible symmetry-axis candidates is reduced significantly to  $n$ , where  $n$  is the number of vertices in the shape. Each possible candidate symmetry-axis passes through the vertices  $v_k$  and  $v_{k+n/2}$ , where  $k = 1/2, 2/2, 3/2, \dots, n/2$  and, for example,  $v_{2/2} = (v_2 + v_3)/2$ .

The relationship between the vertices of the shape and the candidate symmetry-axis determines whether the axis can serve as a skewed symmetry axis. This relationship is represented, per axis  $k$ , by the weighting coefficient  $w_k$ . The maximal  $w_k$  determines the selected symmetry-axis if the face possesses the skewed symmetry characteristic at all.

$$w_{skewed}^{skewed} = \max_{k=1/2, 2/2, 3/2, \dots, n/2} [w_k] \quad w_k = \mu_{0,0.2} \left( \frac{\sigma(\text{skew}_i)}{i=1/2, 2/2, 3/2, \dots, n/2} + \frac{\sigma(\text{sym}_i)}{i=1/2, 2/2, 3/2, \dots, n/2} \right) \quad (12)$$

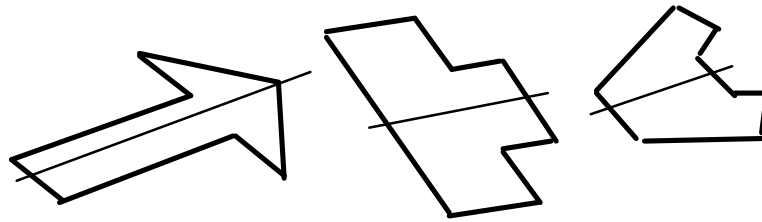
$$\text{skew}_i = \left[ \frac{(v'_k - v'_{k+n/2})}{\|v'_k - v'_{k+n/2}\|} \cdot \frac{(v'_{k+1} - v'_{k-1})}{\|v'_{k+1} - v'_{k-1}\|} \right] \cdot \left[ \frac{(v'_k - v'_{k+n/2})}{\|v'_k - v'_{k+n/2}\|} \times \frac{(v'_{k+1} - v'_{k-1})}{\|v'_{k+1} - v'_{k-1}\|} \right] \quad (13)$$

$$\text{sym}_i = \frac{\text{dist of } v_{k+i} \text{ from axis}}{\text{dist of } v_{k-i} \text{ from axis}} - 1 \quad (14)$$

Note that the vertices  $v'_k$  are in the 2D sketch plane. Two conditions are required for skewed symmetry to occur. First, corresponding points must be evenly distanced from the symmetry axis, a condition denoted by the **sym** term above; second, lines stretched between corresponding points must form a consistent angle with the symmetry axis, a condition denoted by the **skew** term above. If skewed symmetry has been detected, the optimization term will be

$$\alpha_{skewed}^{skewed} = w_{skewed}^{skewed} \sum_{i=1}^{n/2} \left[ \sin^{-1} \left( \left[ \frac{(v_{k+1} - v_{k-1})}{\|v_{k+1} - v_{k-1}\|} \right] \cdot \left[ \frac{(v_k - v_{k+n/2})}{\|v_k - v_{k+n/2}\|} \right] \right) \right]^2 \quad (15)$$

where  $k$  denotes the axis that has been selected.



**Fig 7.** Faces showing skewed symmetry.

**Line orthogonality.** All line pairs in a junction *except those that are collinear* are perpendicular in 3D. This statement does not represent a pure regularity in the sense that it does not depend entirely on the appearance of the entity in the image plane apart from the exception clause. For a junction, the regularity serves mainly to initially "inflate" the flat projection into 3D. The term used for this computation is:

$$\alpha_{junction}^{junction} = \sum_{i=1}^n w_i \left[ \sin^{-1} (\hat{l}_1 \cdot \hat{l}_2) \right]^2 \quad w_i = 1 = \mu_{0^\circ, 7^\circ} \left( \cos^{-1} (\hat{l}'_1 \cdot \hat{l}'_2) \right) \quad (16)$$

where  $n$  is the number of non-collinear pairs of lines meeting at the junction. This regularity is termed MSDP (Minimum Sum of Dot Products).

**Minimum Standard Deviation of Angles.** All angles between all pairs of lines meeting at junctions must be similar (MSDA = Minimum Standard Deviation of Angles). The term used for this computation for the entire body is

$$\alpha_{MSDA} = n \cdot \sigma^2 \left( \cos^{-1}(\hat{l}_1 \cdot \hat{l}_2) \right) \quad (17)$$

where  $\hat{l}_1$  and  $\hat{l}_2$  represent the unit direction vectors of all possible line pairs meeting at vertices of the object. When arcs are introduced into the object domain, their tangents at the endpoints are used in evaluating the MSDA and MSDP. The MSDA criterion has been suggested by Marill [9].

**Face perpendicularity.** All adjacent faces must be perpendicular. Again, this criterion serves to initially "inflate" the flat projection to a convex shape in 3D space from which optimization is more easily achieved. The term used here is

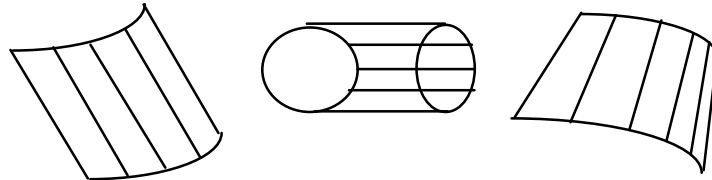
$$\alpha_{Perpfaces} = \sum_{i=1}^n \left[ \sin^{-1}(\hat{n}_1 \cdot \hat{n}_2) \right]^2 \quad (18)$$

where  $\hat{n}_1$  and  $\hat{n}_2$  denote all possible combinations of normals of adjacent faces, and  $n$  is the number of such combinations.

**Prismatic face.** A face joining two parallel elliptic arcs in the projection plane is prismatic in space. Being prismatic, it can resolve both cylindricality and planarity. The term to evaluate this behavior sums up the misalignment of lines. For an exact prismatic surface, this value should become zero.

$$\alpha_{prismatic} = \sum_{i=1}^n \left[ \cos^{-1}(\hat{l}_i \cdot \hat{l}_{i+1}) \right]^2 \quad (19)$$

where  $\hat{l}_i$  denotes an extrusion line and  $n$  represents the number of subdivisions made to the arcs. In our implementation, this condition is sought by segmenting the elliptic arcs at the edge of the cylindrical face into linear components at constant angular steps. Although the arc has been transformed into a group of lines and vertices, the solver ensures they remain on the same plane. The cylindrical face is then approximated with a set of rectangular faces. The parallelism criterion along with the face-planarity criterion serve to implement the prismatic-face regularity.



**Fig. 8** Equi-parameter extrusion lines are parallel for prismatic surfaces.

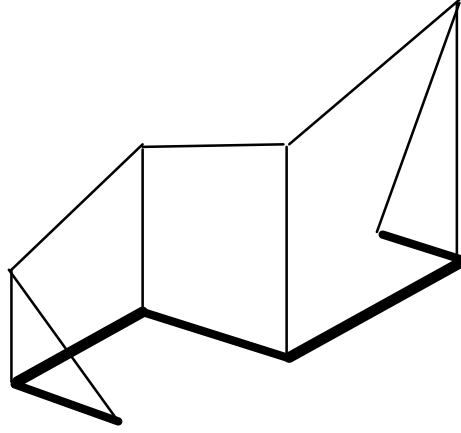
**Line collinearity.** Lines collinear in the sketch plane are collinear in space. The term used to denote this heuristic is

$$\alpha_{collinear} = \sum_{i=1}^n w_i \cdot \max_{j=1\dots 4} \left[ \frac{\det[\bar{v}_j, \bar{v}_{j+1}, \bar{v}_{j+2}]}{\max(\|\bar{v}_j - \bar{v}_{j+1}\|, \|\bar{v}_{j+1} - \bar{v}_{j+2}\|, \|\bar{v}_{j+2} - \bar{v}_j\|)} \right]^2 \quad (20)$$

$$w_i = 1 = \mu_{0^\circ, 7^\circ} \left( \cos^{-1}(\hat{l}'_1 \cdot \hat{l}'_2) \right) \quad (21)$$

where  $n$  is the number of such collinear pairs and  $v_{j=1\dots 4}$  are the four 3D end vertices of the two lines.

**Planarity of skewed chains.** When a *chain* of entities is found to exhibit skewed symmetry of skewed orthogonality, both *planarity* and *symmetry* or *orthogonality* are required, through application of the corresponding formulation above.



**Fig. 9** A skewed orthogonal chain of entities.

## Optimization-based reconstruction

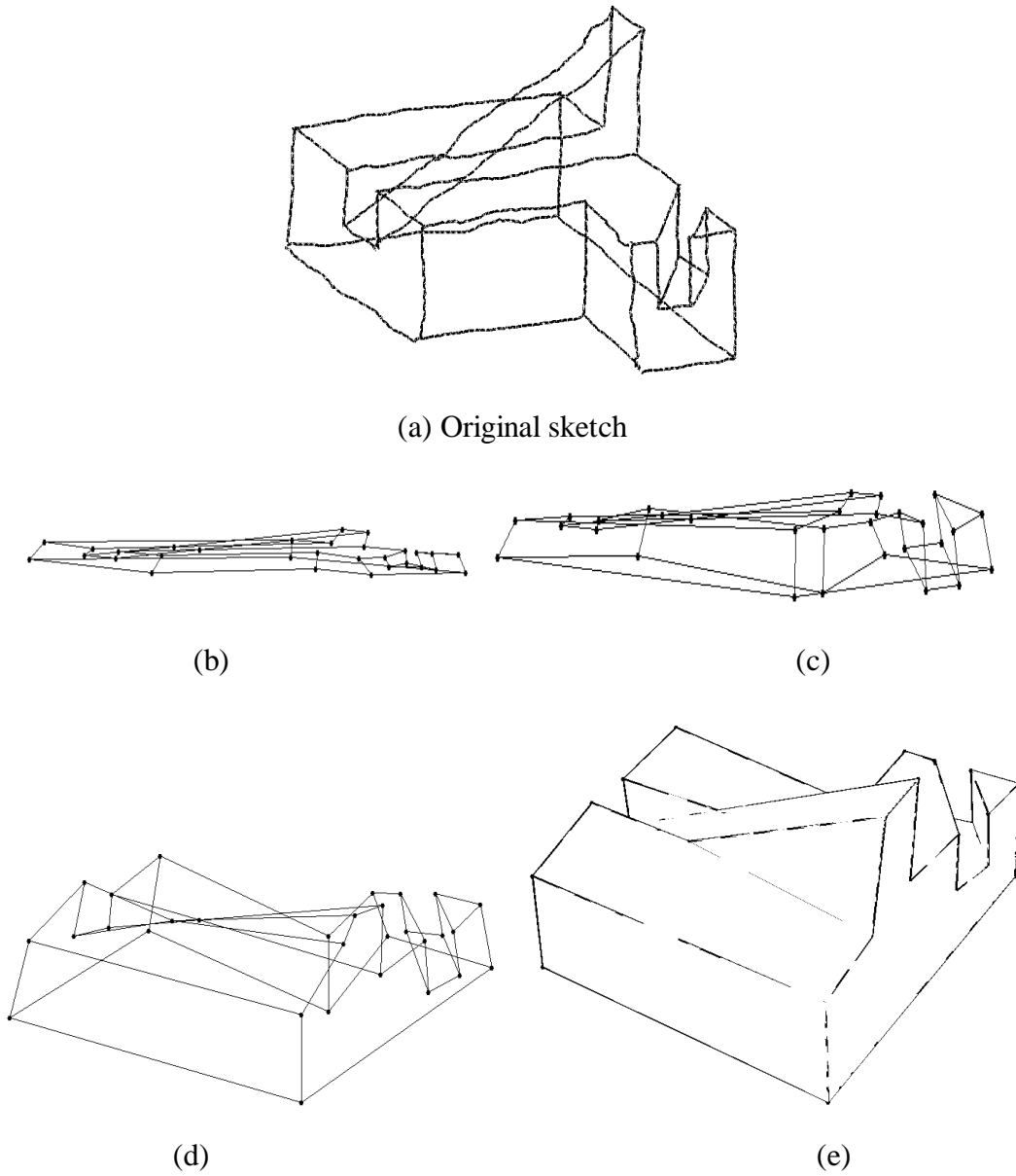
When the reconstruction process begins, the given 2D edge-vertex graph is analyzed and image regularities are identified. For each regularity, the corresponding weighting coefficient is computed according to the formulations presented in the previous section. A 3D configuration can be represented by a vector  $\mathbf{Z}$  containing the  $z$  coordinates of the vertices. A compliance function  $F(\mathbf{Z})$  can then be computed for any 3D configuration by summing the contributions of the regularity terms. Regularities are prefixed by a global balancing coefficient vector  $\mathbf{W}$ . The final compliance function to be optimized takes the form

$$F(\mathbf{Z}) = \mathbf{W}^T \Sigma[\alpha] \quad (22)$$

where

$$[\alpha] = \begin{bmatrix} \alpha_{planarity} \\ \alpha_{parallel} \\ \alpha_{vertical} \\ \alpha_{isometry} \\ \alpha_{corner} \\ \alpha_{skewed} \\ \alpha_{orthogonality} \\ \alpha_{skewed} \\ \alpha_{symmetry} \\ \alpha_{MSDP} \\ \alpha_{MSDA} \\ \alpha_{perpendicular} \\ \alpha_{faces} \\ \alpha_{prismatic} \\ \alpha_{collinear} \end{bmatrix}$$

The appropriate configuration of  $z$ 's (the vector  $\mathbf{Z}$ ) is sought using optimization. Fig. 10 shows various stages of the object as it evolves from a flat 2D projection to a full 3D reconstruction.



**Fig 10.** A 2D sketch "inflated" into 3D by optimization

The process of manipulating the  $z$ 's while seeking the best reconstruction is a full  $n$ -dimensional nonlinear optimization problem, where  $n$  denotes the number of vertices. In practice, the first  $z$  can be arbitrarily set to 0, so an  $n-1$ -dimensional problem is to be solved. Since all the regularities (not the weights, which are constant during the optimization) are defined using continuous and differentiable algebraic terms, the compliance function is well defined and continuous, and its derivatives defined everywhere. Under such circumstances, the optimization procedure is guaranteed to converge, since it can always proceed from the current point downhill in some direction until it cannot go any further. However, the point of convergence is not guaranteed to be the global optimum. This means that the reconstruction algorithm will always return a solution, but not necessarily the correct one. According to the requirements posed on page 5, an incorrect solution can be any collection of lines and faces randomly positioned in space, in a form that happens to comply with the given projection from some point of view. These cannot be considered as valid objects, and some heuristic measures of correctness suggested in [1] can be used to identify such situations and acknowledge

failure or restart at a different point. There is no method, as yet, to determine in advance whether the optimization approach will succeed or fail. Mathematical inconsistency does not necessarily indicate that the object is not reconstructable, as shown in the truncated pyramid example in Fig. 3. Similarly, there are no restrictions to any specific object domain, such as manifold objects or trihedral polyhedrons, there are no means of confirming the consistency of the input graph. The input sketch could depict a skeletal structure with no faces at all. The reconstruction algorithm proposed in this paper relies to some extent on the fact that it is embedded in an interactive system, and the user can remove and re-sketch incorrect reconstructions. Some interesting projections can have several topologically different solutions. Such cases can usually be identified mathematically in advance. Refer to [2] for more details on the analysis of general non-manifold sketch topology. The famous Neckers-Cube illusion can easily be overcome by reversing the signs of the depth coordinates.

Multiple-dimensional nonlinear optimization (without derivatives) is in itself a subject for extensive research and is beyond the scope of this work. Several optimization methods, however, have been examined in order to perform the optimization:

**Brent's minimization** [17]. This one-dimensional minimization technique is based on parabolic interpolation and is adequate for squared deviation formulations of image regularity terms. This method is applied cyclically, vertex by vertex, until the system appears to reach equilibrium. This method cannot be guaranteed to arrive at the global optimum, but it is fairly robust.

**Conjugate gradients** [18]. This method uses a one-dimensional minimization procedure to minimize the target function separately in each direction and then computes a modified set of directions to travel along in the next iteration. If the target function can be minimized faster by traveling along a direction that does not correspond to one of the main axes of the  $n$  dimensions, then this method is likely to find that direction. Again, this method typically converges rapidly but is less robust and may find a local optimum.

**Genetic algorithm** [19]. This algorithm consists of: a mechanism for generating a large population of random objects corresponding to the projection; a mechanism for comparing solutions (the compliance function); and a mechanism for selecting parents and combining them to introduce new solutions to the population. Some individual solutions are randomly mutated from time to time. After a large number of iterations, a good solution can be found among the population. This method is slow but has a better chance of finding the global optimum.

Of the three methods tested, cyclic application of Brent's minimization seemed most promising. As opposed to the other multiple-dimensional, nonlinear optimization techniques mentioned above, Brent's method allows for intervention by various operations, such as control over dynamic vertex ordering and restrictive vertex displacement. The conjugate gradient method generally converges in one-third of the time, but in some cases does not converge at all and thus is less robust. The genetic approach has the important benefit of successfully avoiding local minima which are problematic in multi-dimensional optimization problems. However, for a genetic algorithm to reach a good solution, a very large number of iterations is necessary.

## Complexity of the reconstruction algorithm

The complexity of the reconstruction algorithm is a multiple of two factors: (a) the complexity of the optimization procedure optimizing the compliance function  $C$ , and (b) the complexity of evaluating the compliance function  $C$  itself. The complexity of an optimization procedure is, at least,  $O(n)$  ( $n^2$  for Brent's optimization), where  $n$  is the number of dimensions, or, in this case, the number of vertices. In each evaluation of the compliance function, all the regularity terms are evaluated and summed. Assume that there are  $k$  regularity types; then, in the worst case,  $k$  regularities are observable between each pair of entities in the sketch. Noting that in an object with  $n$  vertices there are  $O(n)$  edges (Euler's equation), it can be concluded that each evaluation of the compliance function may be, in the worst case, of complexity  $O(kn^2)$ , i.e.  $O(n^2)$ , since  $k$  is constant. The overall complexity of the optimization process is thus at least  $O(n^3)$ , depending on the optimization procedure.

## Accelerating and improving convergence: Angular distribution graph

As is evident from the complexity analysis, the basic drawbacks of the optimization procedure are its slow convergence and susceptibility to arriving at local minima. These problems become more severe when the dimensionality of the problem increases, as is the case for complex models with a large number of vertices. A partial solution in an application using sketch input is part-by-part sketching of the input model. Thus, each part in itself is a smaller reconstruction problem.

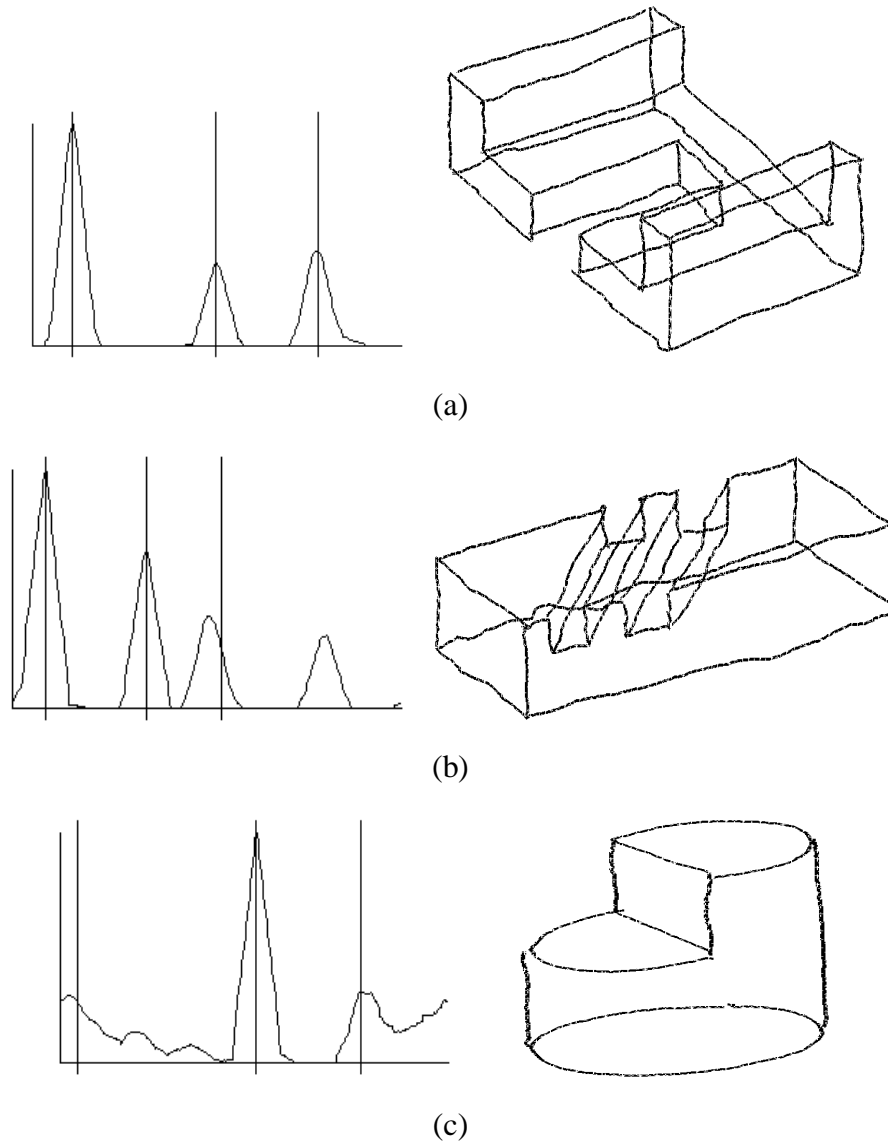
Another approach is based on obtaining a preliminary approximation of the model before starting the optimization process, namely, starting with an improved initial position. As in many numerical nonlinear optimization problems, the distance of the initial guess from the target has the greatest influence both on convergence time and on ability to converge to the global minimum. The initial guess would be a preliminary estimation of the shape of the body represented by the sketch.

The human process of understanding a line drawing may also go through some sort of preliminary understanding phase in which the basic and overall shape is conceptualized but the fine details and features are not yet comprehended. It is not clear how this understanding is accomplished, but it is possible that the human observer identifies some kind of main trend in the direction of the lines corresponding to the main dimensions of the body. The main trend may suggest that the object is basically orthogonal, spherical or perhaps cylindrical in shape. It is also possible that different areas of a sketch encompass different trends, thus implying that the object is perhaps spherical on one side and orthogonally thin and tall on the other. A general sketch stroke trend might also identify some sort of symmetry or extrusion or the existence of two distinct parts separated by an angle of 30 degrees.

To understand the term *trend* more specifically, it is necessary to statistically analyze the direction of strokes in the sketch to find some general consistency. In this study, orthogonality is sought because it is the prevailing trend in most engineering drawings and also the simplest to identify.

The distribution of strokes is analyzed by means of an angular distribution graph (ADG). The ADG is constructed by sampling the angle of every entity in the sketch and plotting it on an angular distribution histogram. When an angle is added to the graph, a Gaussian

distribution curve with  $s=7$  is superpositioned onto the graph to account for the inaccuracy of the sketch. After all the curves have been superpositioned and combined with their different weights, the graph is normalized with its maximum at 1.0. The resulting ADG qualitatively shows three prevailing angles in the sketch, as can be seen in the examples shown in Fig. 11.



**Fig 11.** Angular distribution graphs (ADG's) of various sketches

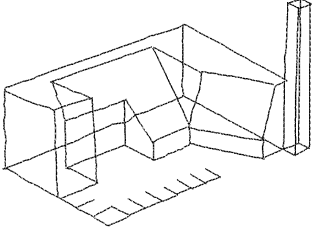
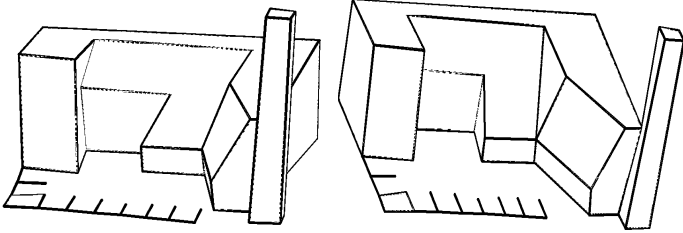
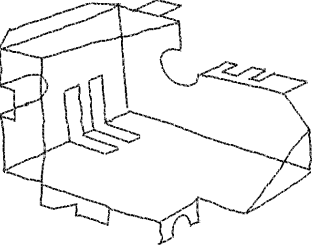
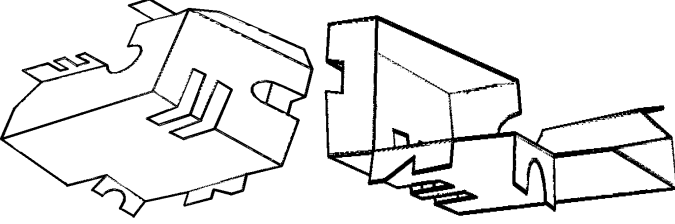
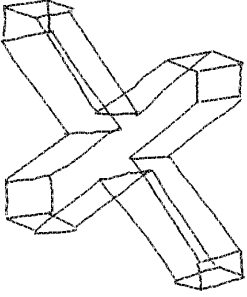
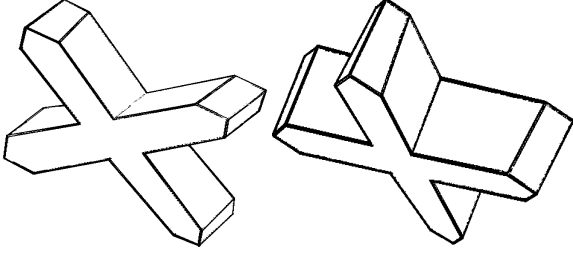
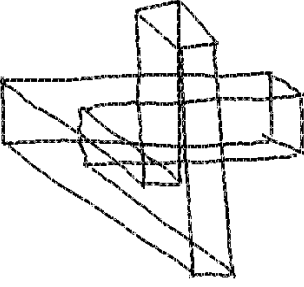
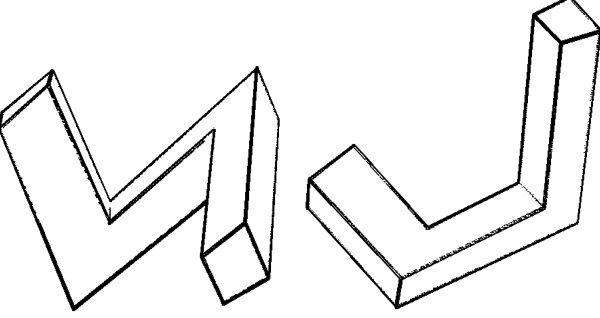
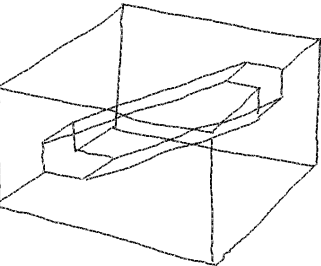
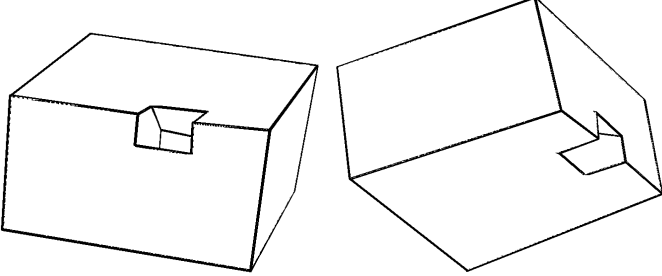
Fig. 11 shows that strictly orthogonal objects have clear three hump ADG's. Other objects exhibiting some orthogonality still show a few noticeable humps surrounded by angular "noise." It is possible to identify the main axis directions of the intended object from this graph. To do so, an axis system must be found at a spatial orientation such that it will be projected onto the sketch plane with angles corresponding to maxima in the ADG.

Once the prevailing axis system is identified, lines in the sketch are associated with one of the axes and assigned a compatibility factor. For example, lines that are found nearly parallel (in the sketch plane) to the X axis are marked as parallel to X, with a compatibility factor denoting the angular difference between the line and the axis (in the

sketch plane). The term "X" is an arbitrary label assigned to any one of the axes. Vertex positions are computed starting at an arbitrary origin point with an arbitrary depth of zero and advancing to adjacent points using lines with the best compatibility factors. The resulting reconstruction process is extremely simple and is equivalent to a sorting procedure with complexity of  $O(n \log n)$ , where  $n$  is the number of vertices. It provides a good initial guess for most typical engineering parts exhibiting some degree of orthogonality. The initial phase of building the ADG is a process of  $O(n)$  and analyzing it is a constant procedure independent of  $n$ . It appears that if this process were to be further developed to identify different groups of faces with corresponding ADG's, a better initial guess could be found for complex parts.

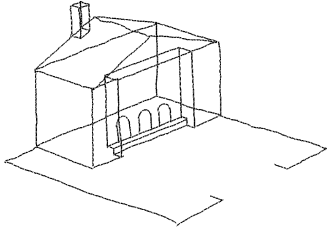
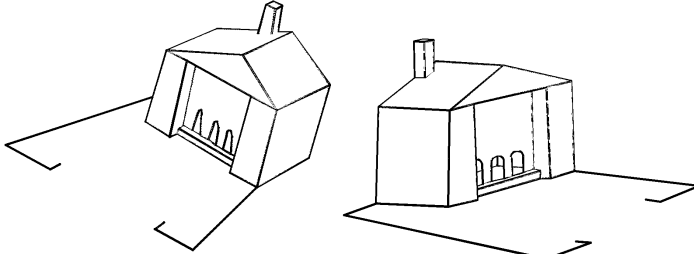
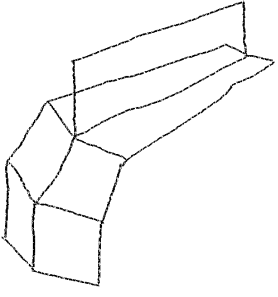
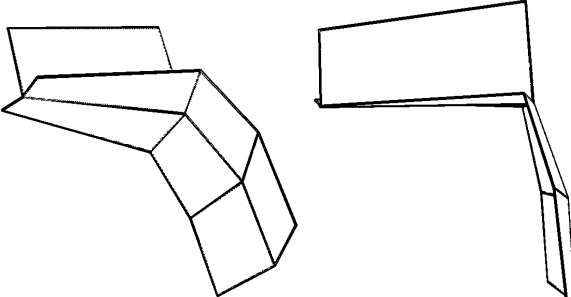
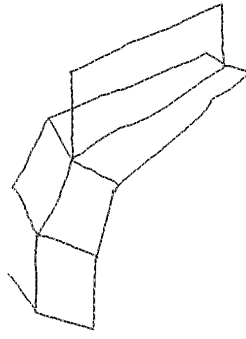
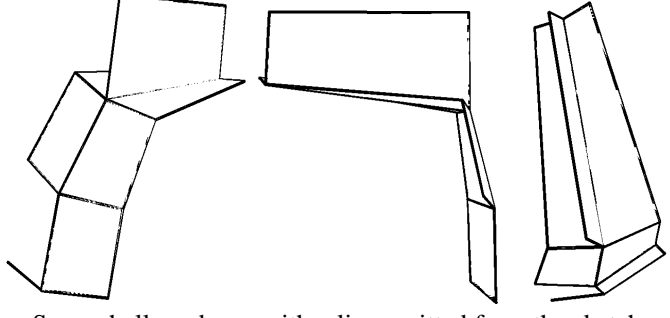
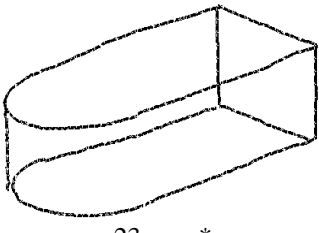
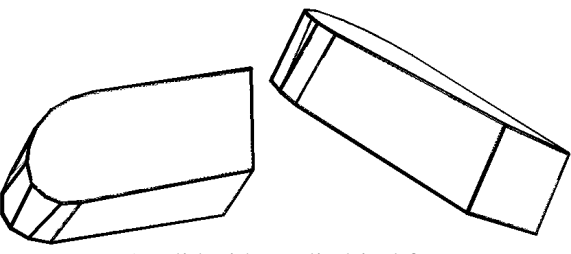
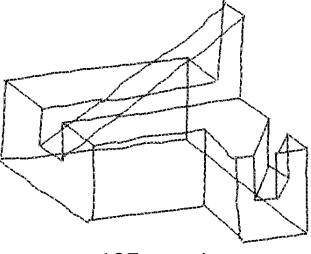
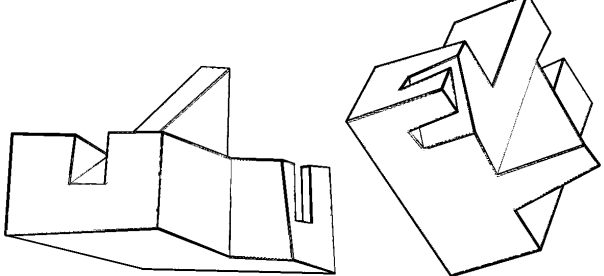
## **Results**

Figure 12 below shows some reconstruction results on a variety of sketches representing objects of various types. The sketches shown here consist of 10 to 70 vertices and are more complex than those appearing in previous publications.

Original 2D Sketch	3D Reconstruction
 <p>530 secs.*</p>	 <p>A mixed-dimensions non-manifold object</p>
 <p>1674 secs.*</p>	 <p>A non-manifold sheet-metal product with holes &amp; some curved edges</p>
 <p>133 secs.*</p>	 <p>A concave extruded solid</p>
 <p>43 secs.*</p>	 <p>An orthogonal object</p>
 <p>45 secs.*</p>	 <p>A manifold solid with a piercing hole</p>

\*Reconstruction time on SG Personal Iris.

**Fig 12.** Some examples of input sketches and their interpretations. The reconstruction was performed from the sketch alone, without any prior knowledge about the 3D object.

Original 2D Sketch	3D Reconstruction
 <p>1183 secs.*</p>	 <p>An open surface model of a house (peep through the door)</p>
 <p>18 secs.*</p>	 <p>A non-manifold shell surface</p>
 <p>12 secs.*</p>	 <p>Same shell as above, with a line omitted from the sketch</p>
 <p>23 secs.*</p>	 <p>A solid with a cylindrical face</p>
 <p>127 secs.*</p>	 <p>A concave non-trihedral solid</p>

\*Reconstruction time on SG Personal Iris.

**Fig 12 (cont'd).** Some examples of input sketches and their interpretations. Note that the reconstruction was performed from the sketch alone, without any prior knowledge about the 3D object represented or its type.

## Summary

In this research, an optimization-based method for reconstructing a 3D object from a single sketch has been developed. The process is based on identifying and formulating geometrical regularities and seeking their associated 3D configuration. The algorithm presented here is capable of reconstructing objects from a wide object domain with considerable fault tolerance. Improved performance has been achieved through ADG analysis prior to processing.

The reconstruction results appear to correctly reflect the concept of the depicted object but tend to produce a somewhat distorted 3D model. In part, this distortion is due to the inherent inaccuracies in the sketch, but it also relates to the failure to accurately distinguish between important and less important regularities. Evidently, more research is required pertaining to this distinction ability. In addition, the reconstruction process is far more prone to errors when the object involves curved faces. This can be attributed to the fact that the majority of regularities used are applicable only to straight line segments. More research in curved-face regularities is necessary as well.

When these difficulties are overcome, we believe that this approach to sketch analysis can be incorporated into CAD systems to produce a new kind of natural user interface aimed particularly at the conceptual design stage. This type of interface will enable designers to communicate more freely with the computer and among themselves.

## Acknowledgments

This work was carried out in partial fulfillment of the requirements for the degree of D.Sc. in Mechanical Engineering at the Technion [Hod Lipson]. This research has been supported in part by the Fund for the Promotion of Research at the Technion, Research No. 033-028.

## References

- [1] **Lipson, H and Shpitalni, M** 'A new interface of conceptual design based on object reconstruction from a single freehand sketch' *Annals of the CIRP* Vol. 44/1 (1995) pp 133-136
- [2] **Shpitalni, M and Lipson, H** 'Identification of faces in a 2D line drawing projection of a wireframe object' submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 1995
- [3] **Nagendra, I V and Gujar, U G** '3D objects from 2D orthographic views - a survey' *Computer Graphics* Vol 12 No 1 (1988) pp 111-114
- [4] **Wang, W and Grinstein, G** 'A survey of 3D solid reconstruction from 2D projection line drawings' *Computer Graphics forum* Vol 12 (1993) pp 137-158
- [5] **Huffman, D A** 'Impossible objects as nonsense sentences' *Machine Intelligence* Edinburgh Univ Press (1971) pp 295-323

- [6] **Clowes, M B** 'On seeing things' *Artificial Intelligence* Vol 2 (1971) pp 79-116
- [7] **Kanade, T** 'Recovery of the three dimensional shape of an object from a single view' *Artificial Intelligence* Vol. 17 (1980) pp 409-460
- [8] **Sugihara, K** *Machine Interpretation of Line Drawings* MIT Press (1986)
- [9] **Marill, T** 'Emulating the human interpretation of line drawings as three dimensional objects' *Int. J. of Computer Vision* Vol 6 No 2 (1991) pp 147-161
- [10] **Leclerc, Y G and Fisler, M A** 'An optimization based approach to the interpretation of single line drawings as 3D wire frames' *Int. J. of Computer Vision* Vol 9 No 2 (1992) pp 113-136
- [11] **Lamb, D and Bandopadhyay, A** 'Interpreting a 3D object from a rough 2D line drawing' *Proc First IEEE Conf on Visualization 90* (1990) pp 59-66
- [12] **Marti, E, Regomcós, J, López-Krahe, J and Villanueva, J J** 'Hand line drawing interpretation as three dimensional object' *Signal Processing* Vol 32 (1993) pp 91-110
- [13] **Grimstead, I J and Martin, R R** 'Creating solid models from single 2D sketches' *Proceedings of the Third Symposium on Solid Modeling Applications, ACM Siggraph* pp 323-337
- [14] **Dori, D** 'Dimensioning analysis - toward automatic understanding of engineering drawings' *Communications of the ACM* Vol 35 No 10 (1992) pp. 92-102
- [15] **Friedberg, S A** 'Finding axes of skewed symmetry' *Computer Vision, Graphics, and Image Processing* No 34 (1986) pp 138-155
- [16] **Yip, R K K, Tam, P K S and Leung, D N K** 'Application of elliptic Fourier descriptors to symmetry detection under parallel projection' *IEEE PAMI* Vol 16 No 3 (1994) pp 277-286
- [17] **Brent, Richard P** *Algorithms for Minimization without Derivatives* Prentice Hall, Englewood Cliffs NJ (1973) Ch 7
- [18] **Polak, E** *Computational Methods in Optimization* Academic Press New York (1971) Ch 2-3
- [19] **Holland, J H** *Adaptation in Natural and Artificial Systems* Univ of Michigan Press Ann. Arbor (1975)