Three-dimensional Interpretation of an Imperfect Line Drawing

Thesis by

Leung Kin Lap

In Partial Fulfillment of the Requirements
for the Degree of
Master of Philosophy

Systems Engineering and Engineering
Management Department,
The Chinese University
of Hong Kong

June 1996

Acknowledgments

I would like to thank Dr. Ronald Chung, my thesis advisor, who made this thesis possible. He provided me with very valuable positive feedback and suggestions.

I would like to thank my colleagues, Rambo Ho, Jimmy Li, Neol Ho, Benny Yuen and Kaiser Lee for providing a friendly environment to work in.

Finally, I would like to thank my friend Connie Cheng. She supported me emotionally and psychologically when I had difficulties and made my studies meaningful.

Abstract

Recovering three-dimensional (3-D) shape of an object from a single line drawing is a classical problem in computer vision. Studies on it are abundant. They range from Huffman-Clowes junction labeling, to Kanade's gradient space and skew symmetry analysis, to Sugihara's necessary and sufficient condition for a realizable polyhedral object, to Marill's MSDA shape recovery procedure, to Leclerc-Fischler's shape recovery procedure which assures planar faces, and to the recent Baird-Wang's gradient-descent algorithm which has a favorable time complexity. Yet all these assume perfect line drawings as the input. In this thesis, a method is proposed that through the use of iterative clustering interprets an imperfect line drawing of a polyhedral scene. It distinguishes the true surface boundaries from the extraneous ones like the surface markings, restores the missing surface boundaries, and recovers 3-D shapes satisfying constraints of planarity of faces and parallel symmetry of lines, all at the same time. Experiments also show that the 3-D interpretation agrees with human perception.

Table of Contents

Table of Figures

Chapter 1 Introduction

Line drawing is the simplest form of picture to represent the shapes of three dimensional (3-D) objects. It consists of only straight lines, curves and vertices.

Humans can interpret line drawings or cartoon-like pictures with ease and without the feeling of much loss of information about object shapes. This shows that even if other cues like shading and texture are absent, contours alone already convey much information about shapes, and it is especially true for objects which display significant degree of regularity. Such an ability, if emulated in a machine, can have applications ranging from mechanical drawings interpretation (CAD-from-drawing) to autonomous navigation in man-made structures.

There have been a great deal of work on such a line-drawing interpretation problem, mostly for polyhedral scenes. Huffman [12] and Clowes [10] separately proposed a junction labeling method to recover 3-D description of polyhedral objects, though the description is qualitative. Mackworth [20] and Kanade [14] proposed gradient space analysis and skew symmetry to help recover quantitative 3-D description. Sugihara [26] provided an algebraic criterion as a necessary and sufficient condition for a line drawing to represent a physically realizable polyhedral object. Marill [22] went further to propose a method of reconstructing polyhedral objects; the method is based on a criterion called the MSDA criterion which minimizes the standard deviation of the object's internal angles in 3-D. Later, a few important variations of the MSDA method were proposed. Leclerc and Fischler [18] added a planarity term to the optimization criterion, which enforces each surface to be planar. Recently, Baird and Wang [1] proposed a gradient-descent, algorithm to implement the MSDA criterion, and improved much on the time complexity.

However, all these methods assume perfect line drawings. In real images detectable edges are not necessarily surface boundaries; they can come from surface markings, cracks, shadows, imaging noise, and others. Some of the surface boundaries may not even be directly detectable because of too weak contrast. Yet human perception seems not to require a perfect line drawing to function; for many line drawings it would tell surface boundaries and markings apart, and it would fill-in the missing boundaries if necessary.

1.1 Contributions of the thesis

There already exist techniques like region segmentation [25], edge detection [7, 24], and perceptual grouping [16] that would locate apparent edges in an image and even extract a number of closed regions of defined homogeneity. Such techniques do not infer about 3-D shapes; they simply return a line drawing which is most possible based on image information like intensity profile and co-linearity of lines. Such a line drawing is likely to be corrupted, in the sense that some detected lines are not surface boundaries and some surface boundaries are not detected. Questions are, given such an imperfect line drawing, how surface boundaries can be distinguished from other contours, how missing boundaries can be filled-in, and how 3-D information can be inferred about the objects.

The traditional belief is that these can be done in a sequential manner: segmentation comes first which identifies the true surface boundaries, then the above shape recovery techniques follow which recover 3-D information.

Here a different point of view is taken. It is argued that segmentation has to return boundaries which are discontinuities of depth or orientation, while shape recovery has to know where the true surface boundaries are, a typical chicken-and-egg problem. From this perspective, the two processes should work in parallel cooperatively to come up with a consistent output.

This thesis addresses the problem of interpreting an imperfect line drawing of a polyhedral scene. A mechanism is proposed that, given a line drawing extracted based on the image information alone, would distinguish surface boundaries from surface markings and other extraneous lines, infer about missing surface boundaries, and recover 3-D shapes of the objects in the line drawing. The mechanism consists of an iterative clustering procedure that allows the segmentation and shape recovery processes to come into mutual agreement, before which one process acts on the intermediate result of the other in turns. It should be mentioned that the proposed algorithm does not condier the case of missing vertices or extra vertices. Actually even human has difficulty in interpreting such line drawings.

The following assumptions are made in this thesis. The input line drawing is in the form of a set of closed polygons as extracted from [16]. It is also assumed that the image projection process can be approximated as an orthographic one, i.e., the object size is small compared with the viewing distance so that the object point with camera coordinates $[x, y, z]^T$ projects to the image point $[x, y]^T$.

Part of the work in this thesis has appeared in [8, 9].

1.2 Organization of the thesis

In Chapter 2 some of the major previous work on polyhedral scene understanding is outlined. In Chapter 3, I describe what shape constraints are important for solving the problem, how to formulate the problem, and how to solve it using an iterative clustering approach. In Chapter 4 some experiments illustrating the performance of the algorithm are described. The conclusion and some possible future work are given in Chapter 5.

Chapter 2 Previous Work

A major task of computer vision is to recover the 3-D object from the 2-D projection. Although many different methods have been proposed, the problem is far from solved. In this section, an overview of the related literature is given.

2.1 An overview of 3-D interpretation

It is well known that some information, in particular the depth information, lost in 2-D projection process of a 3-D object. It results that the reversed process is an ill-posed problem. To solve this problem, some assumptions must be made, some properties of the scenes must be known or multiple images must be taken. There is a variety of properties that may be exploited to derive the shape. The properties collectively give rise to a class of algorithms called "Shape from X", namely, Shape from Motion, Shape from Stereo, Shape from Shading, Shape from Texture and Shape from Contour. The first two Shape from X's require multiple images.

2.1.1 Multiple-View Clues

Shape from Motion and Shape from Stereo require multiple images of the objects. The elementary study of motion in images can be found in the book written by Ullman[27]. The pioneering work on stereo machine Vision was performed by Marr and Poggio[22]. Basically, the two methods- Shape from Motion and Shape from Stereo compare the corresponding points in the images with difficult views of the objects and thus the depth data is obtained. However, it may be difficult to obtain multiple images at occasions. Also some experiments show that a man with a single eye can still perceive 3-D shapes correctly. That means multiple images are not always necessary in human perception.

2.1.2 Single-View Clues

Shading, texture and contour are the clues existing in a single image. Shape from shading [32] uses the information about the property of surface (reflectivity function). Shape from Texture requires regular pattern on the images[1,11, 32]. Both of these two cues, however, require strong assumptions about the scene. Shape from Shading requires accurate modeling of the incident illumination and surface photometry, which is difficult to do for many natural scenes. Determining surface Shape from Texture requires presence of regular textural elements.

The last one, shape from contour is what this thesis is about. A surface contour is the image of a curve across a physical surface, such as the edge of a shadow cast across a surface, a gloss contour, wrinkle, seam, or pigmentation marking. Several psychophysical demonstrations show that Shape from Contour is significantly more powerful than Shape from texture. Similarly, Barrow and Tenenbaum[3] show that when there is a conflict between the clues (shading and contour, texture and contour), contour properly dominates human perception. Biederman[5] claims that in the experiments with humans the recognition of a fully colored image of an object is not faster than the recognition of the line drawing of the object. The line drawing interpretation problem is classified into the category of Shape from Contour.

2.2 Line Drawing Interpretation

Different Algorithms have been proposed to give a 3-D interpretation of the line drawing. The previous work can be classified into two categories: qualitative and quantitative.

2.2.1 Qualitative Interpretation

The major work on qualitative interpretation is the junction labeling scheme proposed by Huffman [12] and Clowes [10] separately. The scheme classifies each line on a line drawing into one of four labels ('+', ' - ', '→', '←'); '+' means the line is convex to the viewer, ' - ' means the line is concave, '←' means the line is an occluding edge with the occluding face on the right of the line, and '→' is similar to '←' but with the occluding face on the left. For a trihedral scene in which every vertex consists of three surfaces, there are four types: V, W, Y and T junctions. Catalog is produced listing for each type of junction sets of possible interpretations to the emanating lines. There are totally 16 possible different junction labels for trihedral objects. The labeling allows topology of the 3D objects to be understood. An example of this labeling scheme is shown in figure 2.1. Later, Waltz [31] extended the labeling scheme to include shadows and cracks, and devised an efficient labeling procedure, called filtering. Malik [21] further extended it for curved objects. It is overwhelming that Waltz's catalog, enumerated nearly 3000 physical interpretations of the junctions. On the other side, T. Kanade[15] uses the labeling technique on the so-called Origami world (objects made of single sheet paper). His theory uses the concept of selecting surfaces as basic components of the world, rather than the conventional solid polyhedra. Figure 2.2 shows a 'W-folded paper' line drawing and a 'box' line drawing which are included by Kanade into the labeling catalog.

It should be noted that the labeling scheme is based on a necessary, but not sufficient, condition for a line drawing to represent a physically realizable scene. That means that a line drawing which can be labeled may be an impossible object. (See figure 2.3) Moreover, qualitative method only gives the topology of the 3-D objects but not the actual shape of the objects. Also, qualitative description is not sufficient for many applications; quantitative description is sometimes desirable.

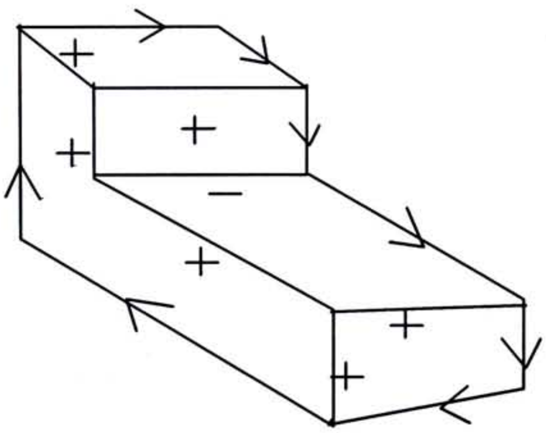


Figure 2.1 : An example of the line labeling of an L-shaped block.

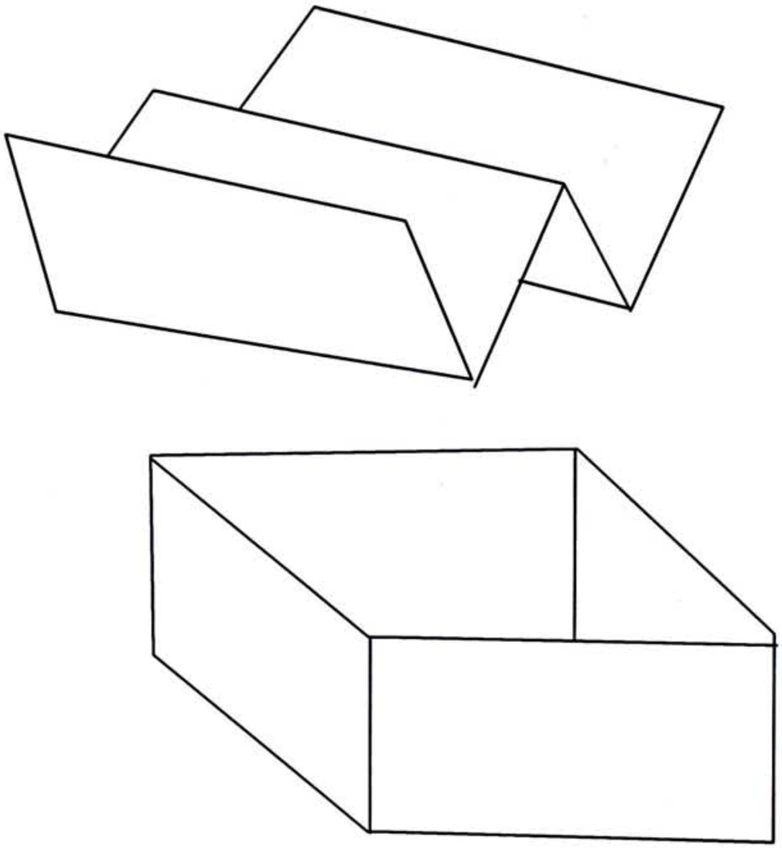


Figure 2.2. Two examples of line drawings in "Origami" world which is considered non-labellable in Huffman and Clowes labeling method.

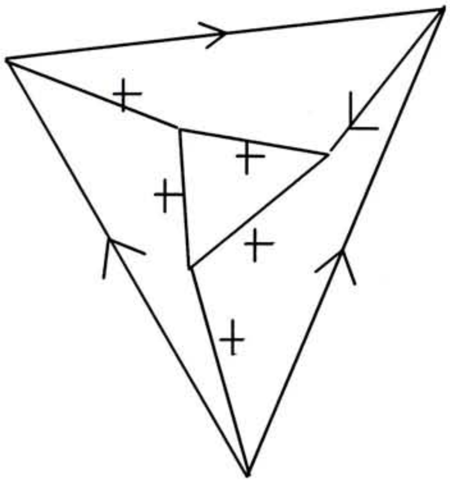


Figure 2.3 : An example of labellable drawing but an impossible object. It is impossible because the faces cannot be all planar.

2.2.2 Quantitative Interpretation

Huffman [13] introduced the gradient space to represent surface orientations. The definition of the gradient space is given in the appendix A.1. One of the most useful properties of the gradient space is the properties of dual lines. It states that if two planes meet and the intersection line is projected to a picture line L, then the gradients of the two planes are on a gradient-space line which is perpendicular to L. Mackworth and Kanade [14] employed it with the use of the shared boundary constraint and skew symmetry to compute quantitative measures about the object, and such measures also supplement the labeling scheme on the correctness of the labeled line drawing. Sugihara [26] later provided an algebraic criterion as a necessary and sufficient condition for a line drawing to represent a physically realizable polyhedral object.

On the shape recovery side, some have suggested that it is a minimization process. Brady and Yuille [6] proposed the use of an extremum principle in determining the surface orientation from its 2-D contour. It maximizes the ratio of the area of a closed figure to the square of its perimeter. Barrow and Tenenbaum [6] proposed another similar measure. However, such measures only deal with a single surface; they do not apply to the entire object. A 3-D object consists of a number of surfaces, and it would require a global method applied towards the surfaces as a whole to recover the object shape. The globalness of the method is even more essential if the line drawing is contaminated, as it is not even known which contours are surface boundaries.

Recently, Marill [22] used a simple objective function to reconstruct the entire 3-D object. The function is simply the standard deviation of all angles (SDA) in the recovered 3-D object, where an angle refers to the one between any two lines of a vertex. The goal is to find the shape with the minimum SDA: the so-called MSDA principle, the idea behind which is to recover the shape that is most three-dimensional. Later, Baird and Wang [1] proposed a gradient-descent algorithm to implement the MSDA principle with much less time complexity. However, the MSDA principle sometimes fails to give a "correct" interpretation (in the sense of whether it is in agreement with human perception) even for simple line drawings, as it only enforces angle symmetry but not planarity of faces. Leclerc and Fischler [18] added a planarity term to the objective function. The new objective function gives extremely good results for perfect line drawings. Ulupinar and Nevatia [28, 29] proposed methods that even reconstruct curved objects. All such shape recovery methods share the commonality that they treat the set of contours as a whole. However, neither of them work with imperfect line drawings.

Since the work in this thesis can be classified as quantitative interpretation of line drawing, this previous work on this class is described in details in next section.

2.3 Previous Methods of Quantitative Interpretation by Optimization

Optimization plays an important role in the line drawing interpretation because a lot of work (including the work in this thesis) use this technique.

Two important methods on the line drawing interpretation are to be described in this section. Basically they both use the optimization process.

2.3.1 Extremum Principle for Shape from Contour

Brady and Horn survey the use of extremum principles in image understanding. The choice of performance index or measure to be extremized, and the class of functions over which the extremization takes place, are justified by appealing to a model of the geometry or photometry of image forming and constraints such as smoothness. There are several plausible measures of a curve that might be extremized in order to compute Shape-from-Contour. First, $\oint \kappa^2 ds$, where κ is the curvature of the contour and s is the contour, has been investigated as a curve of least energy for interpolating across gaps in plane curves. However, contrary to what appears to be a popular belief, this measure is not extremized in the plane that transforms the ellipse into a circle.

Another possible measure is proposed by Barrow and Tenenbaum [3]. Assuming planarity (the torsion is zero), it reduces to

$$\oint \left(\frac{d\kappa}{ds} \right)^2 ds$$

However, this measure involves high-order derivatives of the curve. This means that it is overly dependent on small scale behavior. Consider, for example, a curve which is circular except for a small kink. The circular part of the curve will contribute a tiny proportion to the integral even when the plane containing the curve is rotated. The kink, on the other hand, will contribute an arbitrary large proportion and so will dominate the integral no matter how small it is compared with the rest of the curve.

Because of these weaknesses of these two measures, Brady and Yuille[6] propose another simple measure as:

$$M = \frac{area}{(perimeter)^2}$$

The area, as well as the perimeter, can be obtained by an integral round the contour. If **n** is the normal to the curve then it can be shown that

$$(area)n = \tfrac{1}{2}\oint r \times dr$$

For all possible curves it is maximized by the most compact one, a circle. In general, given a contour, the extremum principle with this measure will choose the orientation in which the projected contour maximizes M. For example, an ellipse is interpreted as a slanted circle. The tilt angle is given by the minor axis of the ellipse. It can also be shown that a parallelogram is interpreted as a rotated square and a triangle as a slanted equilateral triangle.

As mentioned before, these extremum principles only consider individual faces. It does not apply to a whole object. Clearly, in an entire object the faces link and constrain each other. It also requires the line drawing to be perfect. Any missing or extra lines on it will greatly affect the correctness of the interpretation.

2.3.2 MSDA Algorithm

This algorithm is first proposed by Marill[22]. It consists of two components, an objective function and a simple descent optimization procedure for finding a local minimum of this objective function. The objective function is simply the standard deviation of all of the angles (SDA) in the recovered 3-D object with respect to their common mean. Marill calls the minimization of the SDA the MSDA principle. The angle α_{ij} is obtained by $\cos^{-1}(\mathbf{u_i} \bullet \mathbf{u_j})$ where $\mathbf{u_i}$ and $\mathbf{u_j}$ are 3-D unit vectors parallel to the lines connected to a vertex. There is no any absolute constraint in Marill's principle and therefore the recovered objects often are not realized objects or agreed with human perception.

Later Leclerc and Fischler see the defect of the principle. They add a planarity term DP on the objective function as an absolute constraint. The term DP is the sum of the terms DP_i where DP_i is zero when the face f_i is planar, and increases as the face deviates from planarity. They define DP_i as

$$DP_i = \left[(n-2)\pi - \sum_j \alpha_j \right]^2$$

where n is the number of sides in the face i and α_j is the angle at the j th vertex.

The new objective function becomes

$$E(\lambda) = \lambda \cdot SDA^2 + (1 - \lambda)DP$$

where λ is decreased from one to a small value.

If the line drawing is perfect, the 3D interpretation by this algorithm agrees with the human perception. In practice, it is not so ideal because the main problem occurs during the process of edge detection, i.e., the process of getting the line drawing from the intensity image. An example of its failure for an imperfect line drawing is shown in figure 2.4.

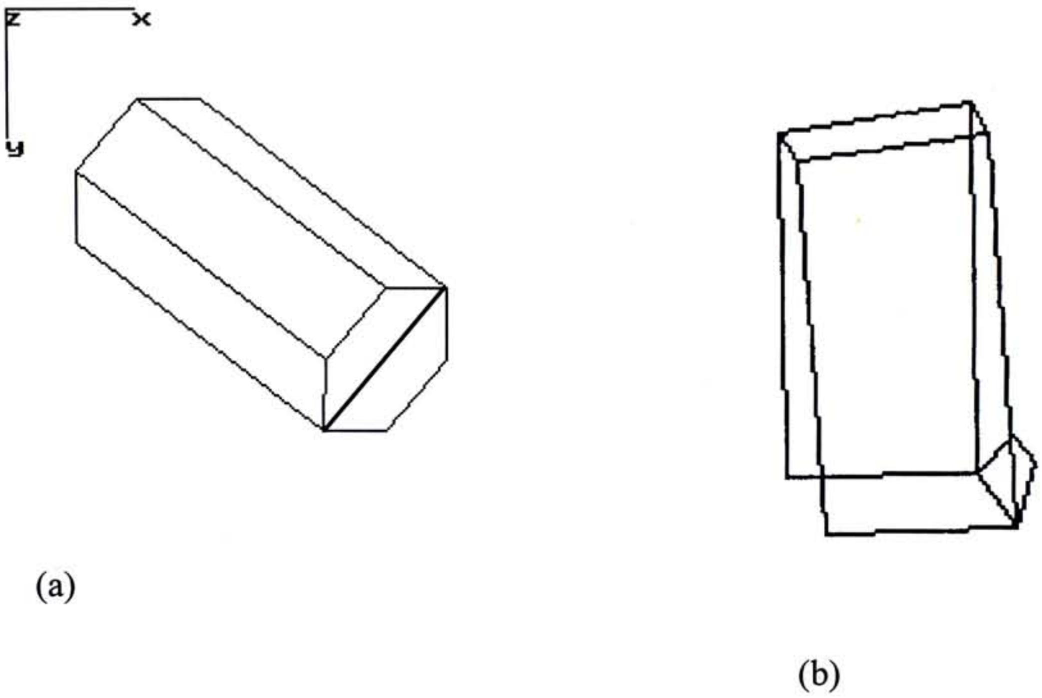


Figure 2.4 : (a) The line drawing of a imperfect line drawing and (b) one view of its Leclerc and Fischer's reconstruction

There are some cases that the algorithm fails to converge, for example, a single trapezoid face(figure 2.5). The minimization of the function will cause the two parallel lines to be separated at very long distance such that the four angles are close to be equal. This failure is due to the fact that the feature of equi-length is not included in the objective function. In other words, the diversity of lengths should be minimized as that of the angles.

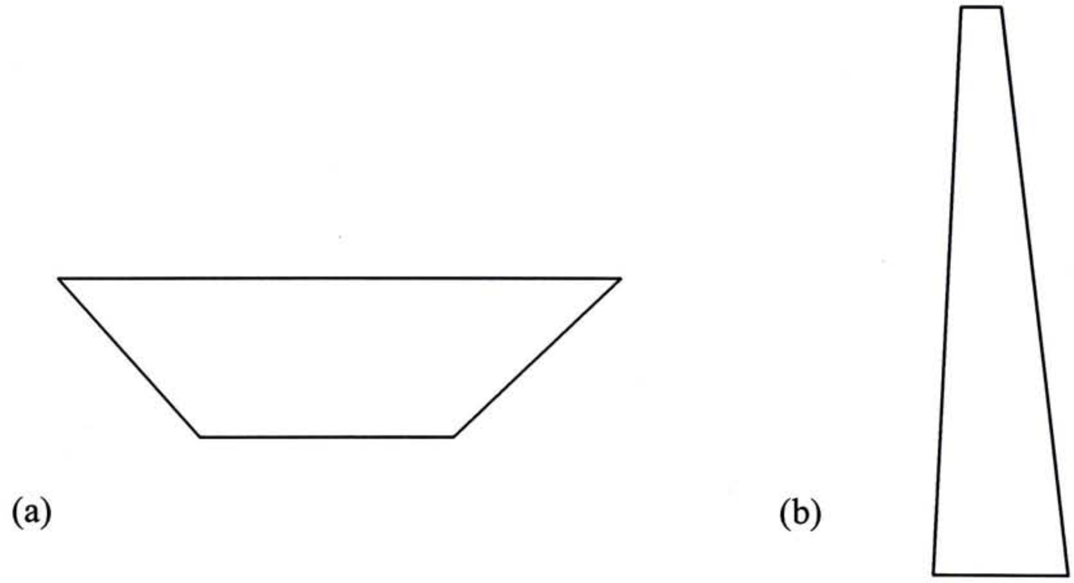


Figure 2.5: This shows the case of the invalidity of the objective function.
(a) The trapezoid line drawing.
(b) The output of the MSDA's principle.

2.4 Comments on Previous Work on Line Drawing Interpretation

The failure of the MSDA algorithm as shown in Figure 2.4 is not a special example. The problem is that it actually assumes the line drawing to be perfect from the very beginning. Not only the MSDA algorithm, but all the previous work described above require perfect line drawings to work with. However, imperfect line drawing interpretations are inevitable in real imageries. To the best of my knowledge, the work in this thesis is the first attempt on imperfect line drawing interpretation.

Chapter 3 An Iterative Clustering Procedure for Imperfect Line Drawings

In this chapter, a new iterative clustering algorithm to interpret an imperfect line drawing as 3-D polyhedral object is described in details. Since the objects being handled are polyhedral, there are no curved surfaces or curved edges. The input of the algorithm is certainly a line drawing. It consists of closed polygons. In other words, it consists of only interconnected straight lines. The line drawing needs not be perfect; there can be surface markings on it and there can be missing surface boundaries. Such a line drawing is available through perceptual grouping techniques like that in [16] which bridge gaps among edges and return hypotheses of closed contours in an image. It can also be obtained using some simple procedures that will be described in the later section.

In the following I first describe what constraints about a 3-D shape are important, and are sufficient as the minimum basis to constrain the shape recovery problem. The shape should be a realizable shape, and should also agree with human perception. Then the line drawing interpretation problem will be formulated in formal terms based upon such constraints. Finally an iterative scheme that makes use of clustering to solve the problem is proposed. Besides, throughout the procedure, it is assumed that the 2-D projection is orthographic.

3.1 Shape Constraints

Recovering 3-D description from a 2-D line drawing is an ill-posed problem; there are an infinite number of different 3-D objects that can give the same 2-D projection (See figure 3.1).

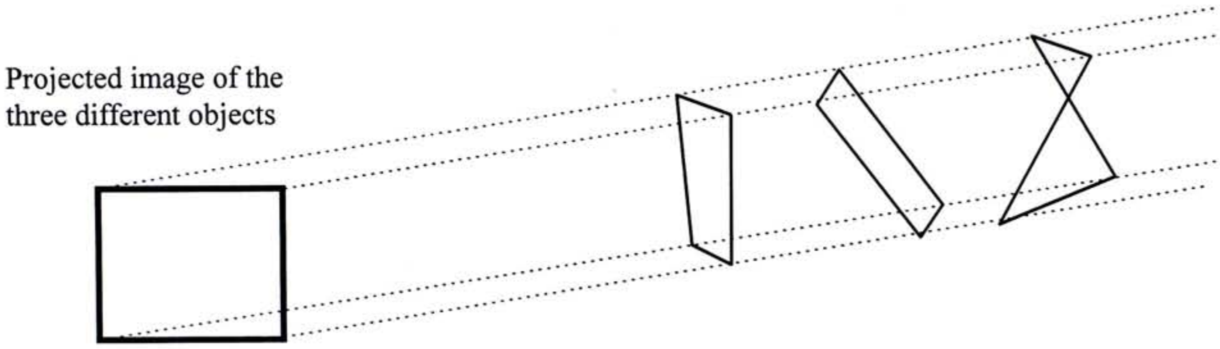


Figure 3.1: Three different objects give the same projection.

Assumptions about the object shape are necessary to constrain the problem. The object that best fits the assumption is considered as the real object. Because of dealing with polyhedral objects, planarity of all surfaces is naturally a fundamental constraint. Under the general viewpoint assumption where observed features and their inter-relationships are stable upon slight perturbations of the viewpoint, 2-D parallelism under orthographic projection can only be projected by 3-D parallelism. Actually it is possible that a pair of 3-D non-parallel lines can project a pair of 2-D parallel line but in only one particular viewpoint. This assumption is also in coherence with the Gestalt school of psychology [17] which proposed that parallel symmetry plays an important role in human perception. Planarity of surfaces and 2-D parallelism to be projected by 3-D parallelism are therefore the two constraints that a sensible 3-D output should satisfy, and they largely narrow the solution space to a few solution points.

However, the original image itself, as a planar object, is one that satisfies all the above constraints. Such a solution is unlikely in reality. It is also not in agreement with human perception; if a line drawing that contains high degree of symmetry as that of a box is given, human does not perceive a set of lines on a plane but a solid object in 3-D. Another term that discourages such a planar object interpretation is needed. To achieve this, a measure of three-dimensionalness of an object is required, similar to Marill's SDA measure, to encourage a 3-D interpretation. If the shape is to be recovered using an optimization scheme, such a measure should be of zero value if the object is planar, and of large value if the object is a solid one with small eccentricity in the 3-D space.

3.2 Problem Formulation

The problem is formulated as a constraint satisfaction problem: given a graph $G = (V, E)$ where $V = \{ v_p : v_p=(x_p,y_p) \}$ are the vertices in the image and $E = \{(v_p,v_q) : v_p , v_q \in V\}$ are the edges among the vertices, the goal is to find a depth measure z_p for each vertex v_p, as well as a set ξ of surfaces in the scene, which is a family of subsets of E (in fact a covering of E), such that they satisfy the above constraints. To put it more formally, I want to come up with a hypergraph $H = (V', E, \xi)$, where $V' = \{ v'_p : v'_p = (x_p, y_p, z_p)\}$ and ξ is a set of hyperedges on E, such that it satisfies the following three constraints.

1. Planarity Constraint

(a) *Vertices in a surface $\varepsilon \in \xi$ are coplanar.*

This can be formulated as: boundary edges of any surface should orient in 3-D such that all connected pairs of them have the same cross-product. That is, the measure

$$\text{PLN}(\{z_p\},\ \xi) = \frac{\sum_{\varepsilon \in \xi} \sum_{i} \{1 - \left[\frac{(e_{\varepsilon,i-1} \times e_{\varepsilon,i}) \cdot (e_{\varepsilon,i} \times e_{\varepsilon,i+1})}{\left\| e_{\varepsilon,i-1} \right\| \left\| e_{\varepsilon,i} \right\|^2 \left\| e_{\varepsilon,i+1} \right\|} \right]^2 \}}{\sum_{\varepsilon \in \xi} \sum_{i} 1}$$

where $e_{\xi,,i}$ is the *i*th edge in 3-D of surface ε, should ideally be zero and practically be a small value. (See figure 3.2)

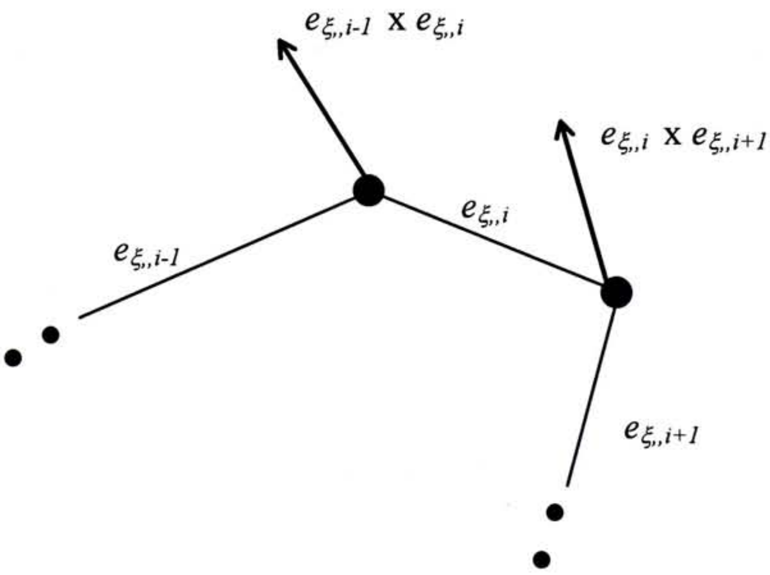


Figure 3.2: Vectors of the cross products of two adjacent edges.

(b) *No two hyperedges sharing edges, i.e., no two neighboring surfaces, are coplanar.*

2. Parallelism Constraint: *Parallel lines in the line drawing are projected by parallel lines in 3-D.*

This can be formulated as: lines parallel in the line drawing should orient in 3-D such that they have unity normalized dot-product. That is, the measure

$$\mathrm{PRL}(\{z_p\}) = \frac{\sum_{i,j}\left\{1-\left[\frac{\mathrm{e}_i \cdot \mathrm{e}_j}{\|\mathrm{e}_i\|\|\mathrm{e}_j\|}\right]^2\right\}}{\sum_{i,j} 1}$$

Where e_i and e_j, are the 3-D orientations of any pair of edges that appear to be parallel in the line drawing, should ideally be zero and practically be a small value. The part $\frac{\mathrm{e}_i \cdot \mathrm{e}_j}{\|\mathrm{e}_i\|\|\mathrm{e}_j\|}$ in the equation is the cosine of the angle between the two edges i ,j. If the edges are parallel, this value should be equal to unity.

3. Maximum Three-dimensionalness: *A solid object is preferred over a sheet object.*

Since a measure that encourages three-dimensionalness in the interpretation is needed but the measure is not supposedly satisfied exactly, it needs not follow Marill's MSDA principle strictly which is more expensive to implement due to its globalness. Instead, a simpler measure which is local to each vertex is used: the constituent edges of any vertex should orient in

3-D such that the triple product of their directions should have a large value. That is, the measure

$$3\text{-Dness}(\{z_p\}) = \frac{\sum_p \sum_i \left[\frac{\left(e_{p,i-1} \times e_{p,i}\right) \cdot e_{p,i+1}}{\left\|e_{p,i-1}\right\| \left\|e_{p,i}\right\| \left\|e_{p,i+1}\right\|} \right]^2}{\sum_p \sum_i 1}$$

where $e_{p.i}$ is the *i*th edge in 3-D of the *p*th vertex, should be maximized. The measure can have a value ranging from 0 to 1; it is of value 1 if the vertex turns out to be rectangular, and of value 0 if the vertex turns out to be flat.

Notice that the first two constraints, planarity and parallelism, are absolute constraints to be satisfied exactly, while the last one, three-dimensionalness, is merely an optimization term to encourage solid object interpretation. It should be noted that this function does not change the $\{z_p\}$ if the line drawing consists of isolated faces like the one in figure 3.3. An isolated face is defined as a face which does not have any share boundary with other faces.

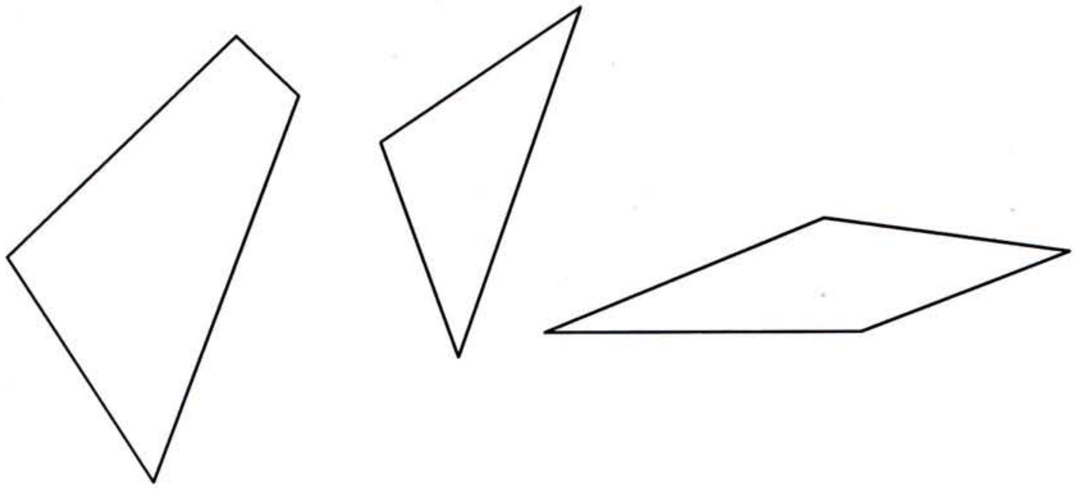


Figure 3.3: An example of line drawing with only isolated faces.

Besides, there is another general viewpoint assumption that can be made. It is the co-linearity: if three points are colinear in 2-D projections, they are also colinear in 3-D. This assumption is not included here as a constraint. There are two reasons.

i) If this constraint is formulated as a function, it will greatly increase the computation time of the objective function.

ii) The case that there are no edges connecting the three colinear vertices does not happen often. For the case that there are edges connecting them, the parallelism also captures the feature of the co-linearity.

Three consecutive points (A, B, C) are co-linear

<==> slope of line AB = slope of line BC

<==> AB// BC

and hence parallelism includes the co-linearity for the three points.

3.3 Solution Steps

If segmentation ξ is known, the problem becomes simple. z_p's are found to minimize the value of the overall objective function

$$E(\{z_p\}, \xi) = -\ 3\text{-Dness}(\{z_p\}) + \alpha\ \text{PRL}(\{z_p\}) + \beta\ \text{PLN}(\{z_p\}, \xi) \qquad (3.1)$$

which is a weighted sum of the above constraints.α and β are the weighting factors with positive values. It should be noted that the term 3-Dness has a negative coefficient because it is to be maximum.

For any α,β, a locally optimal solution of z_p's can be found using the hill-climbing method. The method is simple, and it does not require the derivative expression of the objective function that can be complicated. It can be described as the following. Given any initial value of the solution vector $\mathbf{Z} = (z_1, \ldots, z_p, \ldots, z_{\|V'\|})$, $2\|V'\|$ new solution vectors are formed by adding or subtracting a small value Δz from one of its entries.

The new solution vectors are:

$(z_1 + \Delta z, z_2, \ldots, z_p, \ldots, z_{\|V'\|})$
$(z_1 - \Delta z, z_2, \ldots, z_p, \ldots, z_{\|V'\|})$
$(z_1, z_2 + \Delta z, \ldots, z_p, \ldots, z_{\|V'\|})$
$(z_1, z_2 + \Delta z, \ldots, z_p, \ldots, z_{\|V'\|})$
.
.
.
$(z_1, z_2, \ldots, z_p, \ldots, z_{\|V'\|} + \Delta z)$
$(z_1, z_2, \ldots, z_p, \ldots, z_{\|V'\|} - \Delta z)$

The vector giving the minimum value of the objective function is selected as the new current vector.

The procedure repeats until the current solution vector is stable. Since what comes out is only the locally optimal solution closest to the initial solution vector, the initial values of z_p's are crucial. It will be discussed in later section.

Such a locally optimal solution of z_p's does not assure the absolute constraints of planarity and parallelism be satisfied exactly. To assure that, the penalty method can be used: the weights α and β are increased from small values gradually in small steps until they are at large values, while the locally optimal solution of z_p's is computed at each (α, β)-setting and carried forward. As the weights increase, the absolute constraints dominate the objective function in a smooth fashion. What is missing in the above scheme is that ξ the segmentation solution is not known. Each innermost cycle of the edges, appeared to be a single surface, can be just part of a large surface if some of the apparent edges are merely surface markings. On the other hand, it can be the boundary of more than one surface if some surface boundaries inside it are missing. Segmentation requires knowledge of the 3-D shape, while estimating 3-D shape requires knowledge of segmentation. The ultimate solution will be the state where the two come to a mutual agreement. In light of these, one possible method is to have iterations over a number of intermediate shapes and segmentations until they get to a stable, consistent state. Fortunately, even without knowledge of segmentation, as seen from the objective function (3.1) above, the parallelism and three-dimensionalness constraints can still serve as one absolute constraint and one optimization term to be applied to the input line drawing and a coarse shape description can be constructed. Such a shape description does not have the notion of planar surfaces, but can be close enough to the true shape for subsequent segmentation purpose because of the highly restrictive constraint of parallelism. Obviously, such a scheme would fail if there are no parallel lines in the input line drawing. However, It is conjectured that even humans have difficulty in perceiving shape precisely from a line drawing which is without much regularity or symmetry. With such an approximate shape and the line drawing, the edges can be grouped into different surfaces based on the criterion that edges on the same surface should be more or less coplanar. To facilitate finding out which planes in 3-D the edges vote for, each vertex can be subdivided into a number of L-subvertices, where each L-subvertex is merely the corresponding vertex point itself plus a pair of the vertex's constituent edges. Defining such L-subvertices has the advantage that each of them uniquely defines a plane in the 3-D space.

Then the segmentation problem is treated as a clustering problem, in which the extracted L-subvertices are to be grouped into different surfaces according to how close their corresponding planes are. The resulting segmentation may not be entirely correct, but it introduces the concept of approximate surfaces to the original line drawing. As a result, it allows markings contained in the extracted surfaces to be deleted and missing boundaries of some open surfaces to be added. In other words, a new line drawing is constructed, which is likely to be closer to the true line drawing than the original one. Such a line drawing can be input to the approximate shape recovery process described above and all procedures are repeated until the shape recovered and the revised line drawing are stable, i.e., the segmentation and shape recovery processes agree with each other. Then the planarity term is added to arrive to the final shape.

The approach therefore consists of iterations of the entire set of contours over three steps, as outlined in Figure 3.4, until they come into agreement with one another:

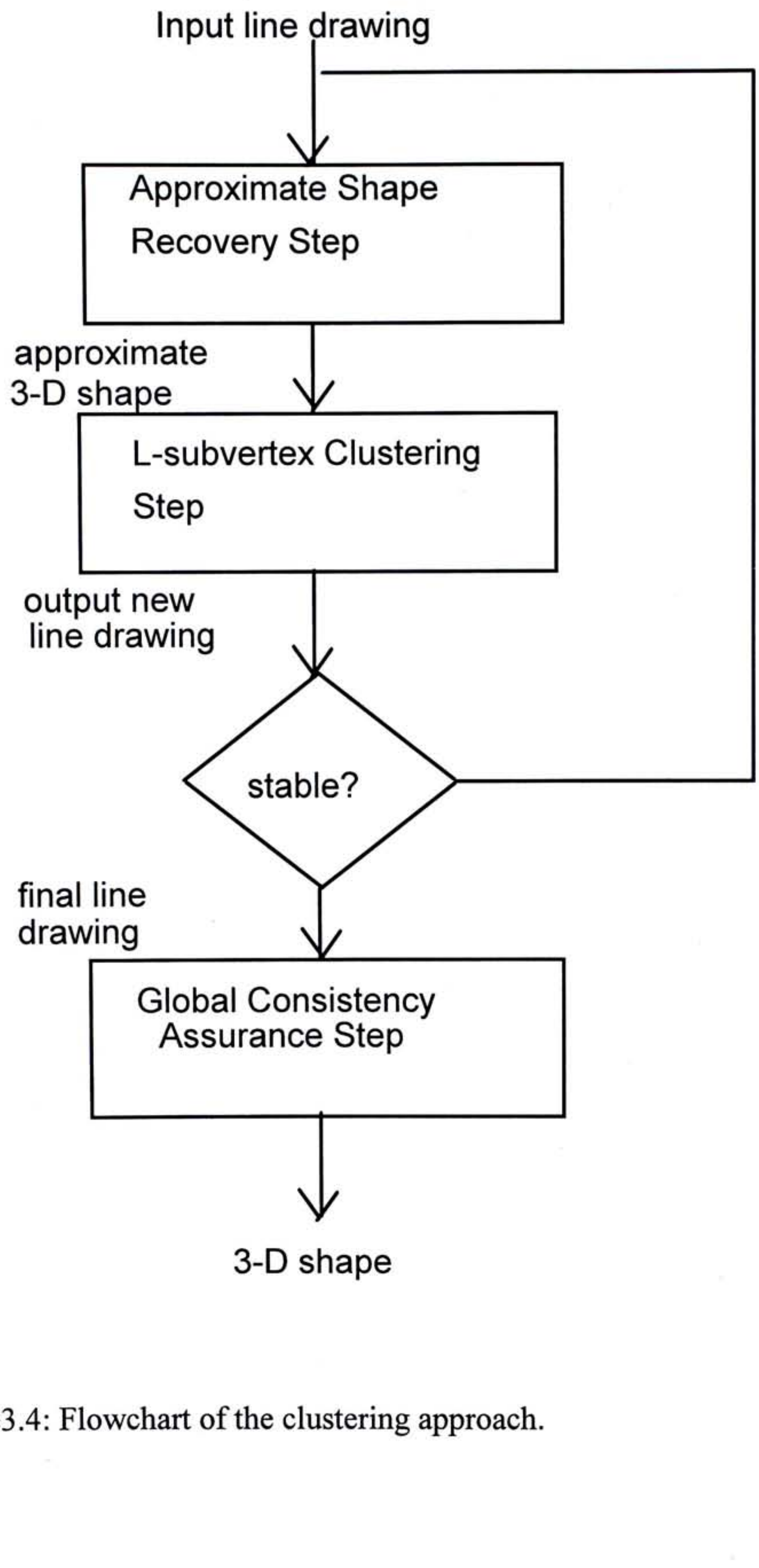


Figure3.4: Flowchart of the clustering approach.

Step 1. Approximate Shape Recovery Step

Given the current line drawing, initially the input line drawing, find an approximate shape that satisfies the parallelism constraint exactly and optimizes the three-dimensionalness constraint. The penalty method plus the hill-climbing algorithm described above can be used. In this step, β is set to be zero while α is increased gradually.

Step 2. L-subvertex Clustering Step

Given the current approximate shape and line drawing, extract the L-subvertices and group them into different surfaces using the simple clustering algorithm [17] outlined in Section 3.4.

Each L-subvertex with 3-D position v_p and 3-D orientations$\{v_{p1},\ v_{p2}\}$ represents a 4-D unit vector $[a, b, c, d]^T$ corresponding to the plane ax +by + cz + d = 0 in space, whose values can be obtained from

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \frac{\begin{bmatrix} v_{p1} \times v_{p2} \\ -(v_{p1} \times v_{p2}) \cdot v_p \end{bmatrix}}{\left\| \begin{bmatrix} v_{p1} \times v_{p2} \\ -(v_{p1} \times v_{p2}) \cdot v_p \end{bmatrix} \right\|}$$

The inter-pattern distance measure for clustering can be the magnitude of the cross-product of such 4-D unit vectors the two L-subvertices represent. With the surface segmentation, the surface boundaries are extracted in the following way. For each surface ε, the smallest polygon that encloses all the edges in it is determined, and the set of edges which overlap with the polygon's boundary is defined as the boundary of the surface. Such surface boundaries may not be all closed. New line segments are then added to close the boundaries of open surfaces, and line segments are removed if they are contained inside surfaces. A new line drawing is formed.

If the new line drawing is different from the previous one, go to step 1, else go to step 3.

Step 3. Global Consistency Enforcing Step

Given a perfect line drawing, find the 3-D shape that is globally consistent and that is optimal: it satisfies constraints of parallelism and planarity exactly and maximizes the three-dimensionalness measure. Again, the penalty method plus the hill-climbing algorithm can be used. This can be regarded as a finishing step of the shape description.

In the above steps, two points are worth noting:

1. Free Vertices:

   A vertex with only two constituent edges is not counted towards three dimensionalness, as it does not involve multiple surfaces. If none of its constituent edges is parallel to other edges in the line drawing, such a vertex is also not involved in the parallelism constraint. As a result, this vertex does not "move" in Step 1. Such a vertex is called as a free vertex. With the concept of planarity of a face, the free vertex can be moved to a reasonable position by assigning it with a surface normal which is the mean of the surface normals of two L-subvertices connected to it through its two constituent edges. This is illustrated in Figure 3.5. There may be a number of such L-subvertices. The two with the most similar surface normals are chosen. A free vertex may be connected to another free vertex. In that case the surface normals are propagated from the non-free vertices to the free vertices sequentially.

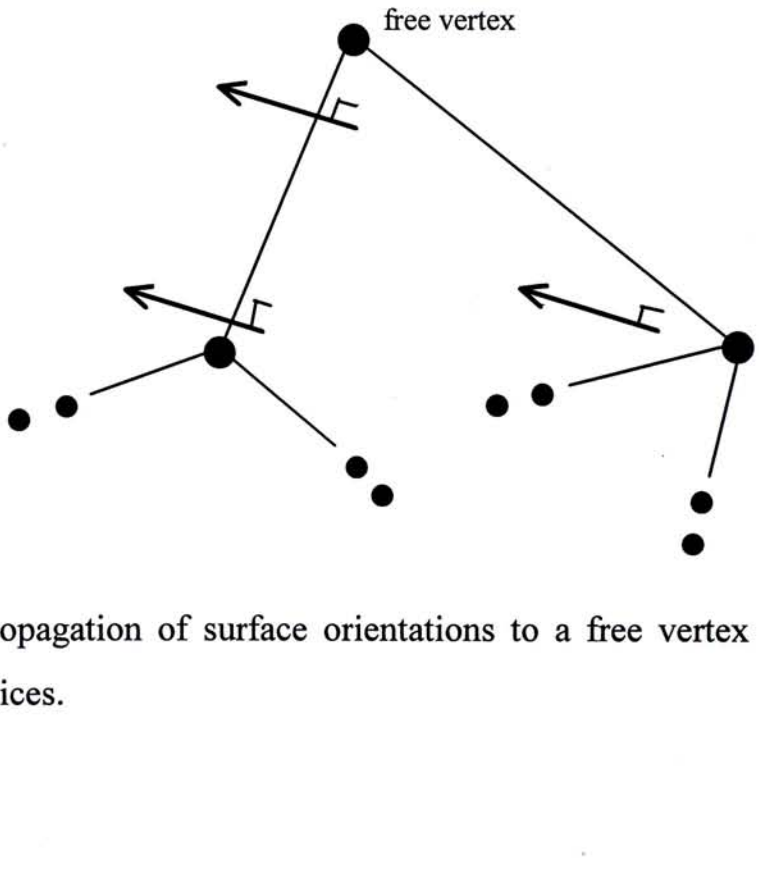


Figure 3.5: Propagation of surface orientations to a free vertex from the connected vertices.

2. Interpretation Ambiguity and Initial Depth Values for Optimization:

A phenomenon that is related to the famous Necker's cube problem (see figure 3.6) is, for some line drawings there are 3-D interpretations which both satisfy the constraints of planarity, parallelism, and three-dimensionalness, and yet are entirely different. Imagine the line drawing of a rectangular box: the vertex in the middle can be interpreted as convex or concave. However, there is a slight preference of a convex vertex in human perception. A reason may be that the inside of objects is often not visible.

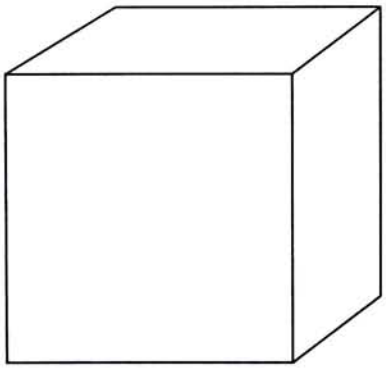
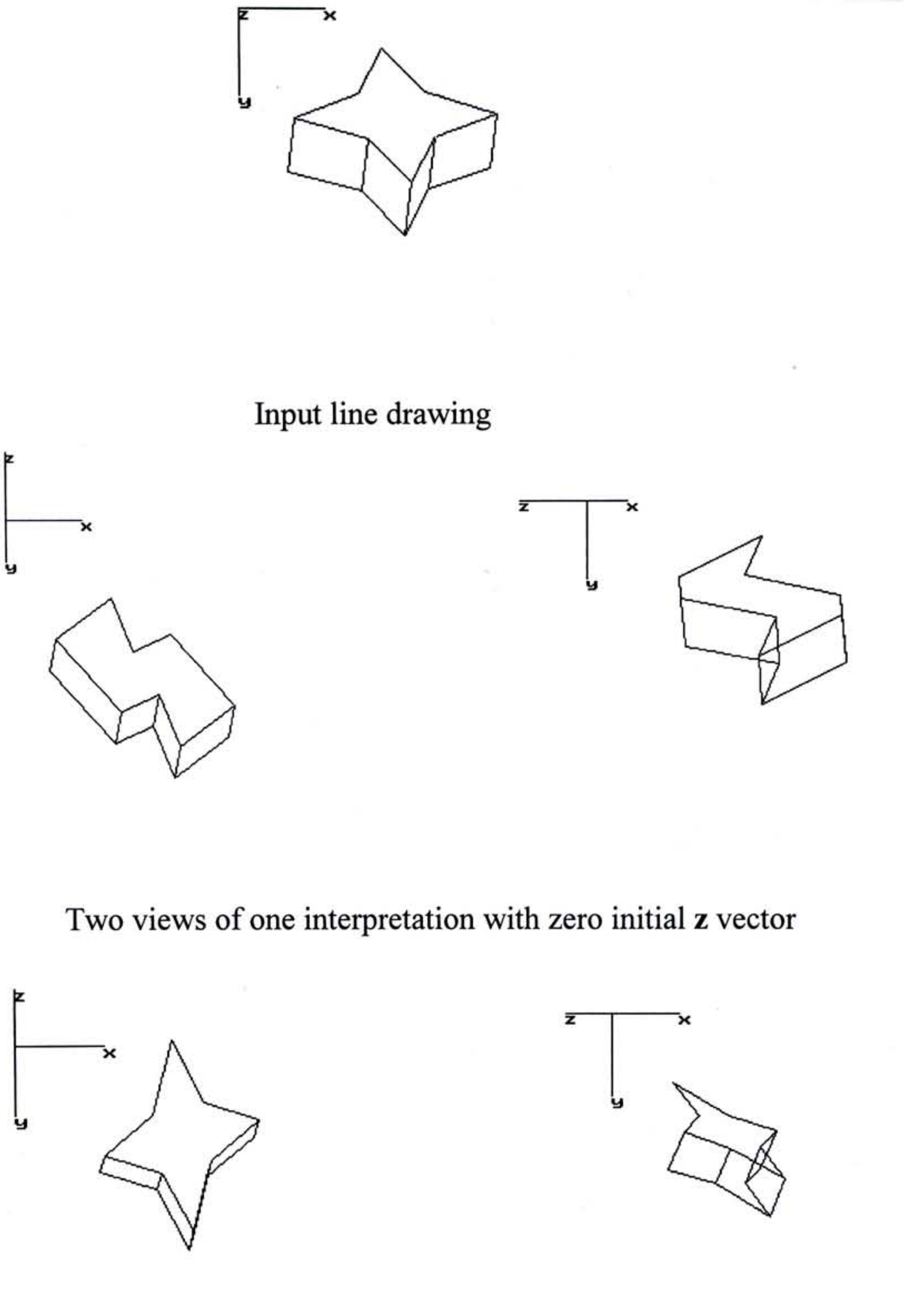


Figure 3.6: Interpretation ambiguity: the line drawing can be interpreted as a box that is being viewed from outside, or as one whose walls are peeled off and whose inside corner is being viewed.

To remove such an ambiguity in our solution and to prefer a shape of bulging outward to the viewer which is more in agreement with human perception, a bias is given to the initial depth values in the optimization procedure of Steps 1 and 3. Zero depth is assigned to all vertices on the outermost boundary of the line drawing (i.e., the smallest polygon enclosing all the line segments), and small depth values to the inside vertices so that they are initially slightly convex to the viewer. If there are no inside vertices, zero initial depth is assigned to all bihedral vertices and small initial depth values to the rest, employing the heuristic that A-vertices (vertices appearing as arrow-shape junctions) and Y-vertices (vertices appearing as star-shape junctions) are more likely to be convex to the viewer than L-vertices (vertices appearing as L-shape junctions). However, it must be emphasized that such initial depth values are merely initial bias to the optimization processes but not the final shape. They may not even be consistent with one another, and in that case the absolute constraints of parallelism and planarity would remove the conflicts.

Two examples are shown in Figures 3.7 and 3.8 to illustrate the effect of the bias. The algorithm in Step 1 is applied to the perfect line drawings. In the middle of each figure two different views of one interpretation are shown, which satisfies all the constraints of parallelism and three-dimensionalness. With an initial bias describe above, the interpretation shown in the bottom is obtained, which agrees more with human perception.

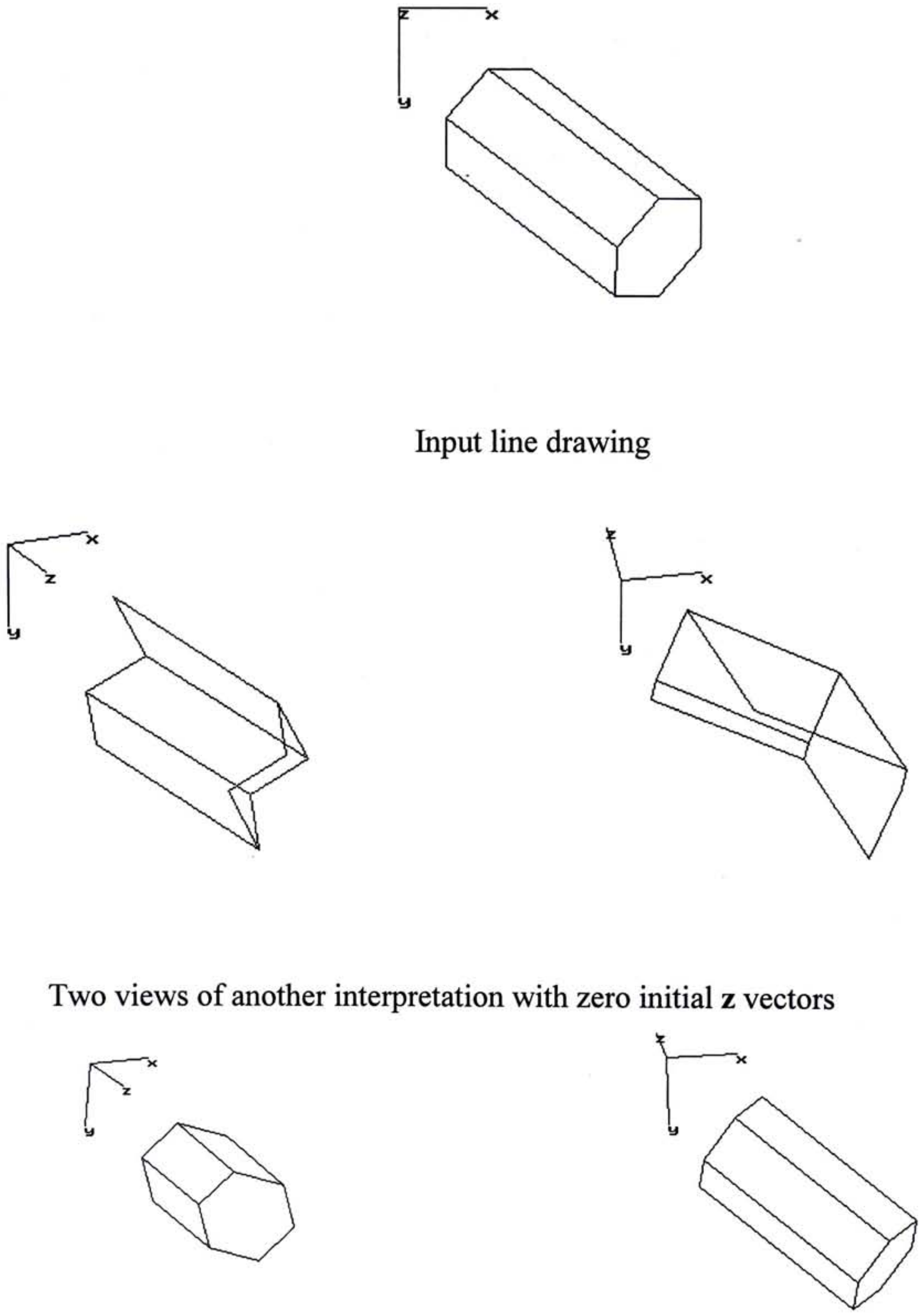


Input line drawing

Two views of one interpretation with zero initial **z** vector

Two views of the interpretation with bias

Figure 3.7: The ambiguity in interpreting the line drawing of a star-shaped object.



Input line drawing

Two views of another interpretation with zero initial **z** vectors

Two views of the interpretation with bias

Figure 3.8: Another example of interpretation with bias

3.4 Nearest-Neighbor Clustering Algorithm

To group a set of patterns $\{x_k\}$ into different clusters C_k's according to a particular inter-pattern distance measure, we can do the following:

- Step 1

  Set $i \leftarrow 1$ and $k \leftarrow 1$. Assign pattern x_i to cluster C_k.

- Step 2

  Set $i \leftarrow i+1$. Find the nearest neighbor of x_i, among the patterns already assigned to clusters. Let d_m denote the distance from x_i to its nearest neighbor which is in cluster C_m

- Step 3

  If $d < t$, then assign x; to C . Otherwise, set $k \leftarrow k + 1$ and assign x_i to a new cluster C_k.

- Step 4

  If every pattern has been assigned to a cluster, stop. Else, go to step 2.

3.5 Discussion

The input line drawing of the presented algorithm is not necessary to be perfect. There can be missing edges due to low intensity gradient between surfaces or extra edges due to surface markings. New line drawings are formed by deleting or adding lines. The algorithm does not consider the case of missing vertices or extra vertices. Actually even human has difficulty in interpreting such line drawings.

One question about the algorithm is: should the iterations pursue an exhaustive and random search. In other words, should it search for all solutions randomly and select the best one? Since there are only finite number of vertices and lines, it is possible to perform an exhaustive search and find out all possible solution. However, for a complicated line drawing, it is very time consuming. The presented algorithm actually performs a reasonable "guess" for forming the new line drawing. The solution is a "good" local minimum. The initial value determines which minimum to be obtained.

It is illustrated in figure 3.9. The initial value (initial shape) here is the one satisfying the hard constraint -- parallelism and the optimal term – 3-Dness. The reason for including the parallelism is that a line parallel to other lines is unlikely to be an extra line. We cannot treat planarity as the initial constrain. It is because we cannot tell whether a closed polygon in an imperfect line drawing is a planar face.

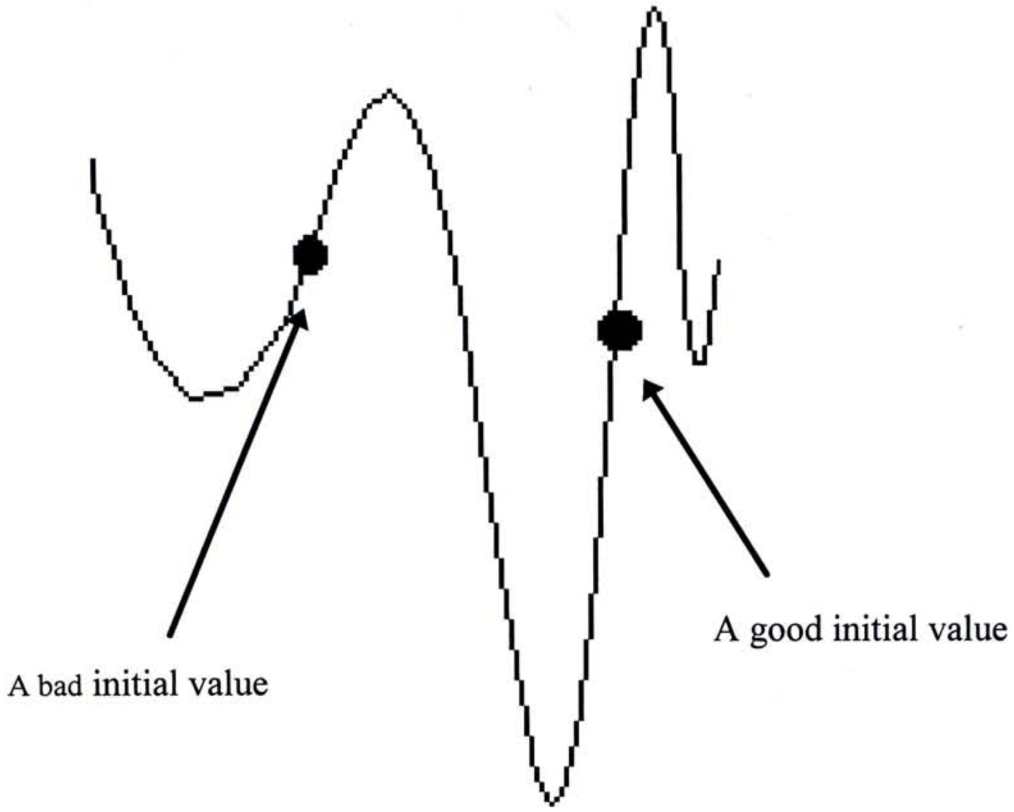


Figure 3.9: An energy map with multiple local minima illustrates the goodness of the initial values.

Chapter 4 Experimental Results

The proposed procedure is applied to a number of different synthetic line drawings with extraneous lines and with some missing boundaries as well as two real line drawings which are obtained from real objects. The method works well with all the examples, while methods like those in [18, 22] would fail as they require perfect line drawings. The procedure is implemented in C language.

4.1 Synthetic Line Drawings

Results on an example used in [18] but with surface markings added and some boundaries removed, are presented in Figures 4.2 to 4.4. The object went through the L-subvertex clustering step twice, and the intermediate line drawings and recovered approximate shapes are shown in Figures 4.2 and 4.3. As the line drawing returned from iteration 2 was the same as that from iteration 1, equilibrium was considered reached and the planarity constraint came in to recover the globally consistent shape. The final interpretation of the line drawing, which is a perfect hexagonal object, is shown in Figure 4.4.

The shaded view of the shape according to the final orientations of the planar faces is also generated and shown. Each surface is treated as lambertian. That means that the intensity is proportional to the cosine of the incident angle.(See Appendix A.2)

Results on another line drawing of an L-shaped block are shown in Figure 4.5 to 4.8. Humans seem to have no difficulty telling the true boundaries and the false boundaries apart, and filling-in the missing boundaries. This time the object went through the L-subvertex clustering step three times to become stable. The final shape recovered, as an L-shaped block, is in agreement with human perception.

One may ask that what if there is no parallelism of lines to exploit in a given line drawing? Would the proposed scheme fail? A few objects with no apparent regularity are tested. Result on one of them is shown in Figure 4.9. Without any parallel symmetry to exploit, the three-dimensionalness and surface planarity constraints would still enforce an output shape of a physically realizable object, which also seems to be in agreement with human perception. However, we restate our position that for an object without any symmetry like a crumpled piece of paper, even humans have difficulty recovering its shape from a single line drawing. Results on a more complicated line drawing with multiple occlusions is shown in Figures 4.10 to and 12. It consists of four objects, some individuals of which are used in [18, 22, 1]. Along the occlusion boundaries there are T-junctions (T-shape junctions), which are where one end of an edge (the stem) meets the middle of another edge (the cap). The stem is an edge of the occluded object, while the cap is an edge of the occluding object. Two depth values are allocated to each T-junction, one for the occluding object and one for the occluded object. Then the iteration steps described above were applied to the line drawing as a whole as if there were a single object in the scene. The line drawing required three passes through the L-subvertex clustering step. All objects are interpreted in agreement with human perception, as shown in Figure 4.12. It should be noted that the hidden lines are not removed in displaying the final shape.

4.2 Real Line Drawing

To test the algorithm on real line drawing, two real images are captured. The first real image is a candy can and the second one is a tape dispenser. Their intermediate results and the final shapes are shown in figures 4.13 to 4.16 and figures 4.17 to 4.20 respectively.

4.2.1 Recovery of real images

To get the line drawings with closed polygons, the real images are first processed by an edge detection and line-fitting software called LINEAR[24]. LINEAR returns a set of line segments according to an adjustable threshold value of the edge strength of the line segments. The edge strength is the intensity gradient across the two adjacent surfaces. A number of corners (vertices) are then extracted at the intersections of the line segments if their endpoints are in close proximity. This process is implemented using the mathematical software MATLAB. Because of weak contrast and image noise, some edges are often missed. The line drawings so obtained therefore is usually imperfect. Nevertheless, such line drawings are input of the iterative algorithm. By the algorithm, missing edges are put back in and extraneous edges are ignored. It should be noted that some edges actually have non-zero curvature, for example, in tape dispenser. However, after the pre-process, the curves are approximated by straight lines. The complete procedure is shown in figure 4.1. Furthermore, one may argue that we can set a low threshold to detect all edges such that there will be no missing boundaries. It is not applicable because a low threshold means many non-boundary lines to be appeared.

In the candy can object (figure 4.13), the processes of edge detection, line fitting and corner detection returned many line segments and corners. However many of the segments were isolated. Using the detected corners, some long segments were connected and at the end closed polygons were extracted. An isolated polygon (a polygon that has no shared edge with any other polygons, i.e., a contour without trihedral vertices) was found and was neglected in the algorithm as discussed in last section. However, some edges were still missing because of the non-ideal process, which could not be avoided. However, the iterative clustering algorithm finally recovered the missing edges and interpreted the imperfect line drawing as a 3-D object which was similar to the original object in the real image. From the recovered 3-D object, it should be notified that the two added edges are not as parallel as in the real images, which is due to the shifting of the corners during corner detection. This effect causes distortion on the recovered object.

Another test image was a tape dispenser (figure 4.17). The interpreting process was similar. However, the nature of the dispenser was different to that of the candy tin. There are two curved edges in the image. Its curvature raises more difficulty. Fortunately, the curvature was not very large and therefore the LINEAR approximated each curve by a straight line. If the curvature was greater, the LINEAR probably approximated the curve by several straight lines. We can also notice that after the pre-processing, the tape cutting part which was very narrow got absorbed because of the too low resolution. Moreover, the inside detail was lost in the closed contour because the image was not a sharp one. At last an outline of the dispenser was extracted but with a major missing edge. This missing edge was also finally restored and a solid object was formed (figure 4.20).

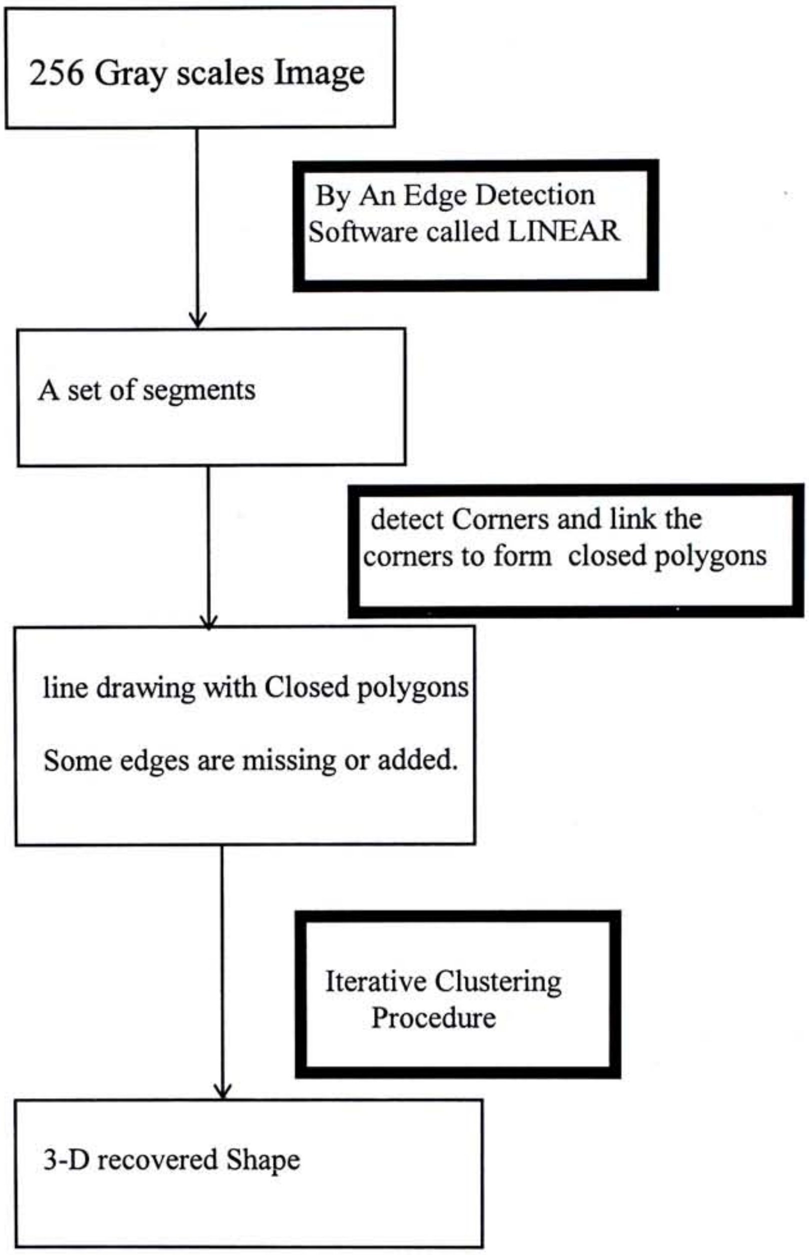
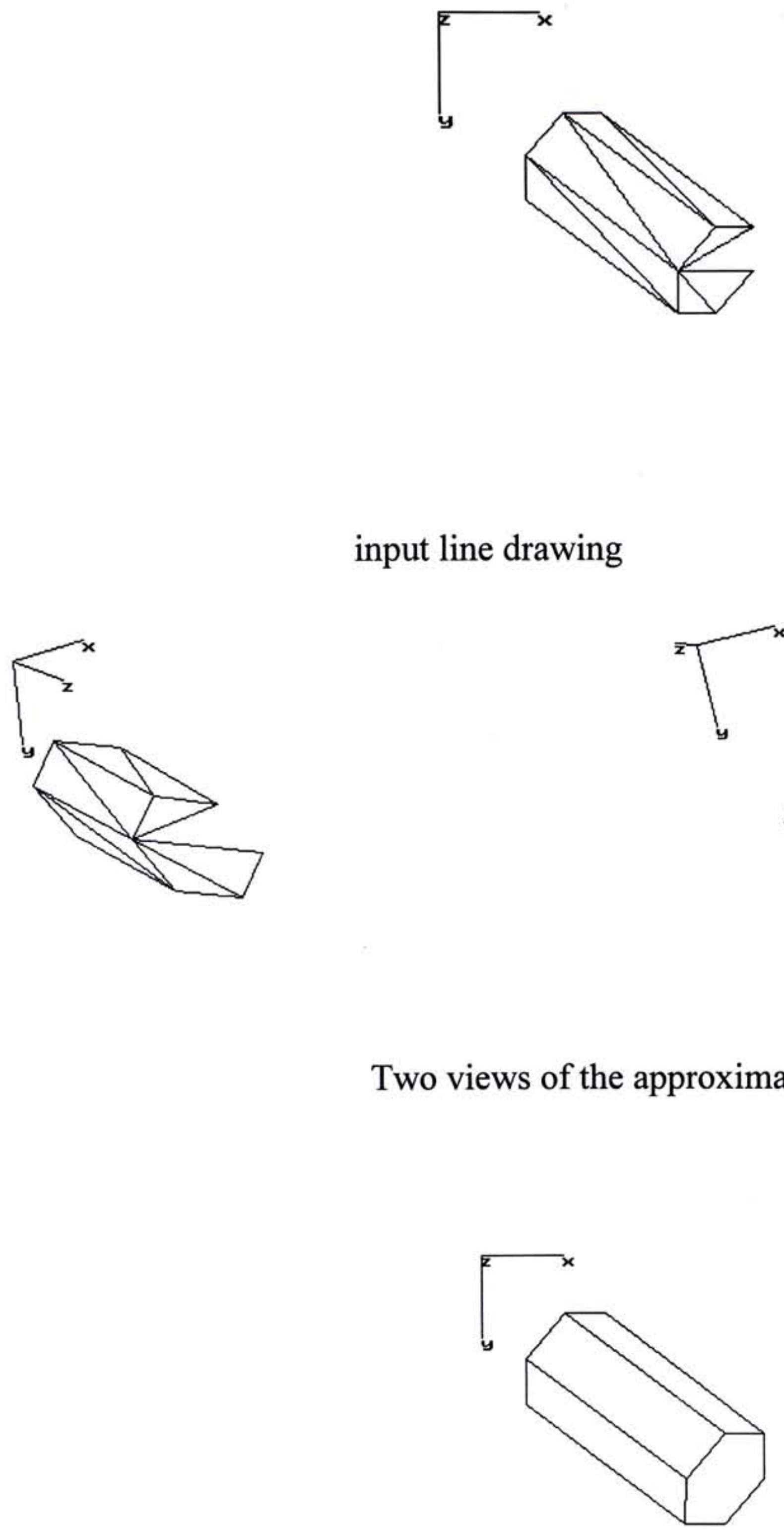

256 Gray scales Image
By An Edge Detection Software called LINEAR
A set of segments
detect Corners and link the corners to form closed polygons
line drawing with Closed polygons
Some edges are missing or added.
Iterative Clustering Procedure
3-D recovered Shape

Figure 4.1: The complete flowchart of recovering a 3-D obtains from a single view 2-D image.

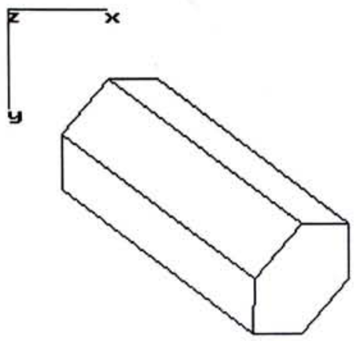


input line drawing

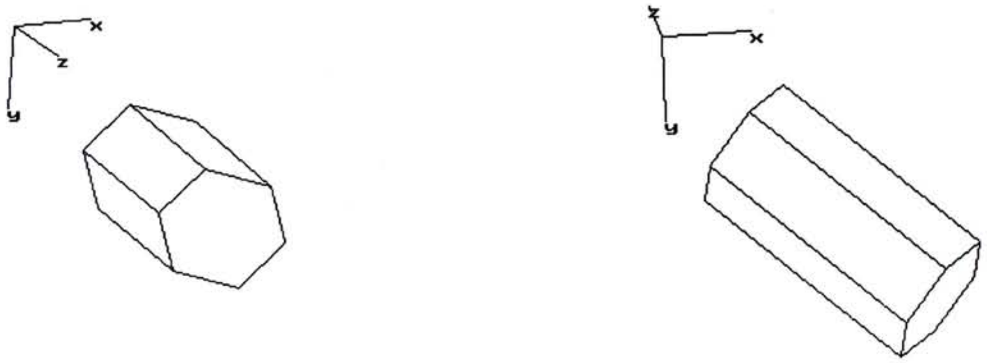
Two views of the approximate shape

new line drawing

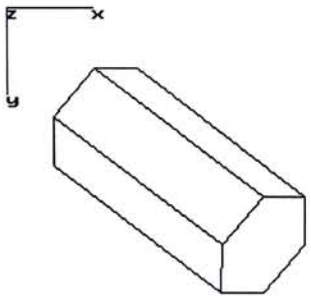
Figure 4.2: Iteration 1 for the line drawing of a hexagonal object.



input line drawing from previous iteration

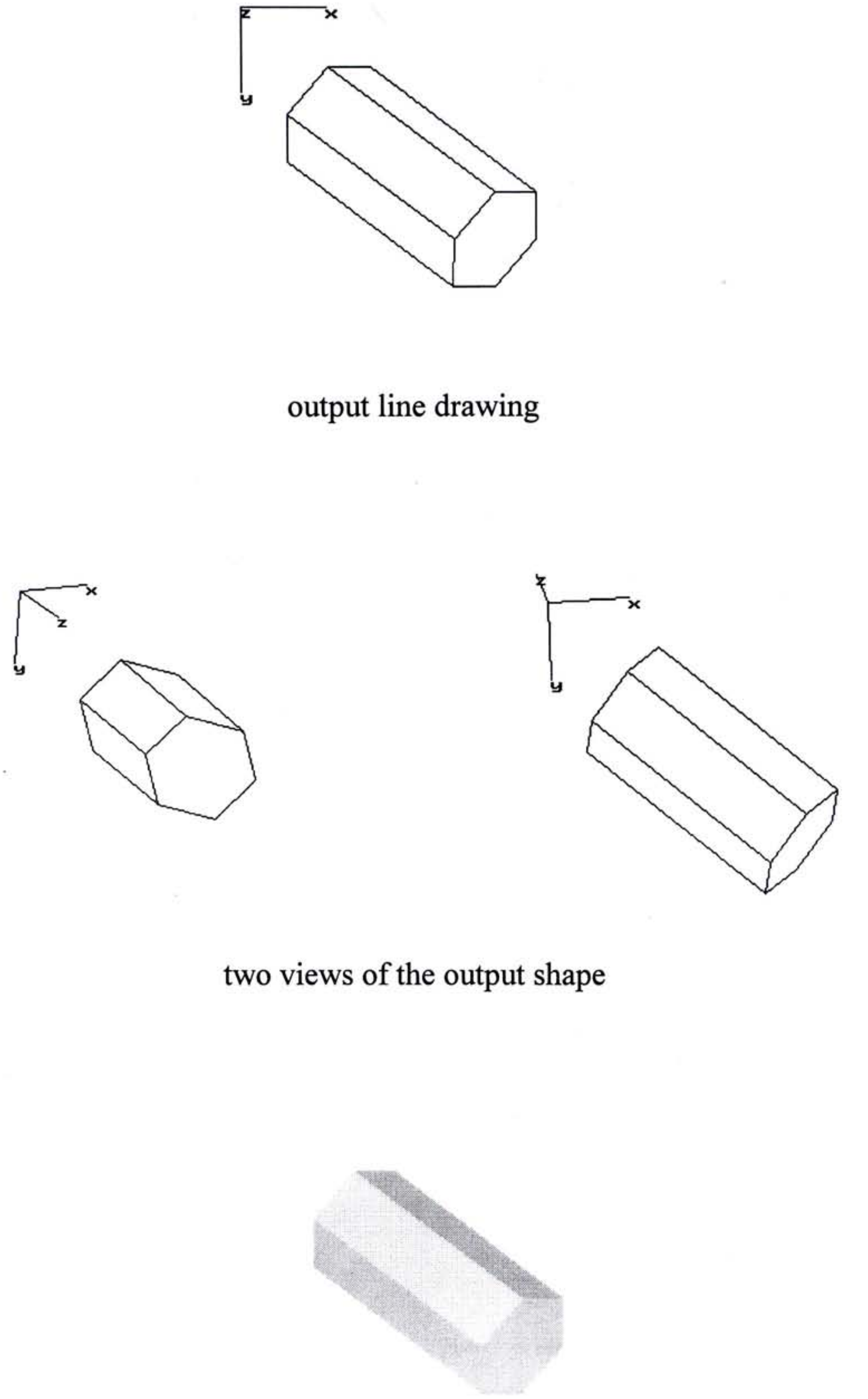


two views of the approximate shape



new line drawing

Figure 4.3: Iteration 2 for the line drawing of a hexagonal object.

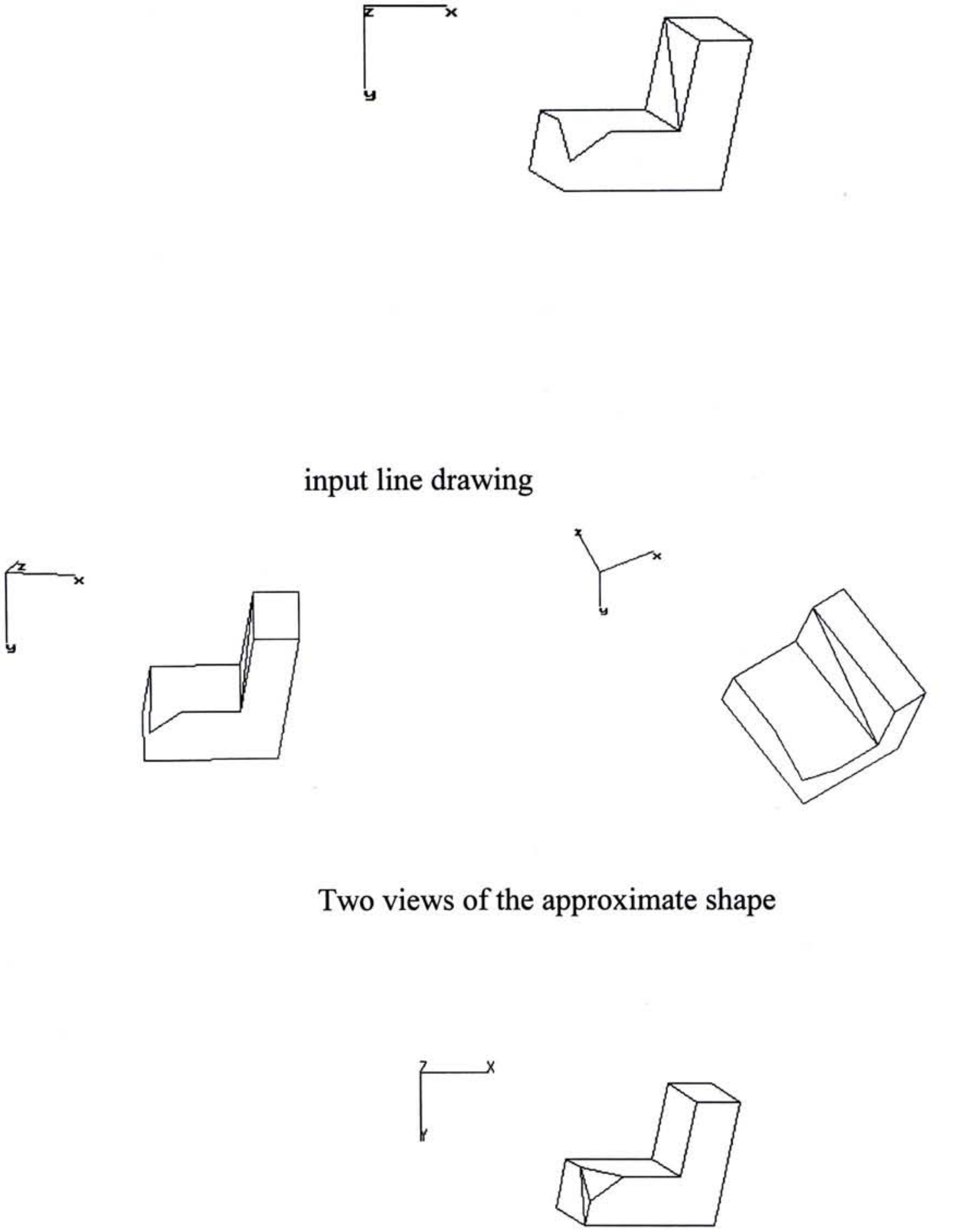


output line drawing

two views of the output shape

shaded view of the output shape

Figure 4.4: Output results for the line drawing of a hexagonal object.

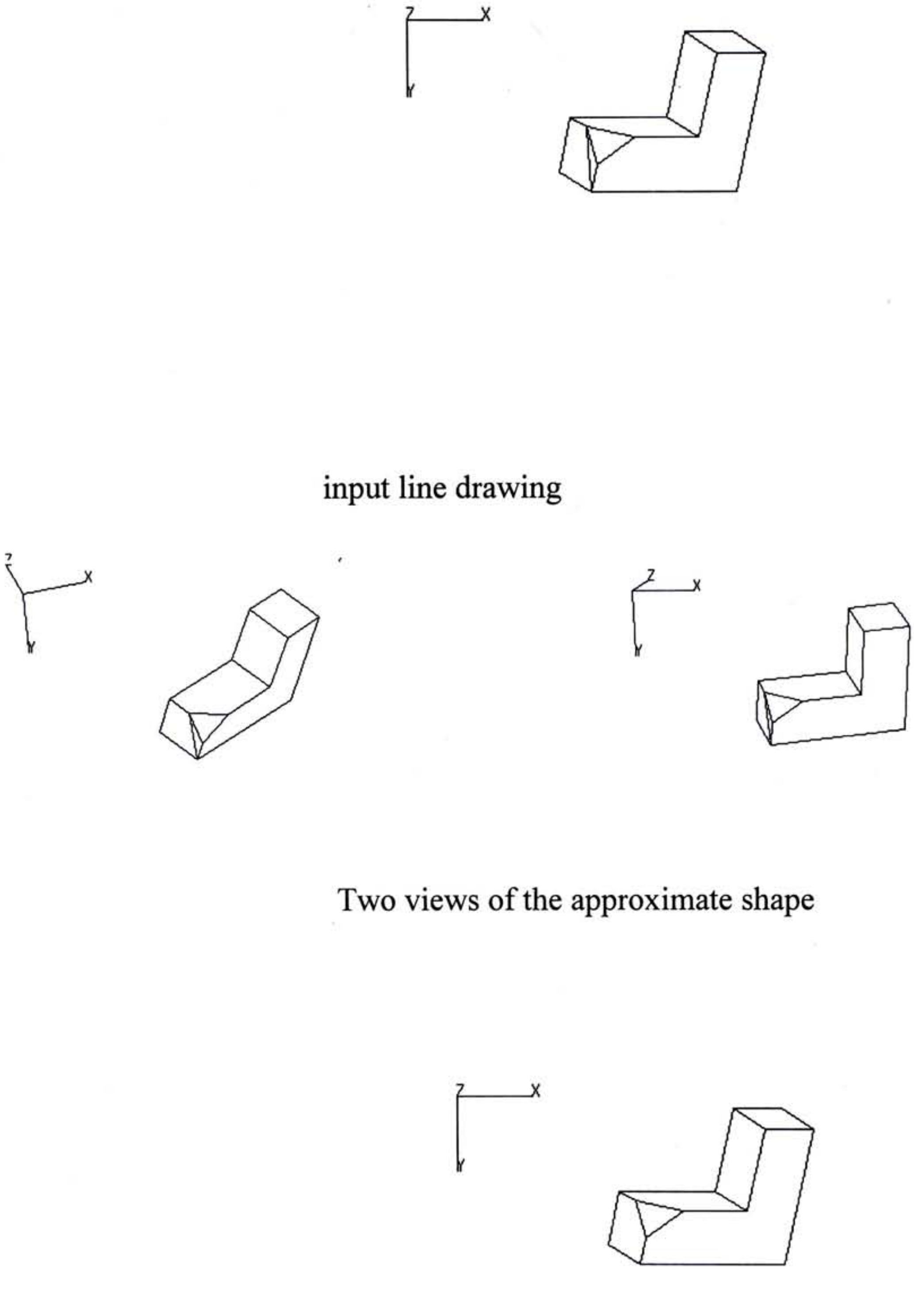


input line drawing

Two views of the approximate shape

new line drawing

Figure 4.5: Iteration 1 for the line drawing of an L-shaped object.

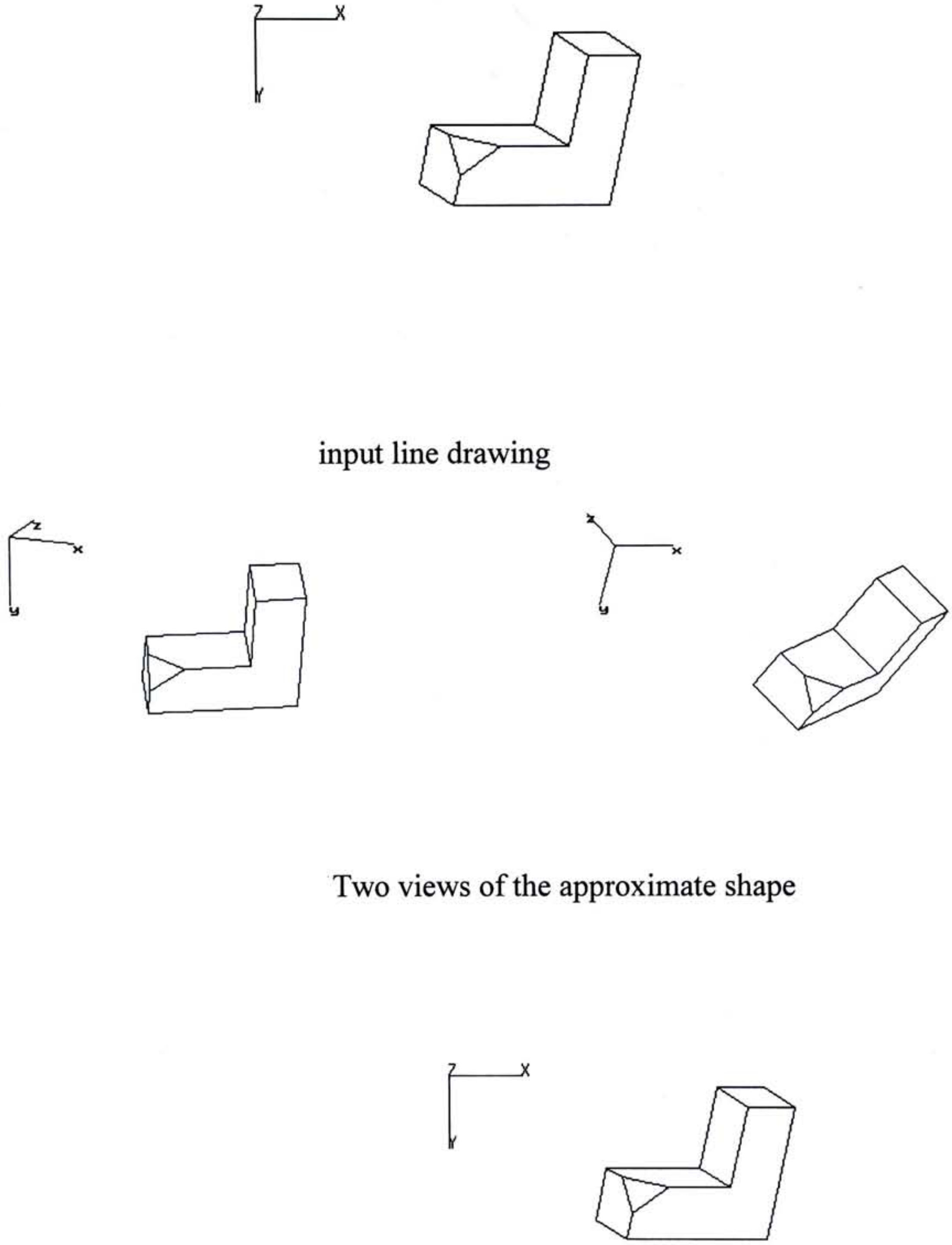


input line drawing

Two views of the approximate shape

new line drawing

Figure 4.6: Iteration 2 for the line drawing of an L-shaped object.

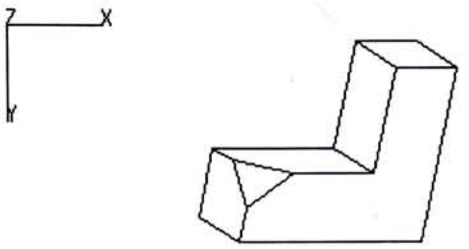


input line drawing

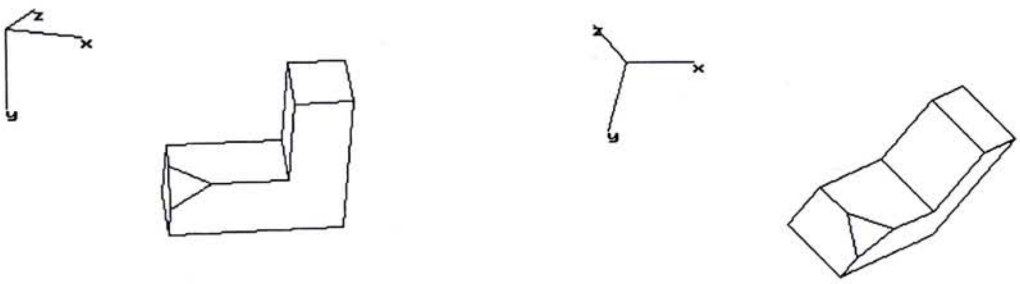
Two views of the approximate shape

new line drawing

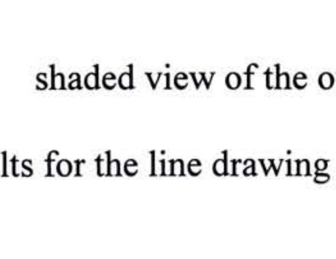
Figure 4.7: Iteration 3 for the line drawing of an L-shaped object.



input line drawing

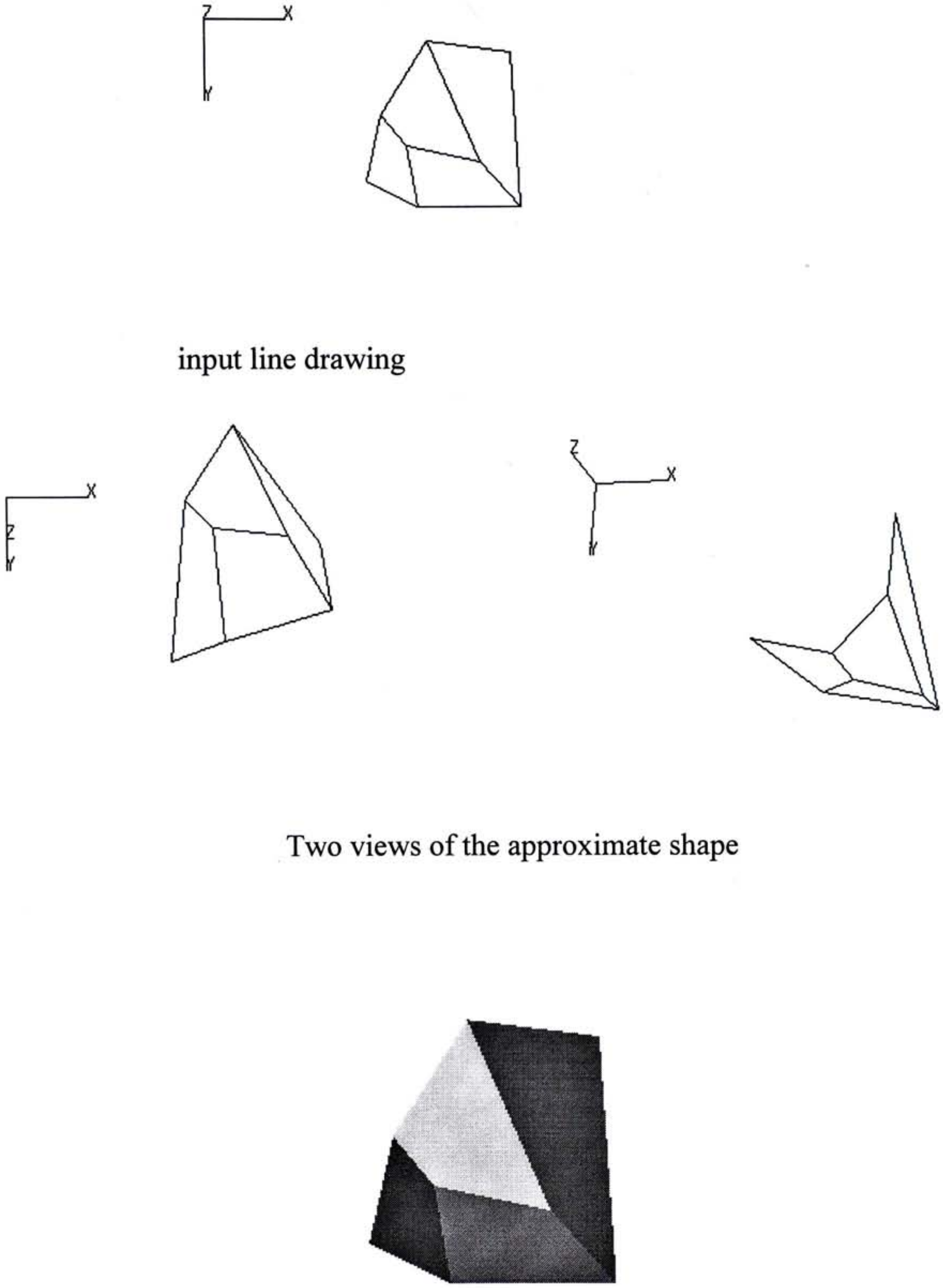


Two views of the approximate shape



shaded view of the output shape

Figure 4.8: Output results for the line drawing of an L-shaped object.

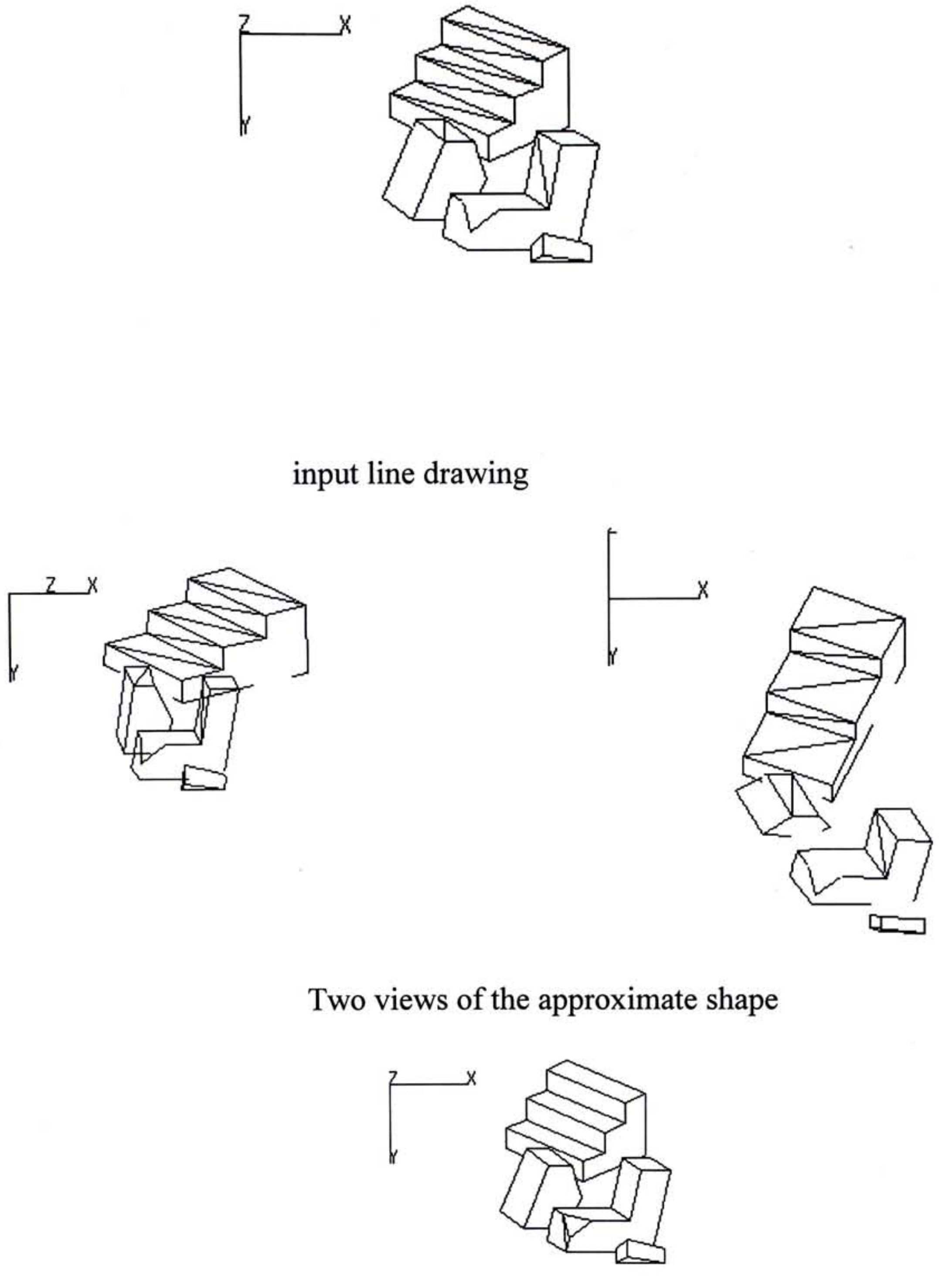


input line drawing

Two views of the approximate shape

Shaded view of output shape

Figure 4.9: Results for the line drawing of an object without any parallel symmetry.

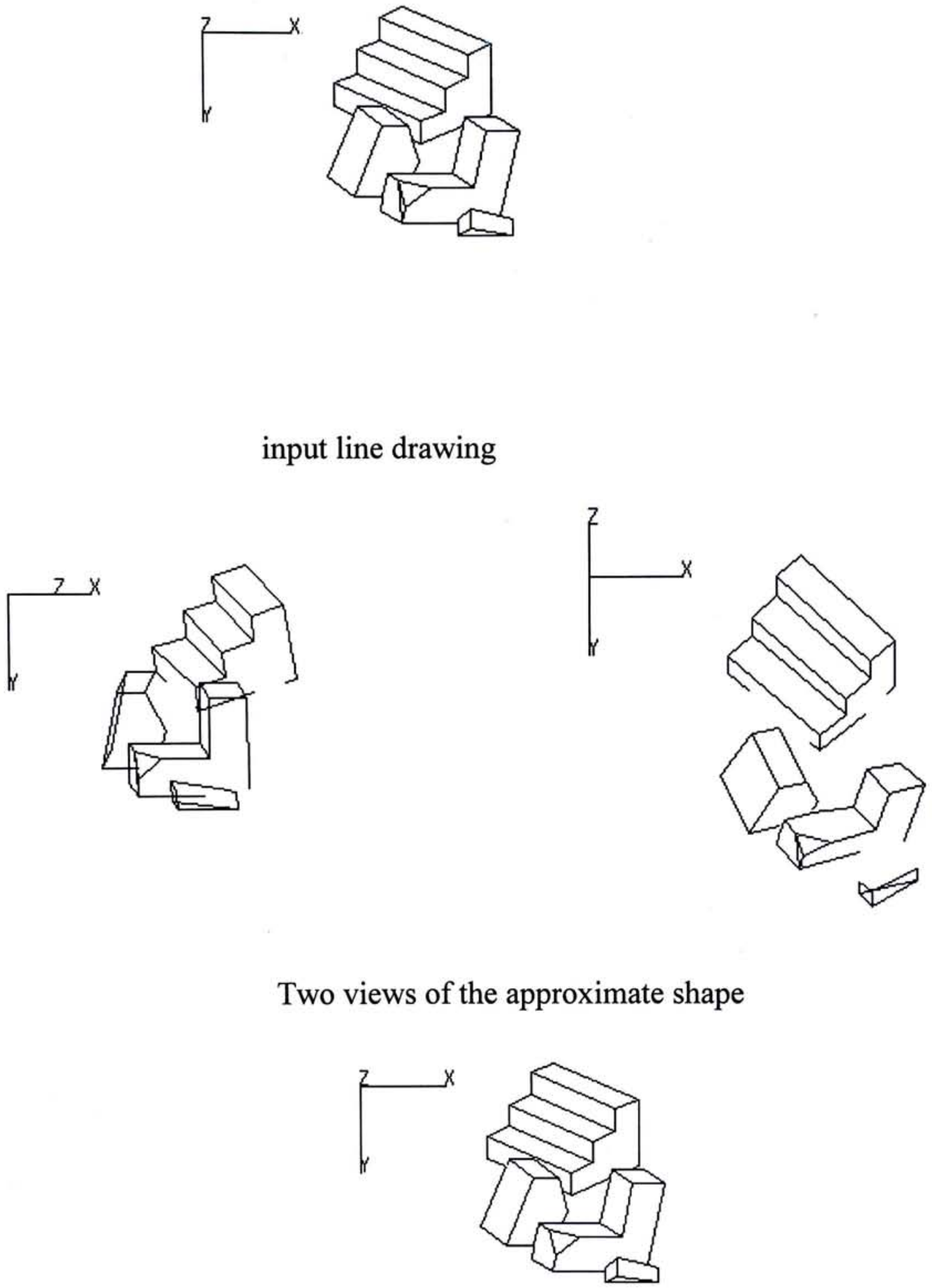


input line drawing

Two views of the approximate shape

new line drawing

Figure 4.10: Iteration 1 for the line drawing of multiple object.

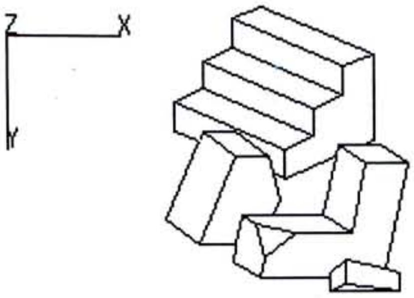


input line drawing

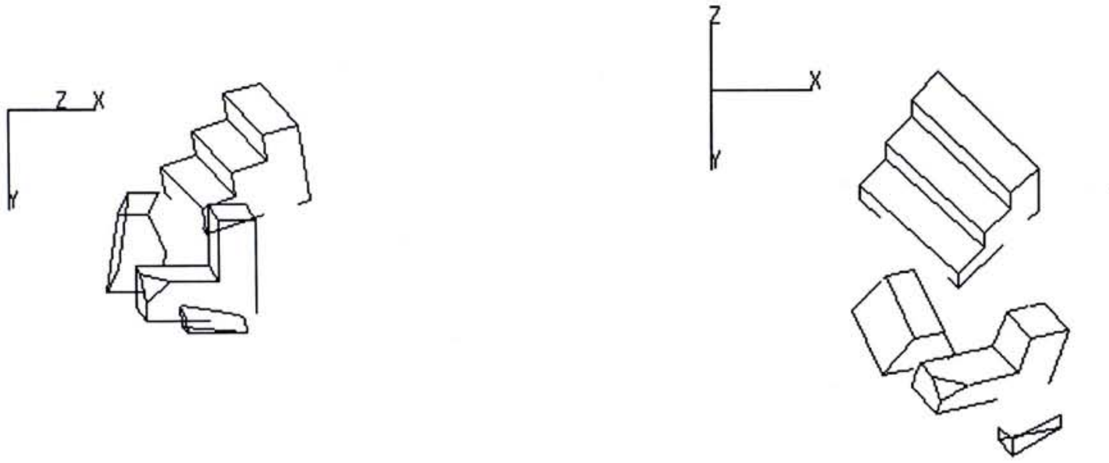
Two views of the approximate shape

new line drawing

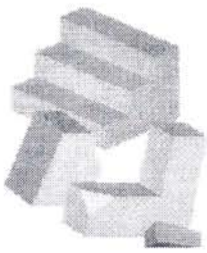
Figure 4.11: Iteration 2 for the line drawing of multiple objects.



input line drawing

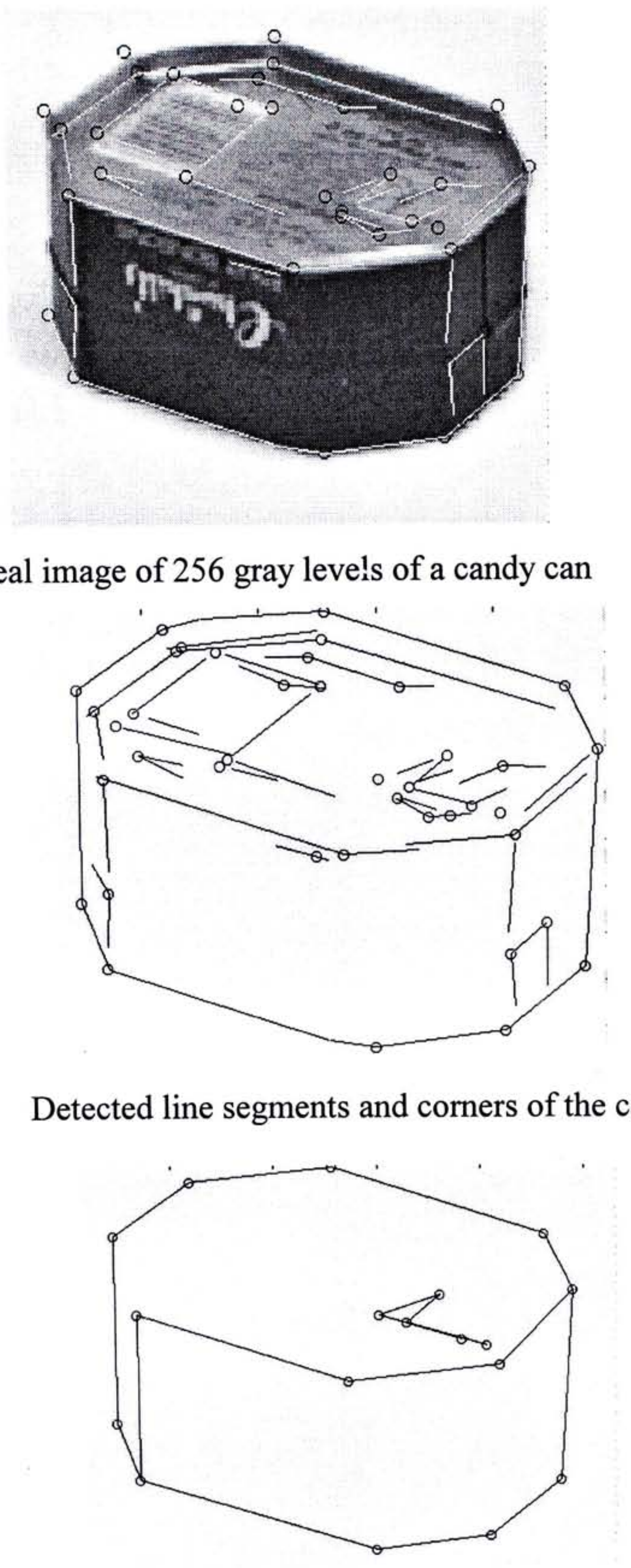


Two views of the approximate shape



new line drawing

Figure 4.12: Final Iteration for the line drawing of multiple object.

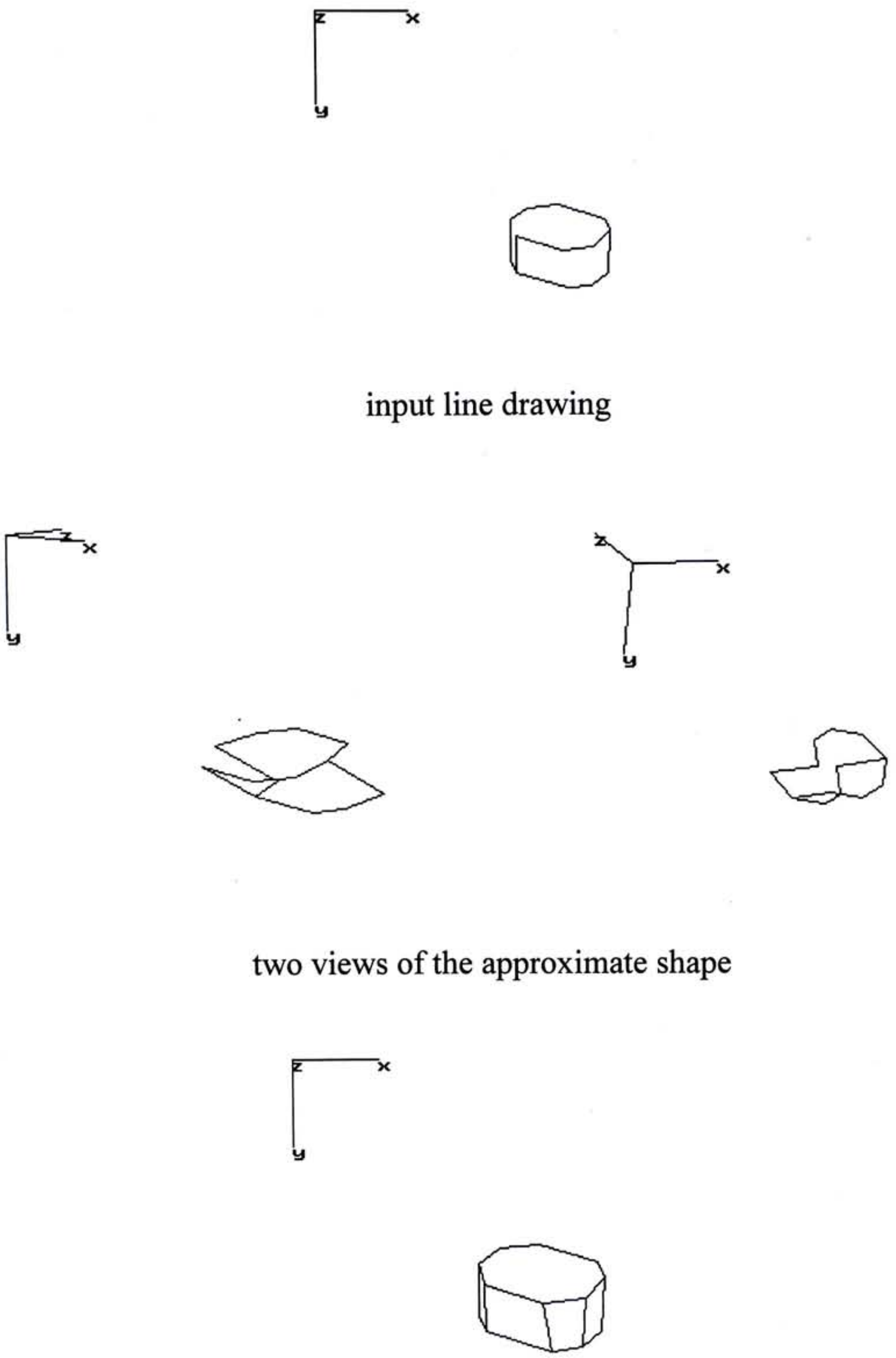


Real image of 256 gray leve!s of a candy can

Detected line segments and corners of the can

Closed contour of the can

Figure 4.13: Edge and Corner detection of the real image of a candy can. A closed contour is finally obtained.

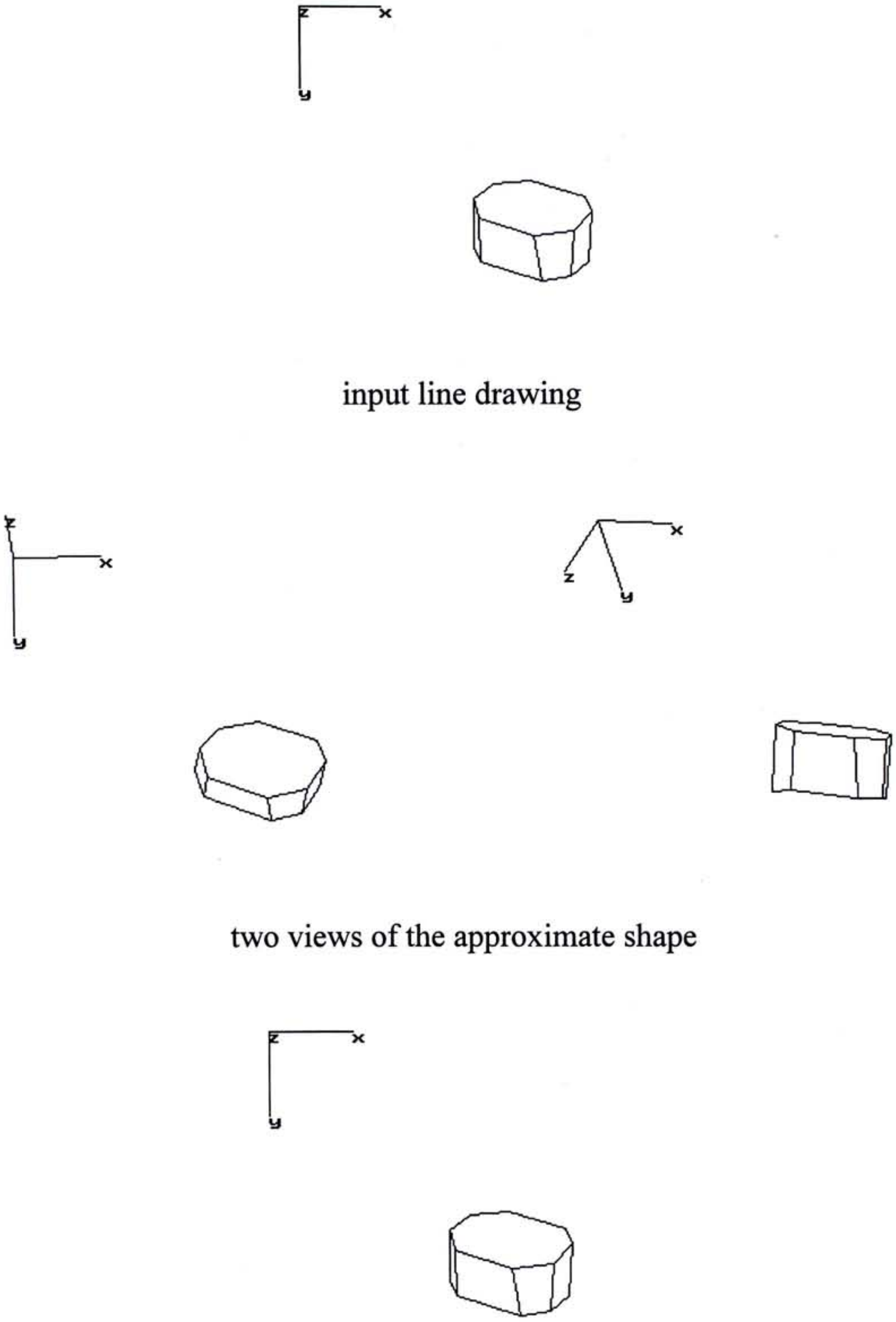


input line drawing

two views of the approximate shape

new line drawing

Figure 4.14: First iteration of the line drawing of the candy can.

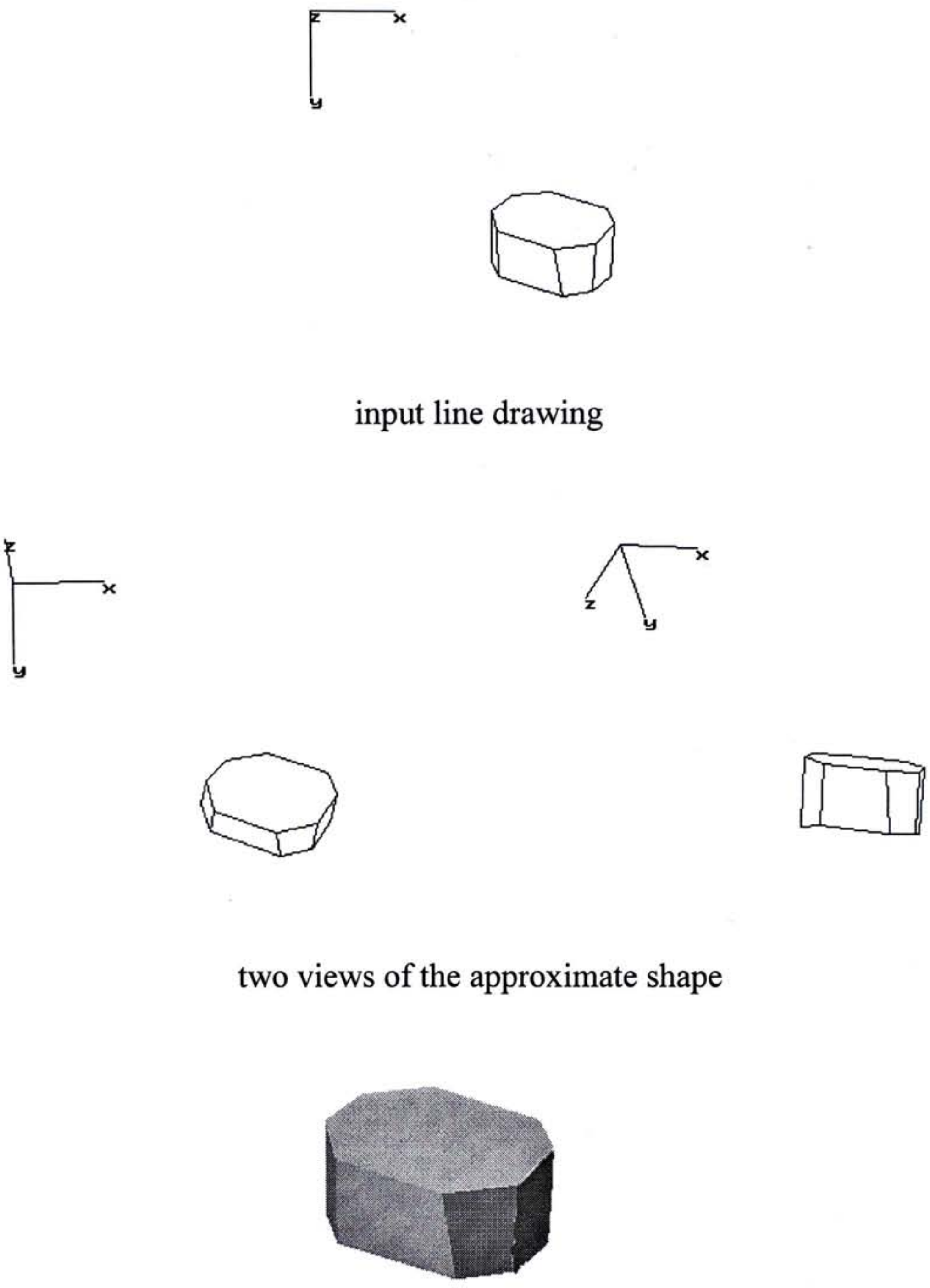


input line drawing

two views of the approximate shape

new line drawing

Figure 4.15: Iteration 2 of the line drawing of the candy can.

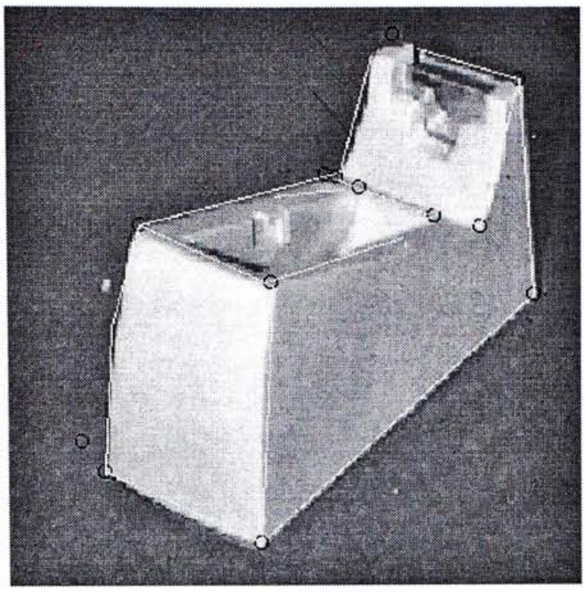


input line drawing

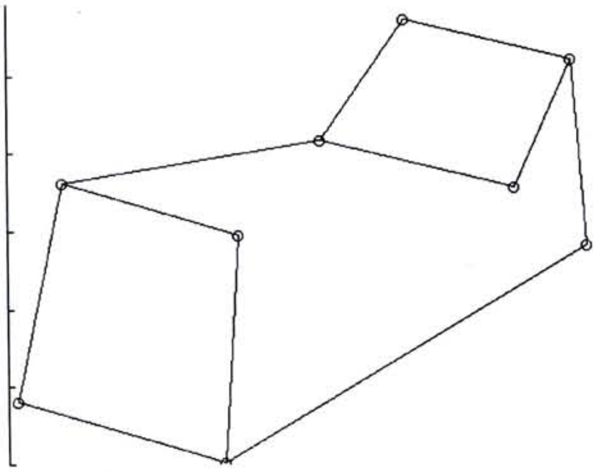
two views of the approximate shape

shaded view of the output shape

Figure 4.16: Output results of the line drawing of the candy can.

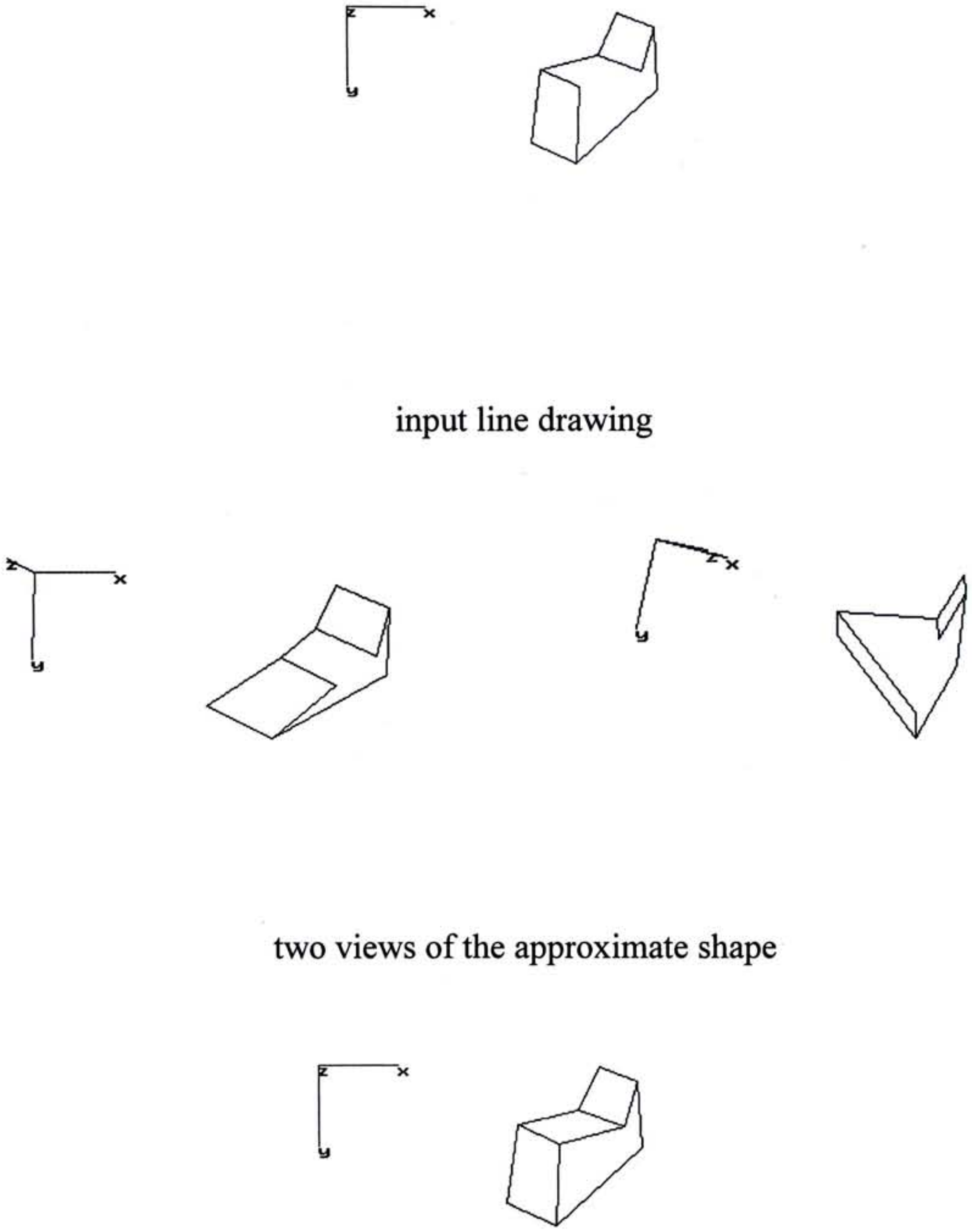


Real image of 256 gray levels of a tape dispenser, and edge segments and corners detected are marked on the image



Closed contour of the box

Figure 4.17: Edge and Corner detection of the real image of a tape dispenser. A closed contour is finally obtained.

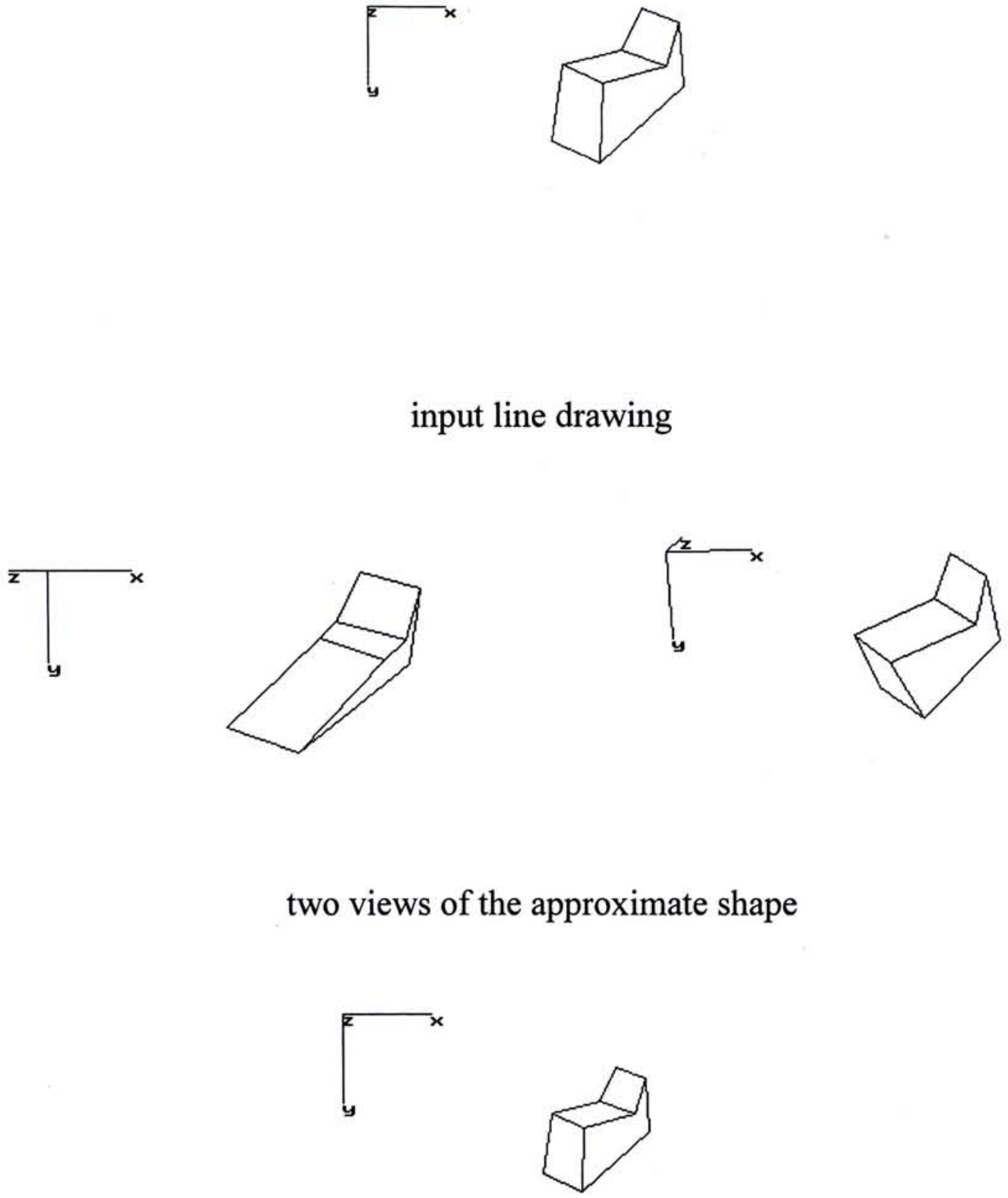


input line drawing

two views of the approximate shape

new line drawing

Figure 4.18 : Iteration 1 of the line drawing of the tape dispenser.

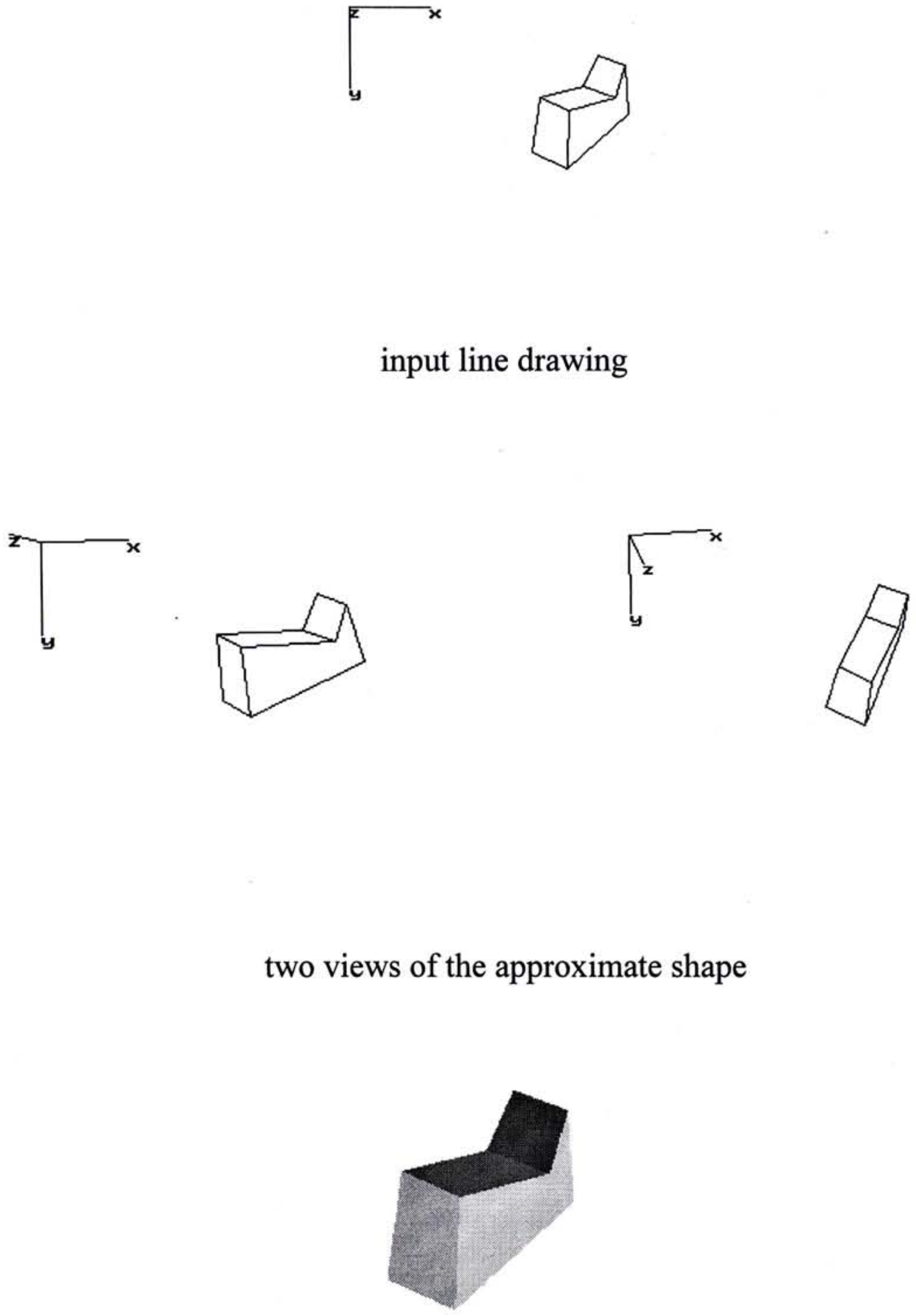


input line drawing

two views of the approximate shape

shaded view of the output shape

Figure 4.19: Iteration 2 of the line drawing of the tape dispenser.



input line drawing

two views of the approximate shape

shaded view of the output shape

Figure 4.20: Output results of the line drawing of the tape dispenser.

Chapter 5 Conclusion and Future Work

Owing to the fact that there is no ideal edge detector existing, line drawings obtained from real images usually do not contain all and only surface boundaries. Some edges may be from the markings or noise, and some important boundary edges may be missing. The previous methods on interpreting a line drawing as the projection of a 3-D object is valid only for perfect line drawings. In this thesis, it is shown how the problem of interpreting an imperfect line drawing can be formulated as a constraint satisfaction problem, and how iterative clustering can help solve it under the representation. The procedure distinguishes the true surface boundaries from surface markings and other extraneous lines, fills-in the missing surface boundaries, and recovers 3-D shapes satisfying constraints of planarity of faces and parallel symmetry of lines, all at the same time. Experiments also show that the 3-D interpretation agrees with human perception for both synthetic and real line drawings. However, it is unrealistic to expect that the algorithm can recover any imperfect line drawing. If a line drawing is too bad because of too bad early visual processing, even humans may have trouble interpreting it. The mechanism proposed in this thesis does require a significant amount of true surface boundaries and a high degree of regularity in the line drawing to work with.

The line drawing tackled here consists of only straight line and the surfaces are planar. An image consisting of curves with non-zero but small curvature can be approximated by several sections of straight lines, as done in the real image experiments. However, for largely curved objects the proposed mechanism will not be adequate. It is because the linear approximation of the curves may not be consistent among the boundaries of the same surface patch, destroying the parallelism clue. The approximation of a surface patch into a number of small planar patches is also less than ideal in terms of shape description. To deal with largely curved objects, further work is needed. The framework of the proposed interpretation mechanism may still be used, but the parallelism clue should be replaced by parallel symmetry, and the planarity constraint by the smoothness constraint.

Appendix A

A.1 Gradient Space Concept

Let Fig. A1 be the geometry involving the viewer, the picture plane, and the object in the scene. The z-axis is taken as parallel with the view line, and the x-y plane is on the picture plane, with the x-axis pointing to the horizontal right. Orthographic projection is assumed here. A plane in the scene whose surface is visible from the viewer can be expressed as

The 2D space made of the ordered pairs (p, q) is called the gradient space G. The 3-D vector (p,q,1) is the vector of the surface normal. When, in general, a surface is represented as

$$-z=f(x,y)\ ,$$

then

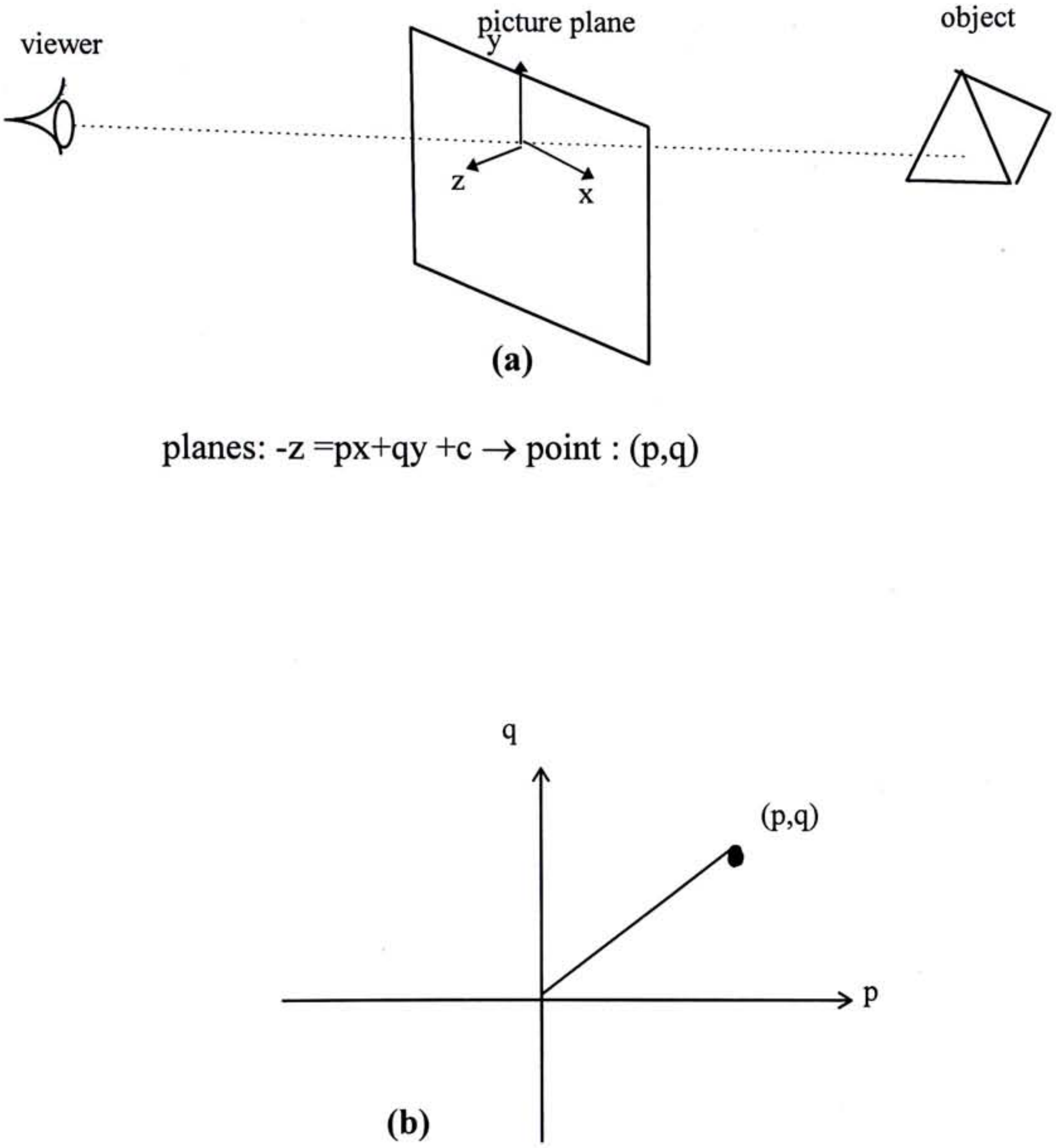


Figure A1: The gradient space (a) geometry including the object, the picture, and the viewer; (b) mapping of planes to a gradient.

A.2 Shading of images

Waltz[31] points out that knowledge of the lighting model can be a valuable aid in determining shape. A simple function represents a common model of surface reflectance. Figure A.2 defines the angles i, e and g used in the function.

(1)

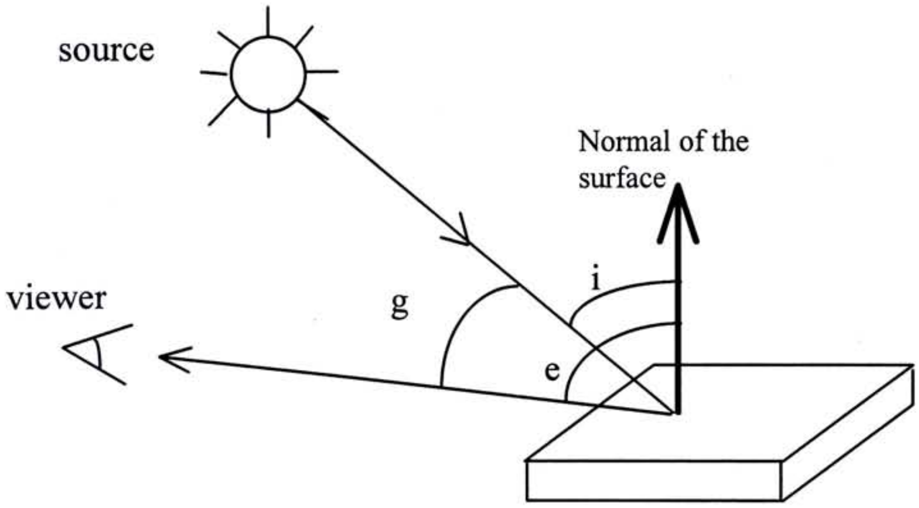


Figure A.2: Defining the angles i, e and g. The incident angle i is the angle between the incident ray and the surface normal. The emergent angle e is the angle between the emergent ray and the surface normal. The phase angle g is the angle between the incident and emergent rays.

The function (1) corresponds to the phenomenological model of a perfectly diffuse (lambertian) surface which appears equally bright from all viewing direction. Here, ρ is a reflectance factor and the cosine of the incident angle accounts for the foreshortening of the surface as seen from the source. This function is used in the recovered shape of the experimental results in the Chapter 4.

Appendix B

Reference List

[1] L. Baird and P. Wang. 3D object perception using gradient descent. *Journal of Mathematical Imaging Vision*, 5:111-117, 1995.

[2] R. Bajcsy and L. Lieberman. Texture gradients as a depth cue. *Computer Grahics and Image Processing* 5:52-67, 1976.

[3] Barrow and J.M. Tenenbaum. Interpreting line drawings as three dimensional surfaces. *Artificial Intelligence,* 17:75 – 116, 1981.

[4] I. Biederman. Recognition by components : A theory of human image understand. *Psychological Review*, 94:115-147, 1987.

[5] D. Boyle and R.C. Thomas. *Computer Vision a First Course,* Blackwell Scientific Publications, 103-119,1988.

[6] Brady and A. Yuille. An extremum principle for shape from contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 6:288 –301, 1984.

[7] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 8(6):679 – 698, November 1986.

[8] R. Chung and K.L. Leung. 3-D interpretation of imperfect line drawings. Proceedings of the Sixth British Machine Vision Conferece, September 1995, Birmingham, U.K..

[9] R. Chung and K.L. Leung. An iterative clustering procedure for interpreting an imperfect line drawing accepted to *International Journal of Pattern Recognition and Artificial Intelligence*.

[10] M. B. Clowes. On seeing things. *Artificial Intelligence,* 2(1):79 – 116, 1971.

[11] J.J. Gibson. The perception of Visual surfaces. *Amer. J. Psychology*, 3:367-384, 1950.

[12] D. Huffman. Impossible objects as nonsense sentences. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6:* 295 – 323. Edinburgh University Press, Edinburgh, 1971.

[13] D. Huffman. Realizable configuration of lines in pictures of polyhedra. *Machine Intelligence* 8. Edinburgh University Press, Edinburgh, 1977.

[14] T. Kanade. Recovery of the three-dimensional shape of an object from a single view. *Artificial Intelligence,* 17:409 – 460, 1981.

[15] T. Kanade. A Theory of Origami World. *Artificial Intelligence*, 279-311, 1979.

[16] H. Kim, W.-c. Choi, and H. Kim. A Hierarchical Approach to Extracting Polygons Based on Perceptual Grouping. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics,* 2402 – 2407, San Antonio, Texas, October 1994.

[17] K. Koffka. *Principles of Gestalt Psychology.* Harcourt Brace, New York, 1935.

[18] Y. G. Leclerc and M. A. Fischler. An optimization-based Approach to the interpretation of single line drawings as 3D wire frames. *International Journal of Computer Vision,* 9(2):113 – 136, 1992.

[19] S. Y. Lu and K. S. Fu. A Sentence-to-Sentence Clustering Procedure for Pattern Analysis. *IEEE Transactions on Systems, Man and Cybernetics,* 8(5):381 – 389, May 1978.

[20] A. Mackworth. Interpreting pictures of polyhedral scenes. *Artificial Intelligence,* 4:121 – 137, 1973.

[21] J. Malik. Interpreting line drawings of curved objects. *International Journal of Computer Vision,* 1:73 – 103, 1987.

[22] T. Marill. Emulating the human interpretation of line-drawings as three-dimensional objects. *International Journal of Computer Vision,* 6(2):147 – 161, 1991.

[23] D. Marr and T. Poggio. A Computational Theory of Human Stereo Vision. *Proceedings of the Royal Society*, B 204:301-328, 1979.

[24] R. Nevatia and K. R. Babu. Linear feature extraction and description. *Computer Graphics and Image Processing,* 13(3):257 – 269, July 1980.

[25] R. Ohlander, K. Price, and R. Reddy. Picture segmentation by a recursive region splitting method. *Computer Graphics and Image Processing,* 8:313 – 333, 1978.

[26] K. Sugihara. Quantitative analysis of line drawings of polyhedral scenes, *Proc. Fourth International Joint Conference on Pattern Recognition*, Kyoto 771-773, 1978.

[27] K. Sugihara. *Machine Interpretation of Line Drawings.* MIT Press, 1986.

[28] S. Ullman. *The interpretation of Visual Motion*, The MIT Press, Cambridge, Massachusetts , 1979.

[29] F. Ulupinar and R. Nevatia. Perception of 3-D Surfaces from 2-D Contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 15(1):3 – 18, January 1993.

[30] F. Ulupinar and R. Nevatia. Shape from Contour: Straight Homogeneous Generalized Cylinders and Constant Cross Section Generalized Cylinders. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 17(2):120 – 135, February 1995.

[31] D. Waltz. Understanding Line Drawings of Scenes with Shadows. *The Psychology of Computer Vision*, Mcgraw-Hill, New York,19-91, 1975.

[32] A.P. Witkin. Recovering Surface Shape and Orientation from Texture. *Computer Vision*, North-Holland, 17-45, 1981.