The Nottingham Trent University

INTERPRETATION OF SCANNED ENGINEERING DRAWINGS - A HIGH LEVEL APPROACH

SABAH MAHDI RAZZAQ

A thesis submitted to The Nottingham Trent University
In partial fulfilment of the requirements for the
degree of Doctor of Philosophy

THE NOTTINGHAM TRENT UNIVERSITY 1997

ABSTRACT

An engineering drawing is a description of a mechanical object which incorporates geometric and technical details as well as information concerning the manufacturing process. In the mechanical engineering domain, where the exact shape of an object is generally an essential aspect of the total product specification, a large part of the drawing represents the 3-D geometry. The drawing usually consists of three orthographic views, together with dimensions and other textual information.

Initial processing begins with scanning of the engineering drawing to produce a binary image that can be stored in a computer. Errors and distortions caused by the scanning process mean that it is not always possible to construct a complete 3-D model using only the low level geometric information. Additionally, simplifications made to the drawing further complicate the process, because some information may only be provided by textual annotations on the drawing. Therefore, it is necessary to interpret textual information to recover simplified features and correct errors and distortions before the reconstruction process takes place.

Reconstruction algorithms, building on previous work of [Shaw95] and others, have been implemented for recognising low level features on a scanned engineering drawing. The algorithm initially identifies and separates the orthographic views. The geometric entities within each view are then recognised, collinear lines are grouped and identified as visible, hidden or centre lines. The identified lines are associated with type identifiers.

High level algorithms have been developed to understand the technical details on the drawing. Annotations are analysed and associated with corresponding geometric representations. Simplifications on the drawings are made explicit for the bottom-up procedure whilst distorted features are recognised and corrected using the information provided by the text. This approach of understanding technical details attempts to model an aspect of human drawing interpretation, whereby an 'envelope of expectation' is developed.

ACKNOWLEDGMENTS

My sincere appreciation to Dr. P. D. Thomas, for his supervision, the many valuable suggestions and encouragement through out this research. Immense gratitude go to Dr. J. F. Poliakoff for the supervision, the many valuable suggestions and the many valuable discussion sessions.

Many thanks to Prof. R. J. Whitrow, for giving me the opportunity to do this research and his support and valuable advice at the start of the research. I also would like to thank all my colleagues in the Microprocessor Centre for their help and friendly support. Special thanks to my good friend Taha Osman for all the support and encouragement and for being a great comrade.

I greatly appreciated the opportunity given to me by the Department of Computing to attend the Workshop and for the organisers to let me participate in presenting the paper. The Workshop took place on 9-11 August 1995 at the Pennsylvania State University, Pennsylvania, USA. Appendix B includes the full paper presented at the conference.

Finally, my love to Debbie and Samantha.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

Annotation - A piece of textual information on an engineering drawings.

Arc - A circular path connecting two nodes on a drawing.

Assembly Drawing - A drawing which consists of a number of sub-assembly drawings of all the parts required.

Auxiliary view - Additional view which shows the true shape of a surface.

B-rep - Boundary Representation is a 3-D representation scheme which represents the faces and surfaces of the object together with the edges forming the boundaries of the faces and the vertices forming the ends of the edges.

Boundary/Contour - A closed loop of connected lines that does not cross itself.

Centre Line - A line used to indicate a centre of a feature on the drawing in a particular projection. It is shown with a Dot-Dashed line.

Collinear lines - Continuous straight broken segments which have a common gradient and directed along a single path.

CSG - Constructive Solid Geometry is a 3-D representation scheme which uses a number of parametrised primitive solids such as polyhedral, cylinders, cones, spheres and tori. The primitive solids can be combined together using the set operators Union, Intersection and Difference to construct more complex shapes.

Dimension line - A line which defines part of the object associated with a dimension text.

Dimension text - A phrase which defines the size or specifies the relative positions of various elements of the object.

Feature - A simple 2-D geometric shape defining a particular part of the object.

Hidden line - Part of the outline of the object which is not actually visible in the given projection, because it is behind another part of the object. It is shown with broken line.

Leader Line - Line used to associate a piece of text with the part of the object to which it refers.

OCR - Optical Character Recognition Techniques.

Orthographic Projection - A method of producing a number of separate 2-D inter-related views which are at right angles to each other. It is usually a multi-view projection.

Phrase - A sequence of tokens which may be an incomplete sentence.

Prepositional Phrase - A sequence of tokens including a preposition which constitute the phrase.

Principal Projections - The main projections of the drawing, namely Front, Plan and Side projections.

Relational Template - A structure which combines a number of semantic templates.

Sectional view - Additional view which shows the interior detail of the object.

Semantic Template - A structure which holds explicit information about features or other geometric information.

Silhouette Node - A point which lies on an arc where it is parallel to an axis of the drawing.

Simple Phrase - A phrase which does not include a preposition.

Standard Node - A point where two or more lines meet on a drawing.

Standard Line - A straight line ending at nodes on a drawing.

Syntactic Net - A structure with inter-related syntactic categories.

Syntactic Type - A single letter code associated with a category of token.

Phrasal Type - A pattern of syntactic types associated with a phrase.

Tangency Node - A point which lies on a circular arc where two arcs or an arc and a line meet and the gradient is continuous.

Token - An individual word or number, where a sequence of tokens constitute a phrase.

Visible Outline - Part of the outline of the object which is visible in the given projection. It is shown with a solid line.

Chapter 1

Introduction

1.1 Engineering Drawing

The engineering drawing is an important element in manufacturing, production, construction, etc. It is the description of a technical object incorporating geometrical and technical details. It shows the shape of the object, specifies the size, and describes important aspects by annotations. It is also the medium to present the design ideas, to store information, and to communicate with other people. The principle of the engineering drawing is based on orthographic projection. A standard three-view is generally used as shown in Figure 1.1, but to avoid ambiguity or to simplify complex parts of the object on the drawing, more than three views may be needed. Sometimes two views or even one view will be sufficient. Also, cross sections are frequently used to show the internal structure and complicated portion of an object. There are textual information, symbols and conventions in the drawing which condense the geometrical information and highlight the contents.

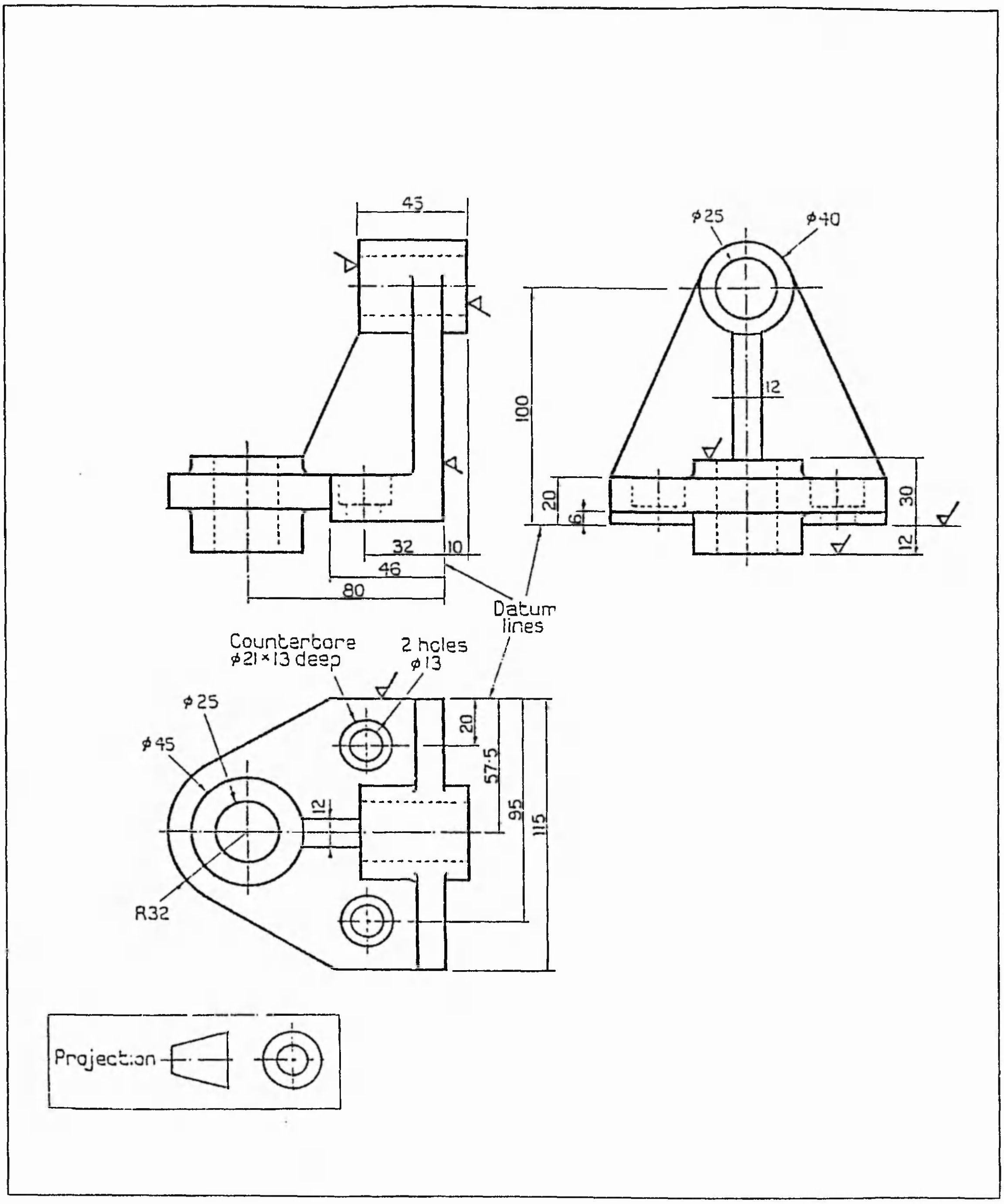


Figure 1.1: Example of Engineering Drawing (taken from [Hewitt75]).

Using a CAD system, an engineering drawing can be easily and accurately created with standardised format. This makes it easier to store, transfer and duplicate. While having the advantages of CAD system, there are some drawbacks too. As a 2-D representation of a 3-D objects, the outlines in an engineering drawing are not unique and very often are not complete [Chou88]. This means that the 3-D representation may be ambiguous if the interpretation relies only on the geometric information. Also the same object can be represented by several different engineering drawings and an engineering drawing can represent more than one object [Chou88]. But the purpose of the complete drawing is to identify an object precisely using both outlines and textual information.

Usually, a well trained person is needed to interpret an engineering drawing and understand the geometric information. However, as computer systems started to take over this task and in some cases where there are ambiguities, the interpretation needed to look for additional information to solve this problem. This thesis therefore, is mainly concerned with the interpretation of the textual information on the drawings.

1.2 Purpose of Engineering Drawings

Throughout the ages, people have found communication with each other to be essential to their development. The means they have used has progressed from grunts to articulate speech and from signs and primitive drawings to competent writing and complicated drawings. All these have served to convey ideas, information and instructions from one person to another.

In present day industry, the *engineering drawing* is a very important means of communication and is part of the international language of engineering. Engineering drawings are a system of communication in which ideas can be expressed precisely, collective information on the drawing can be conveyed completely and unambiguously, and even the most complicated shapes can be specifically described. However, human intelligence is often required to rule out ambiguous interpretations.

The international conventions of engineering drawing are published by the British Standards Institution [BSI] and others. These standards enable the draughtsman to understand clearly the designer's ideas and instructions and the craftsman to interpret an engineering drawing precisely for manufacturing or assembly purposes.

1.3 Reasons for Processing Engineering Drawings

The past two decades has seen an increase in the capability of the computer to perform cognitive tasks usually done by humans, such as chess playing, understanding natural language or interpreting pictures. With today's still expanding technology, computer systems have taken over many more tasks that used to be performed by humans.

For years researchers focused on linking different manufacturing processes via computers and expert systems [Meeran93]. At the same time, the major change in the design process was to use CAD instead of pen and paper [Filipski92]. Because of the existence of vast numbers of old engineering drawings, there is a need to convert engineering drawings into a suitable format that could be stored in a computer and manipulated by a CAD system [Joseph92, Banks89, Elliman90]. It would be of great benefit to many designers and engineers, if a system could automatically transfer the vast number of existing engineering drawings into data suitable for a CAD system.

Engineering drawings stored in a computer have many advantages over paper drawings. It is easier to store and access information in a computer than to search for a paper drawing. Computerised drawings can be shrunk or expanded to give a more detailed view of a part. Information stored on a computer can be modified more easily than paper drawings, and the consistency of changes can be maintained across different views automatically.

The simplest way to store a drawing is to represent it as a bitmap image, but this has the disadvantage that a very large amount of data needs to be stored to represent it with any accuracy. It is also inconvenient to have to modify a bitmap image at the pixel level. A better method is to search for lines, arcs and text in the bitmap image and to create and store data structures representing each of the features found. This considerably reduces the quantity of information that needs to be stored and allows the drawing to be modified more easily.

While a 2-D computer representation of an object may be an improvement over a paper drawing, a consistent 3-D model of an object has even greater advantages. Graphical techniques can be used to view a 3-D model from any direction, and can perform hidden line removal. Shading and texturing give a much more natural representation of an object. A properly integrated 3-D modeling system therefore has the potential to considerably simplify the design and development of a product and to allow rapid modifications to be made, Figure 1.2.

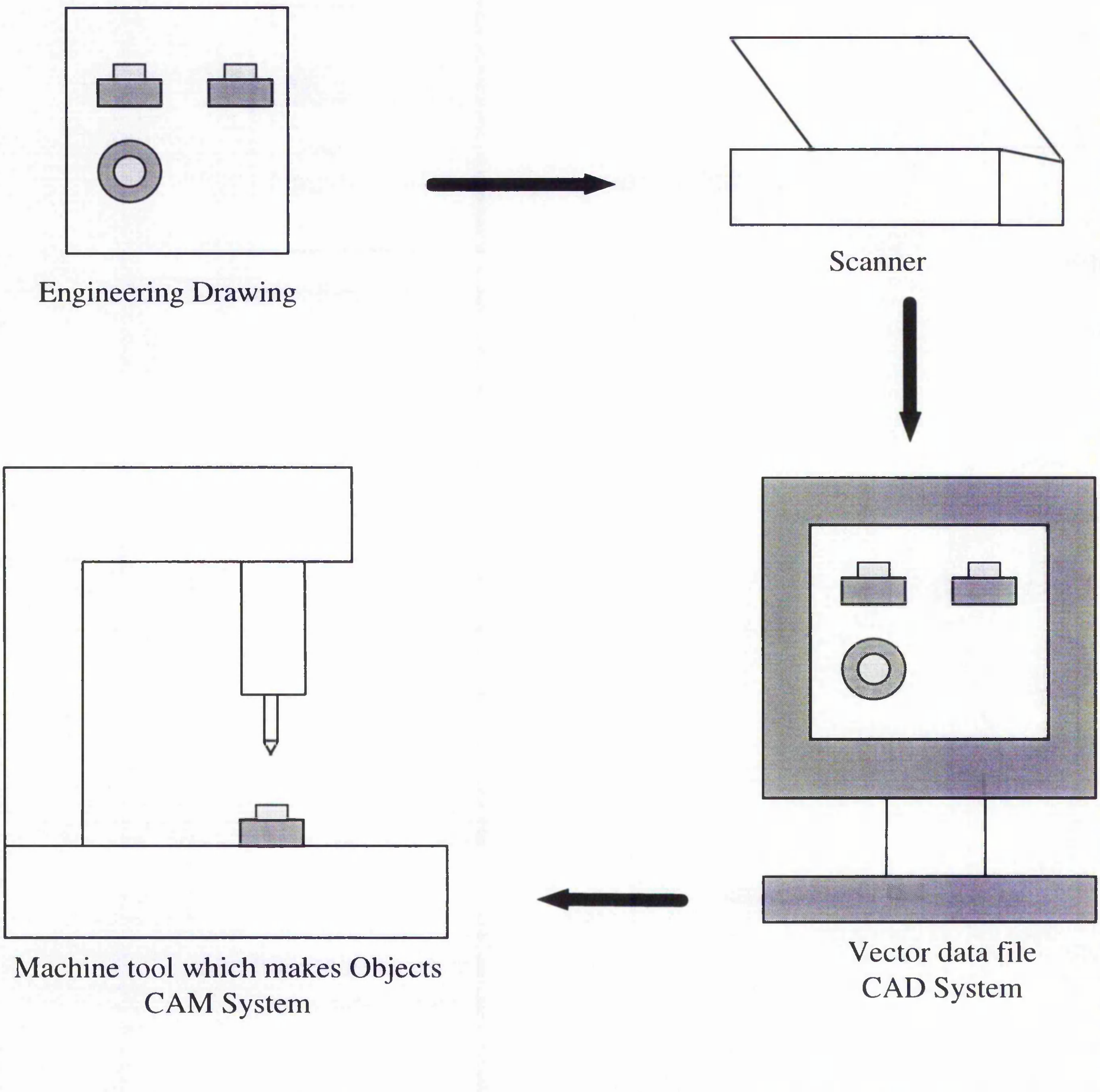


Figure 1.2: Illustration of an integrated manufacturing system utilising automatic interpretation of Engineering Drawings.

1.4 Justification for this Work

Until the late 80's, the problem of reconstruction of a 3-D object was largely considered as a geometric problem [Wesley80, Wesley81], in which line drawings were represented in terms of symbolic vertices and lines forming the orthographic projections of the objects' wireframe. The exact 2-D projection co-ordinates of the vertices and line-endpoints were assumed to be known. It was further assumed that the input had no noise: no lines or vertices can be missing, and no extraneous vertices or lines were present. Many algorithms were published [Aldefeld83a], [Aldefeld83b], [Aldefeld84], [Er83], [Preiss81], [Preiss84] to solve this problem for objects of varied complexity. Typically, these algorithms exploit geometrical constraints to prune the search space of possible 3-D objects.

More recently, people have begun to consider the more realistic computer vision problem starting with line drawings represented as noisy, scanned images and creating a version of the drawing suitable for CAD system, but the geometric information is not represented explicitly. In fact, some lines may be missing, while extraneous ones might be randomly present due to noise, which may have been caused by folding of the paper, stains, or photocopier noise. Typically, some initial processing for noise cleaning and enhancement is performed, followed by the recognition of drawing information. This is then combined to constitute larger patterns which, in turn form the multiple views that are combined together to form a 2-D CAD version of the drawing.

There are many reasons for the recent interest in this problem. Some of them were summarised in a recent paper [Filipski92] as follows:

1 - There are approximately 3.5 billion engineering drawings of various types in the United States and Canada, with about 26 million new ones added every year. The annual cost of filing, copying, accounting and preparing these drawings for distribution exceed one billion dollars.

2 - Once a drawing is in CAD format, all the advantages of database storage, retrieval and query become available.

3 - A major advantage of a CAD systems is that the time needed to modify a drawing is typically 13% to 33% of the time needed to accomplish the same revision using paper-and-pencil techniques, [Filipski92].

4 - Only a small percentage of the existing, active drawings are available in CAD form. In 1990, about 20% of the drawings were created in CAD form and about 25% were CAD revisions of the older CAD drawings. The remaining 55% were done on the drawing board, using traditional paper-and-pencil drafting techniques.

5 - Cost-benefit analysis shows that if a drawing is expected to be modified several times, it is advantageous to convert it from hard copy to electronic format.

1.5 Representation of Engineering Drawings

Drawings are represented in orthographic projection which is a method of producing a number of separate 2-D inter-related views which are mutually at right angles. Using orthographic projection, even very complex shapes can be fully described in 6 views [Haralick82], although normally 3 views are sufficient. This method, however, does not create an immediate 3-D visual picture of the object as does pictorial projection.

Orthographic projection is based on two principal planes; one horizontal (HP) and one vertical (VP). In first-angle projection, an object is positioned in the space of the first-angle quadrant between the two planes. A view of the object is projected by drawing parallel projecting lines, or projectors, from the object to the vertical principal plane (VP). This view on the VP is called a front view. A view similarly projected on to the horizontal principal plane (HP) is called a plan view. For the complete description of the object, an additional plane, called the auxiliary vertical plane (AVP), is used at 90° to the principal planes, and the view projected on to that plane is called an end view.

By means of projectors, all three planes can be unfolded and three views of the object can be shown simultaneously on drawing paper as shown in Figure 1.3.

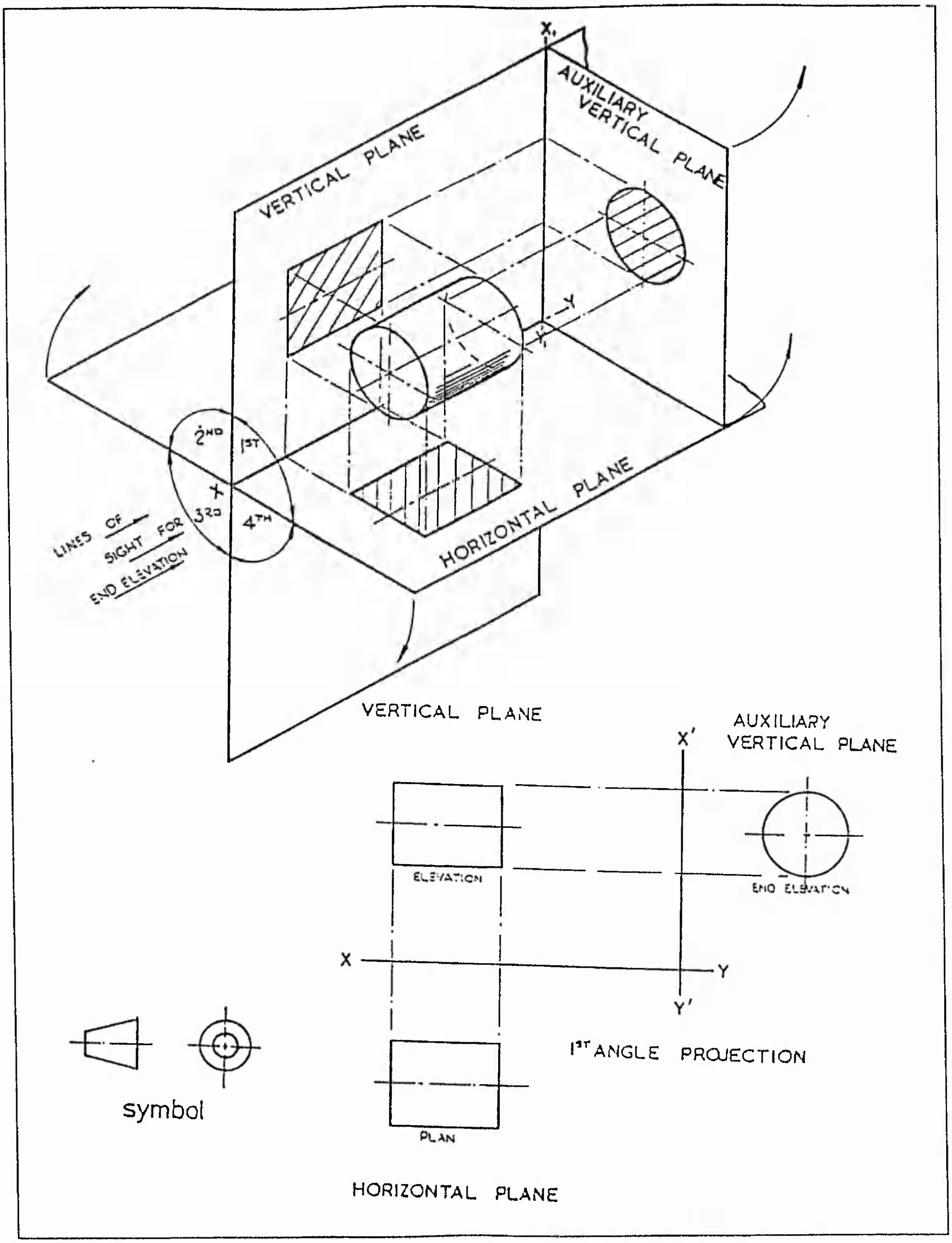


Figure 1.3: Principle of first-angle projection (taken from [Hart75]).

In third-angle projection, an object is positioned in the space of the third-angle quadrant between two principal planes. The planes are imagined to be transparent, and the projected views of the object are viewed through the planes.

By means of projectors, all three planes of the "glass box" can be "unfolded" and three elevations of the object can be shown simultaneously on the drawing paper as in Figure 1.4.

In both first-angle and third-angle projection the elevations are identical, but the positioning of each is different. In third-angle projection, the "transparent" projection plane or view always comes between the eye of the observer and the object.

The symbol used to indicate third-angle projection is derived as for first-angle projection, but the projections are positioned differently. The symbol shows an end projection and a front projection of the circular taper in third-angle projection.

British Standard Institute [BSI] accepts that both first-angle and third-angle projection are internationally accepted methods of presentation, but it stipulates that a drawing should include a symbol indicating which system has been used, to avoid confusion.

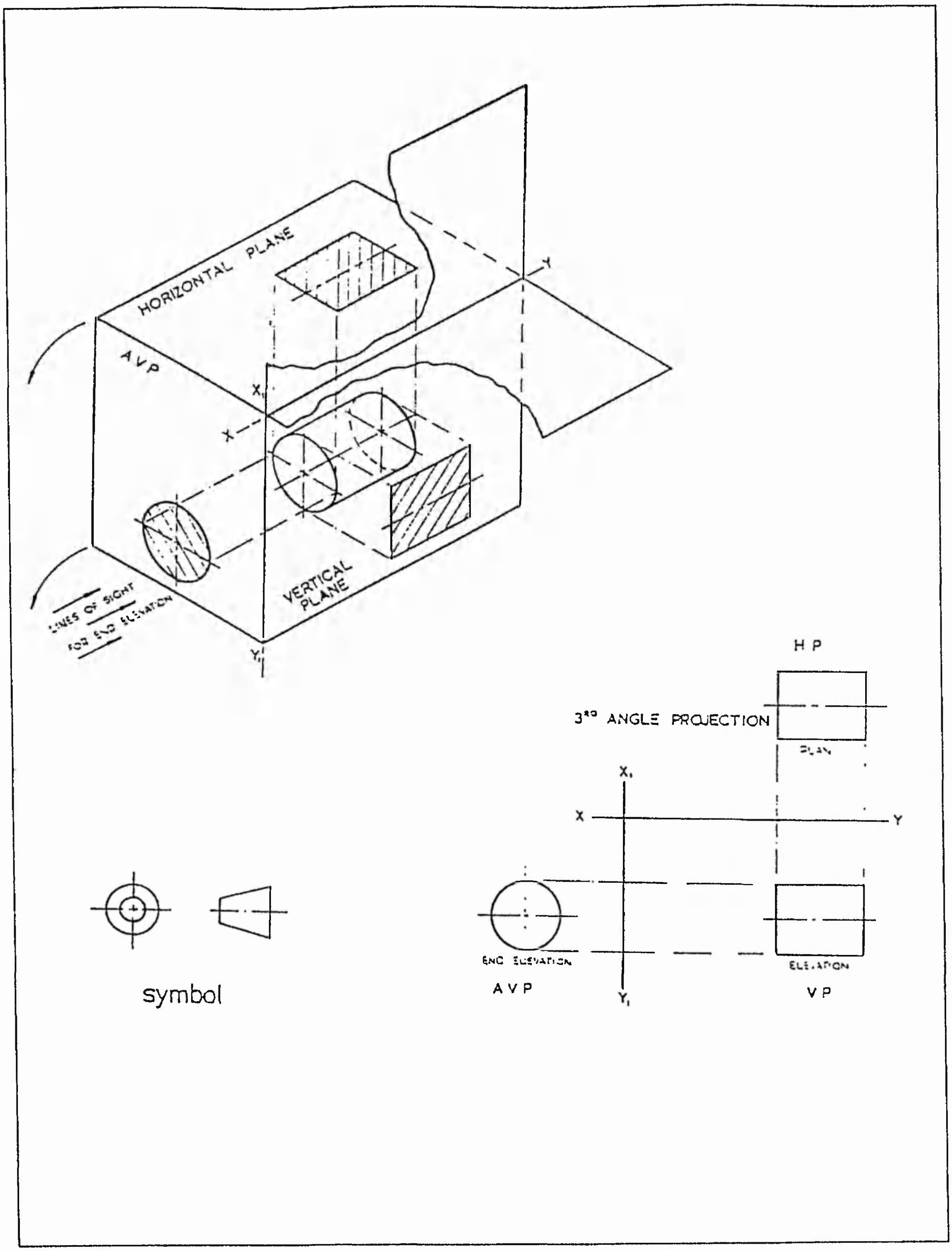


Figure 1.4: Principle of third-angle projection (taken from [Hart75]).

1.6 Principal Projections of Engineering Drawings

In common practice only three projections are used; a Front projection, an End projection, and a plan projection. If chosen carefully, these three projections are often sufficient for a complete description of an object. As a rule, the minimum number of projections should be used, especially to represent simple objects. The projections should be selected so that they clearly indicate all the required detail. Thus, in most orthographic drawings, it is necessary to draw at least two projections of a 3-D object.

A certain amount of imagination is therefore required when interpreting engineering drawings. Obviously, with an object of greater complexity, the reading of three projections, or more, may well be necessary.

1.6.1 Sectional Views

Objects with little interior detail can be represented satisfactorily in orthographic projection by the principal projections, the interior construction being shown by hidden detail lines, Figure 1.5. When the interior detail is more complicated, then the hidden detail lines may be confusing and difficult to interpret correctly. In such cases the draughtsman imagines the object to be cut by a plane as in Figure 1.6, and assumes the part of the object between his eye and the plane to be removed. This exposes the interior detail lines. The resulting view is a sectional view. Hidden detail lines may not be shown on a sectional view unless they are needed to describe the object completely. The position of a cutting plane is shown on a view where it appears as a line, and the direction in which the plane is viewed is given by arrows at each end. Letters on the arrows and a title such as "Section AA" below the sectional view, relate the view to the cutting plane. The cutting plane line is a long thin chain line with a thick long dash at each end. The arrows are placed with their points touching the centre of these thick dash Figure 1.6.

A sectional view is distinguished from principal views by section lines or hatching drawn on the cut surface produced by the section plane. More involved parts may have many cutting planes and projected sections to clearly illustrate involved details.

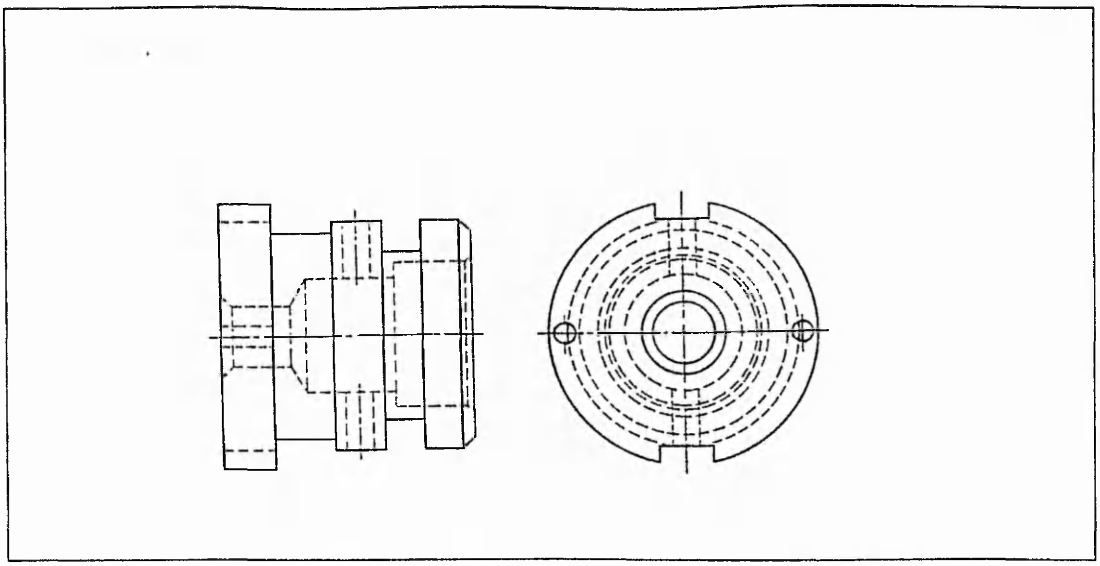


Figure 1.5: Interior shown by hidden detail (taken from [Pickup79]).

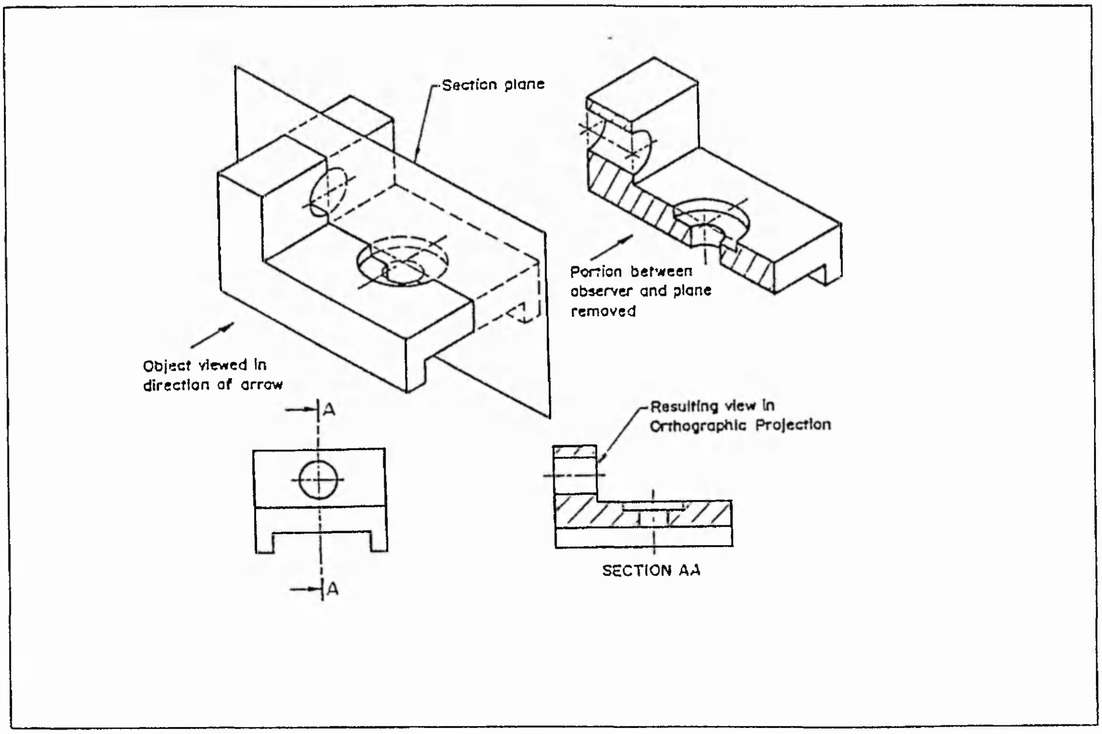


Figure 1.6: Sectional view (taken from [Pickup79]).

1.6.2 Assembly Drawings

In general, a part drawing does not show a clear picture of the parts described by the drawing. Therefore, a number of sub-assembly drawings of all the parts required is added to the drawing which guides the workman to assemble the parts together.

Each item on the part drawing must be fully dimensioned: each is an engineering drawing complete in itself and the workman must be able to make it from the dimensions given without reference to any other part. On the other hand, sub-assembly drawings should have the overall and other important dimensions only, the detail dimensions should be on the detail drawing, Figure 1.7.

The main part of the object (MACHINE VICE) represented in Figure 1.7 is described by two projections and a sectional view. While, all other sub-assembly drawings are described by only two projections. Each sub-assembly drawing is labeled, an indication of the part of the object it represents.

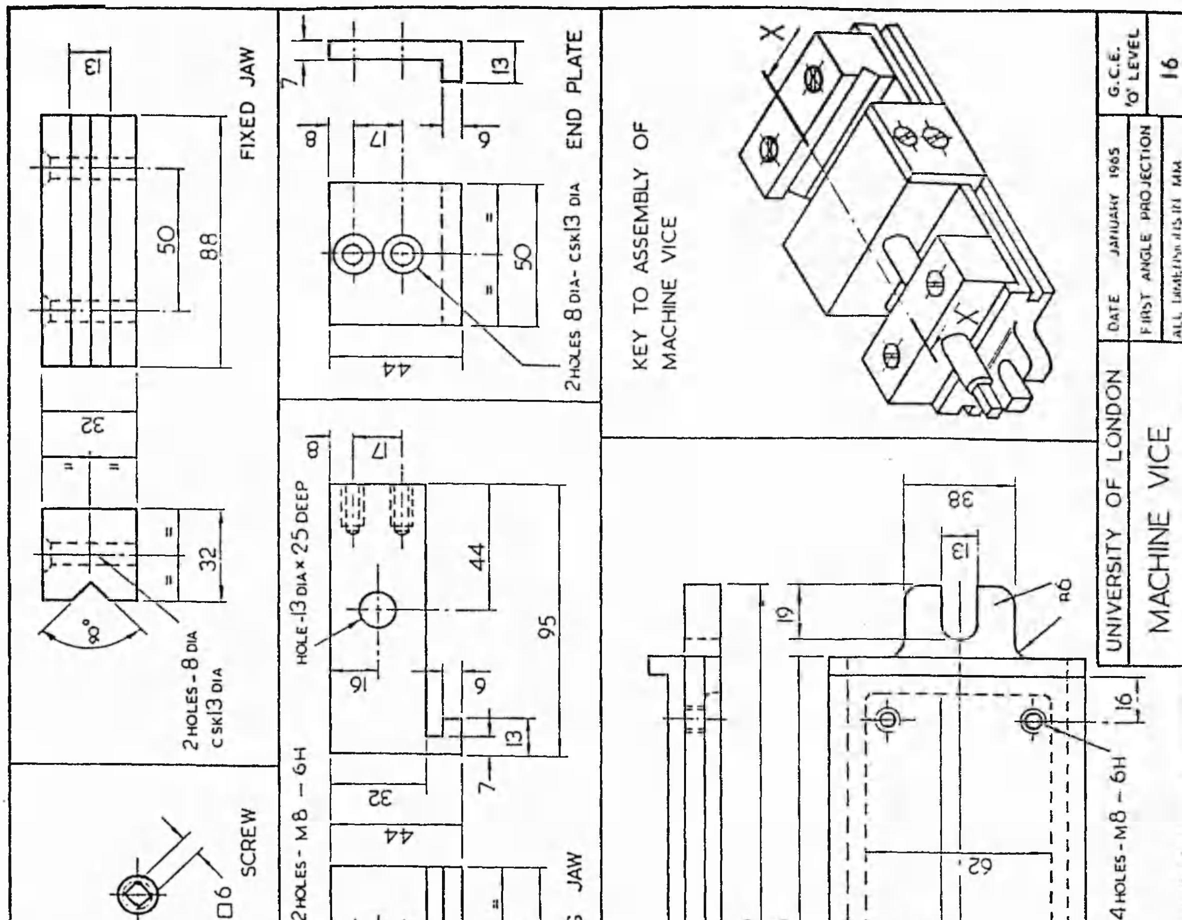


Figure 1.7: Example of Assembly drawing (taken from [Hart75]).

1.6.3 Other Views and Conventions

Producing the drawing for a very simple component may take the draughtsman several hours, whilst the drawing of a large complicated casting may occupy him for days. One way of minimising the time spent on producing a drawing is by using conventions and conventional representations on the drawing.

In one sense the whole drawing is a convention, since it represents the surfaces of the physical object by lines on a sheet of paper. By convention, different types of line have different meanings.

The use of conventions is to make the engineering drawing clearer, it is more like a standard language of communication. Some conventions also save space on the drawing. This may mean that a view can be drawn to a larger scale, thus improving the readability of the drawing. On the other hand, space saved may mean that a smaller size of drawing sheet can be used. Various conventions to achieve both these objectives are recommended in [BSI].

1.6.3.1 Views for Symmetrical Components

Many engineering components are symmetrical about a centre line or axis and can often be represented by a half view. To distinguish this symmetrical view, two short, thick, parallel lines are drawn across the symmetry demarcation line at each end. These symmetry symbols are at right angles to the symmetry demarcation line. To emphasise further that symmetrical view is shown, the outlines of the part extended slightly beyond the line of symmetry. For parts which are symmetrical about two axes at right angles, a quarter view may be used. Figure 1.8 shows an example of symmetrical views.

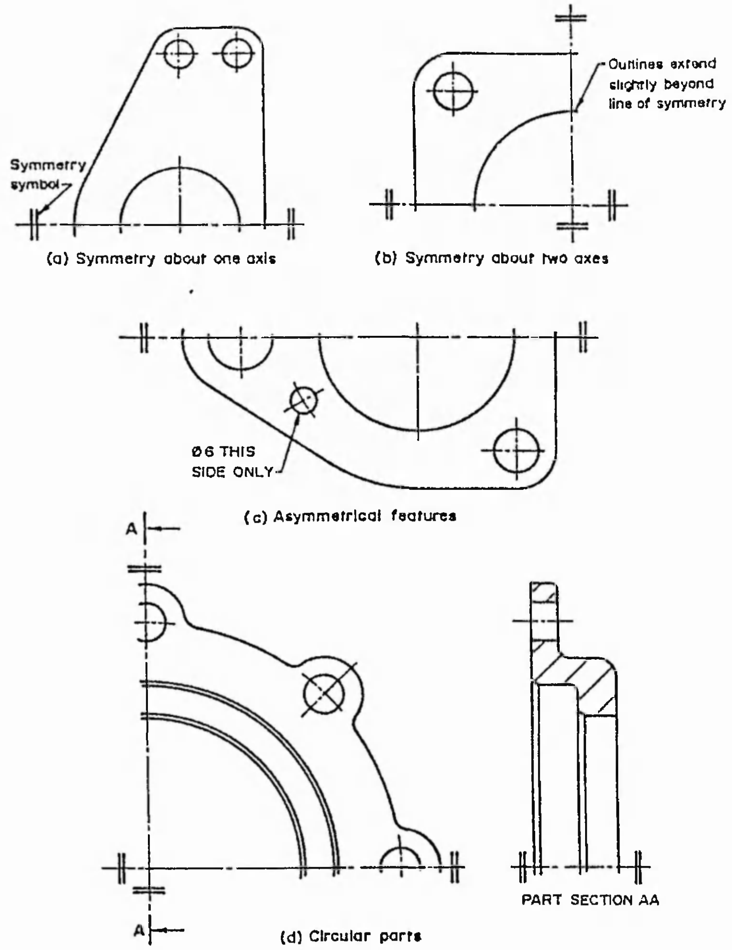
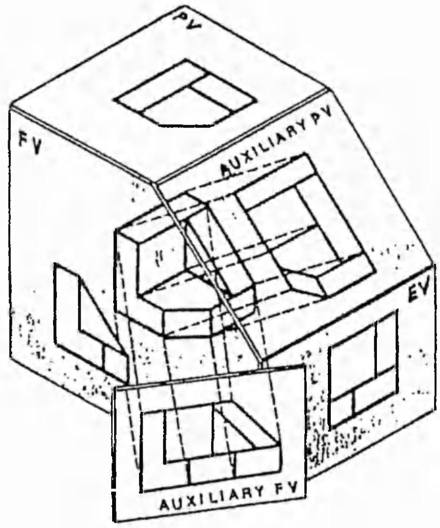


Figure 1.8: Example of symmetrical components (taken from [Pickup79]).

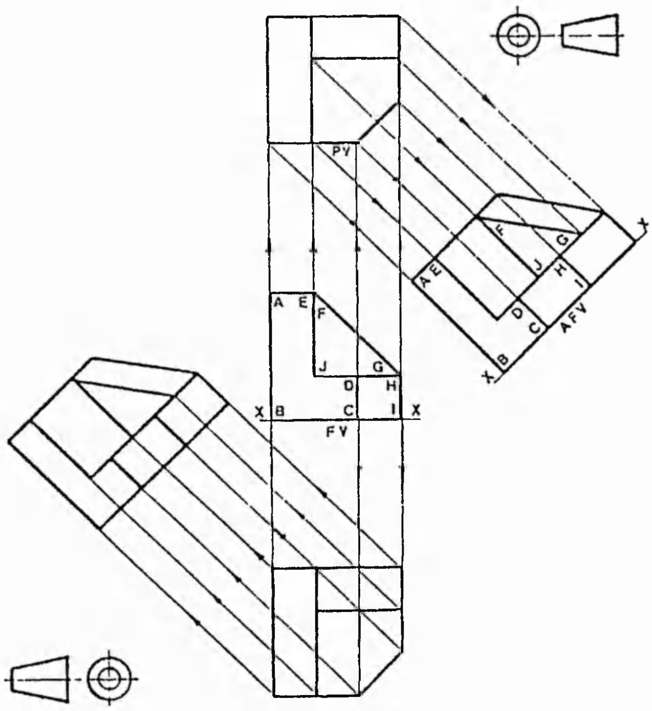
1.6.3.2 Auxiliary Views

Occasionally a component has surfaces which are not parallel with any of the principal planes of orthographic projection and which therefore cannot be clearly defined or dimensioned. To draw the true shapes of those surfaces, additional views are required showing the surfaces as they appear when looking directly at them. These views are called *auxiliary views*.

Figure 1.9a shows an object suspended inside a third-angle projection 'glass box' which consists of three principal planes and two auxiliary planes. The auxiliary horizontal plane shows the auxiliary plan view (APV) and the auxiliary vertical plane shows the auxiliary front view (AFV). Figure 1.9b shows the auxiliary front view is obtained in the third and first-angle projections.



(a) Auxiliary planes



(b) Auxiliary front view

Figure 1.9: Example of Auxiliary view (taken from [Ostrowsky79]).

1.6.3.3 Repetitive Information

Most drawings may have identical parts or *features* present on them. It is therefore, unnecessary to illustrate all of them on the drawing. Repeated illustrations of these *features* may be avoided by drawing one, and indicating the positions of the others by some conventions such as centre lines, Figure 1.10. The rest of the information may be given in a note. This information may indicate the number, size, and other descriptions that enable the reader to visualise the *features*.

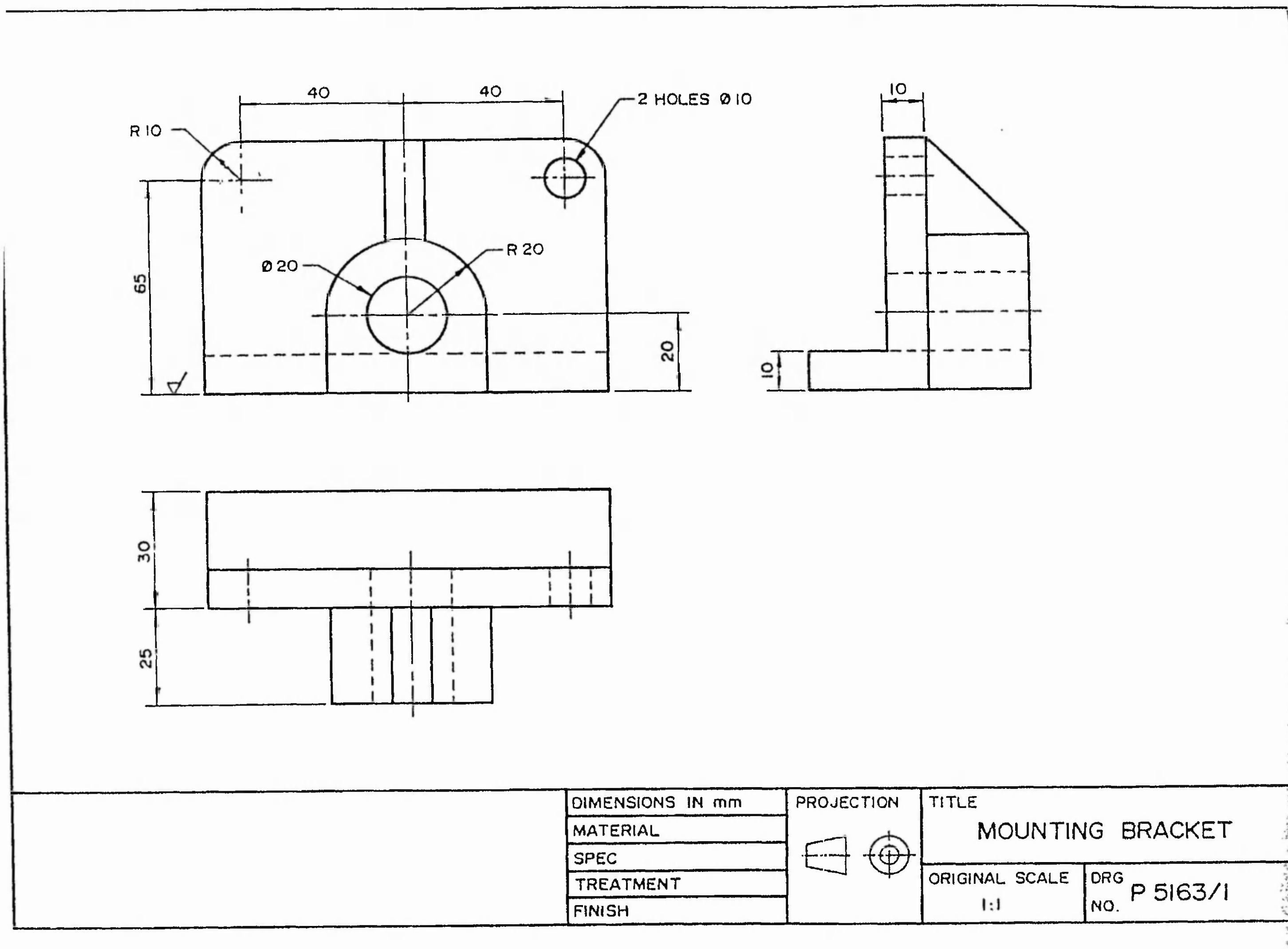


Figure 1.10: Engineering Drawing with repeated features (taken from [Collier82]). The text in the Front view describes 2 holes but only one is drawn.

1.7 Simplifications, Ambiguities and Errors in Engineering Drawings

Most engineering drawings are imperfect, and further imperfections or errors are often caused by the various processes which convert engineering drawings on paper into a CAD data file. In addition, part of a drawing might have been omitted or simplified and replaced by a note referring to that part, as has been shown in Figure 1.11.

During the scanning process, extraneous information can be added to the data file where there are creases or dirt on the paper drawing. Lines can be omitted or disconnected because all or part of them are faded. Algorithms have been developed by a previous research to detect these errors and correct them [Shaw95].

Because of the designer's intention to simplify the drawing, part of the drawing might be omitted and a phrase would describe it instead. Therefore, the geometry does not represent the object on the drawing perfectly. These simplifications cannot be dealt with by any bottom-up procedure that totally relies on the geometry.

These technical details are very important to the reconstruction of a consistent 3-D object. Thus, reconstructing 3-D objects using only the geometric information is most likely to produce inconsistent or incomplete objects.

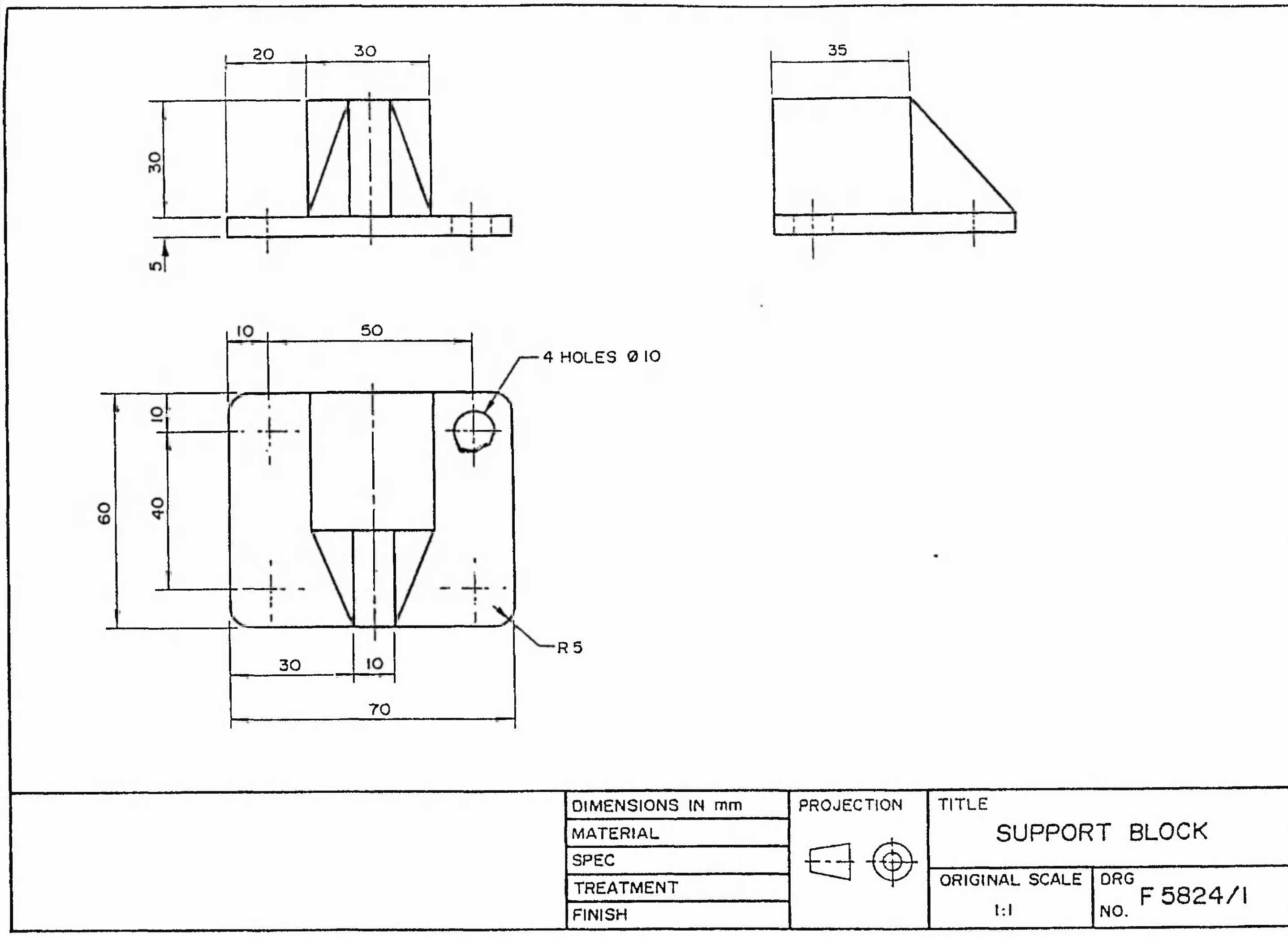


Figure 1.11: Engineering Drawing with repeated features (taken from [Collier82]). The text in the Top view describes 4 holes but only one is drawn and also is distorted.

Some of the sub-assemblies of Figure 1.7 describe *features* which are omitted in order to simplify the drawing. In the BLOCK sub-assembly, the phrase " 2 HOLES 8 DIA - CSK 13 DIA" describes the existence of holes which has no reference in the other projection. Further the END PLATE sub-assembly, the phrase " 2 HOLES 8 DIA - CSK 13 DIA", again describes the existence of holes but has no reference in the other projection.

These representations do not completely describe the engineering parts on them. Therefore, reconstructing 3-D models from these representations using only the low level information (geometric data) will fail to produce consistent models.

Algorithms have been developed to analyse phrases to recover simplified part of the object are described in chapters 4 and 5. This process, of "understanding the drawing", will bridge the gap between recognition of the drawing and the construction of a consistent 3-D model. Therefore, many possibilities of ambiguous interpretation will be eliminated by this process.

1.8 Objectives of this Project

Many existing engineering drawings have part of the geometry simplified or abbreviated with full description provided by the text on the drawing. Others have imperfections and distortions. Scanning may produce more distortions. Reconstructing a 3-D model from these drawings with no consideration given to the text, may produce an incomplete, erroneous or even ambiguous result. Thus, textual descriptions are very important and may hold the answers to many of the omissions or ambiguities that occur in the 3-D reconstruction.

Engineering drawings are scanned to produce a bitmap image that can be stored in a computer. This image is then vectorised to produce a vector representation of the drawing rather than a pixel representation. The output information from the vectorisation process requires the recognition of the different drawing elements on the drawing before a method can embark on interpreting the textual information. Therefore, the aims of this project are to develop algorithms to:

(a) - Identify principal projections (at low level).

(b) - Identify different line types and eliminate unnecessary information (at low level).

(c) - Allow the correction of the geometric information on the drawing by analysing the textual information (at high level).

(d) - Recognise features and their geometric descriptions (at high level).

To this end, all the necessary information on the drawing must be identified. Next it is necessary to interpret the text to recover abbreviated or distorted parts on the drawings. Therefore, the objectives of this project are to provide new algorithms which can be used as a basis for interpreting drawings in the specific "engineering drawing" domain in order to produce a complete geometric description of the simplified engineering drawing.

Several steps are required in order to process the information of an engineering drawing in order to produce a complete geometric description of the drawing.

(1) - Identify views of the drawing.

(2) - Identify different line types.

(3) - Identify simple dimension text and dimension lines.

(4) - Separate more complex textual annotations from drawing information.

(5) - Recognise textual information.

(6) - Generate a CAD data file that contains a line-drawing, text (dimensioning and textual annotations) and symbols.

(7) - Interpret more complex textual annotations.

> (a) - Define rules to parse phrases.
>
> (b) - Define a Syntactic Net of words and their relations.
>
> (c) - Generate meaning from the phrases (Semantic Templates).
>
> (d) - Generate the geometric representation of the phrases.

(8) - Recognise features on engineering drawing and associate them with the text.

This thesis is mainly concerned with the problems indicated in steps 7 and 8, however, simple algorithms have been developed to deal with the problems in steps 1 and 2.

The overall aim of this research, is to develop algorithms capable of interpreting textual information to understand the drawing in order to identify simplifications and correct distortions.

Chapter 2

Survey of Previous Work

2.1 Introduction

In this chapter the different schemes for representing 3-D objects are described, a target system has been outlined and previous research is reviewed. Previous research to interpret engineering drawings can be categorised in several ways. One criterion is the level of information on engineering drawings that can be used to generate a consistent 3-D object. Some systems assume the geometric information on the drawing provides a complete description of the object. While others take the more realistic approach in using additional information such as textual information to strengthen the assumption made by the first group.

Some type of Boundary representation seems to be the most common choice for 3-D reconstruction, while a limited form of Constructive Solid Geometry is used in other systems. The types of objects that can be interpreted depend on the degree of sophistication of the matching between views and the 3-D representation scheme used. The most sophisticated methods produced so far are limited to constructing quadratic surfaces, i.e. planes, cylinders, cones, tori and spheres while others can only deal with plane faced shapes. The types of entities matched in the different views also differ from system to system; some methods match simple entities such as individual lines while others match more complex ones such as contours. A few of the methods start with scanned bitmap images [Bergengruen89], [Yoshiura84], but the majority assume that geometric information is complete [Aldefeld83a], [Aldefeld83b], [Aldefeld84], [Idesawa73], [Preiss81], [Preiss84], [Wesley80], [Wesley81].

Most of the methods for reconstructing 3-D models from projections involve matching entities (nodes, lines or contours) in one view with corresponding entities in the other views. Once a match has been found between two 2-D entities a 3-D entity can be created by combining the co-ordinates of corresponding entities in the two views to produce an entity in 3-D space. In addition the topological relationships between nodes (i.e. how they are connected by lines) in the 2-D views are used to determine which entity in the 3-D space are connected by edges. The results of matching entities are stored in an updated data structures used to represent the 3-D model.

When discussing different methods of interpreting engineering drawings it is necessary to be precise about the terms used to describe different entities. In the literature different people used different terms for the same things, so to avoid confusion in this thesis the definitions given in the glossary are used. Some of these terms are illustrated in Figure 2.1.

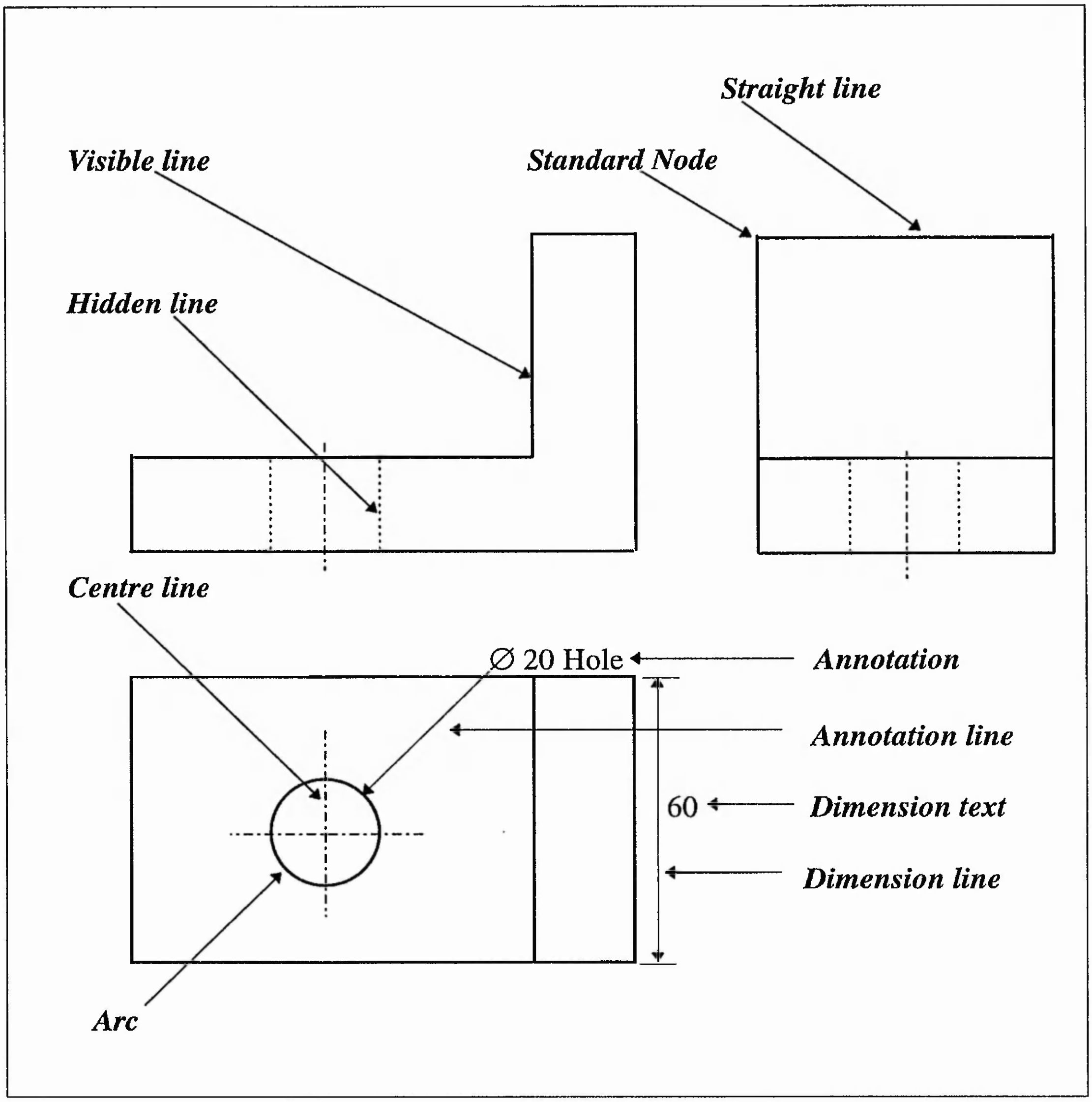


Figure 2.1: Different geometric and textual items of Engineering Drawings.

2.2 Schemes for 3-D representation

In Computer Aided Design systems, the consistency and validity of a model is most important in ensuring that the model can exist and can be reconstructed automatically. Therefore, a solid model which allows the user to specify the final object accurately is very important. Fortunately many engineering objects can be built up by joining together a limited number of types of surface patches or alternatively by combining together a limited number of solid shapes using the set operators: union, intersection and difference.

The majority of present day CAD systems use either a form of Boundary representation or Constructive Solid Geometry or some combination of the two. Others schemes such as Octrees and spacial enumeration techniques probably are not appropriate for the task of interpreting engineering drawings as neither scheme represents edges explicitly.

2.2.1 Constructive Solid Geometry (CSG) Scheme

In Constructive Solid Geometry, an object can be represented by hierarchically combining simple primitives using Boolean set operators: union, intersection and difference. For example, an object can be represented by a binary tree in which each terminal node is a primitive and each nonterminal node is a set operation Figure 2.2. The composite solid can be displayed without determination of the surfaces in the CSG model. An important advantage of CSG is that it has a conciseness of data structure and ease of creation. This makes it less expensive and easy to be modified and manipulated. This scheme is good for representing many of the shapes that occur in engineering drawing, however it cannot cope with the more complicated shapes because of the limited primitives that are used in this scheme.

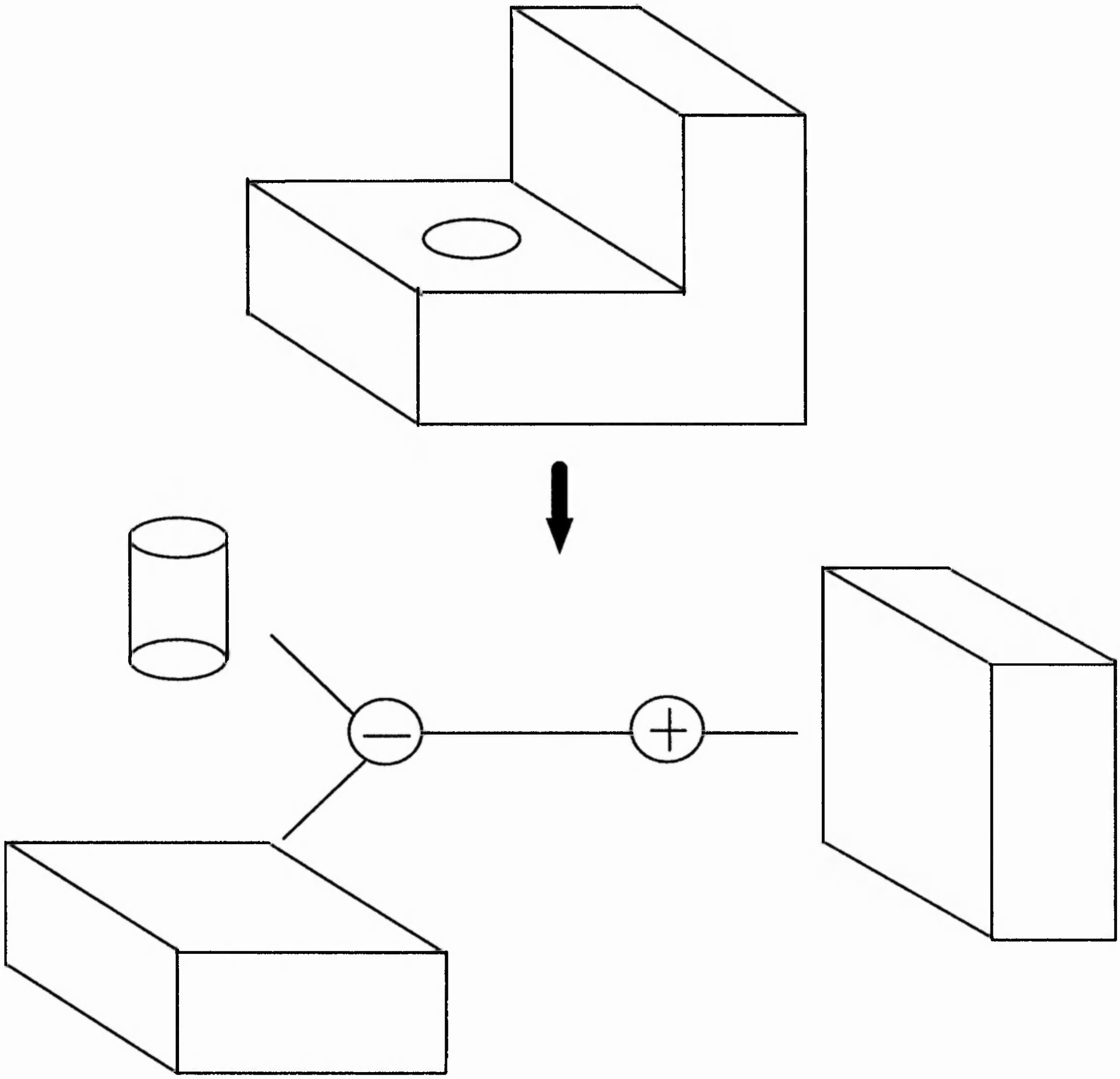


Figure 2.2: A CSG tree representing an object.

2.2.2 Boundary representation (B-rep) Scheme

This scheme represents the faces of an object together with the edges forming the boundaries of the faces and the vertices forming the ends of the edges. In this scheme, any point in space can be determined uniquely. The information is much more explicit than in Constructive Solid Geometry. Using this scheme, an object is represented by topological and geometrical information. Topological information specifies the connection between the faces, edges and vertices of the object; geometrical information specifies the co-ordinates of vertex, edge and face equations, Figure 2.3.

The advantages of this scheme are that it requires less computation than CSG in order to display an object. Surface properties are easier to represent and surface areas can be calculated more easily. Most engineering object can be described using combinations of faces of a few simple types. This is because many manufactured parts are created using tools that move in a linear or a rotational manner giving rise to a limited range of surface types.

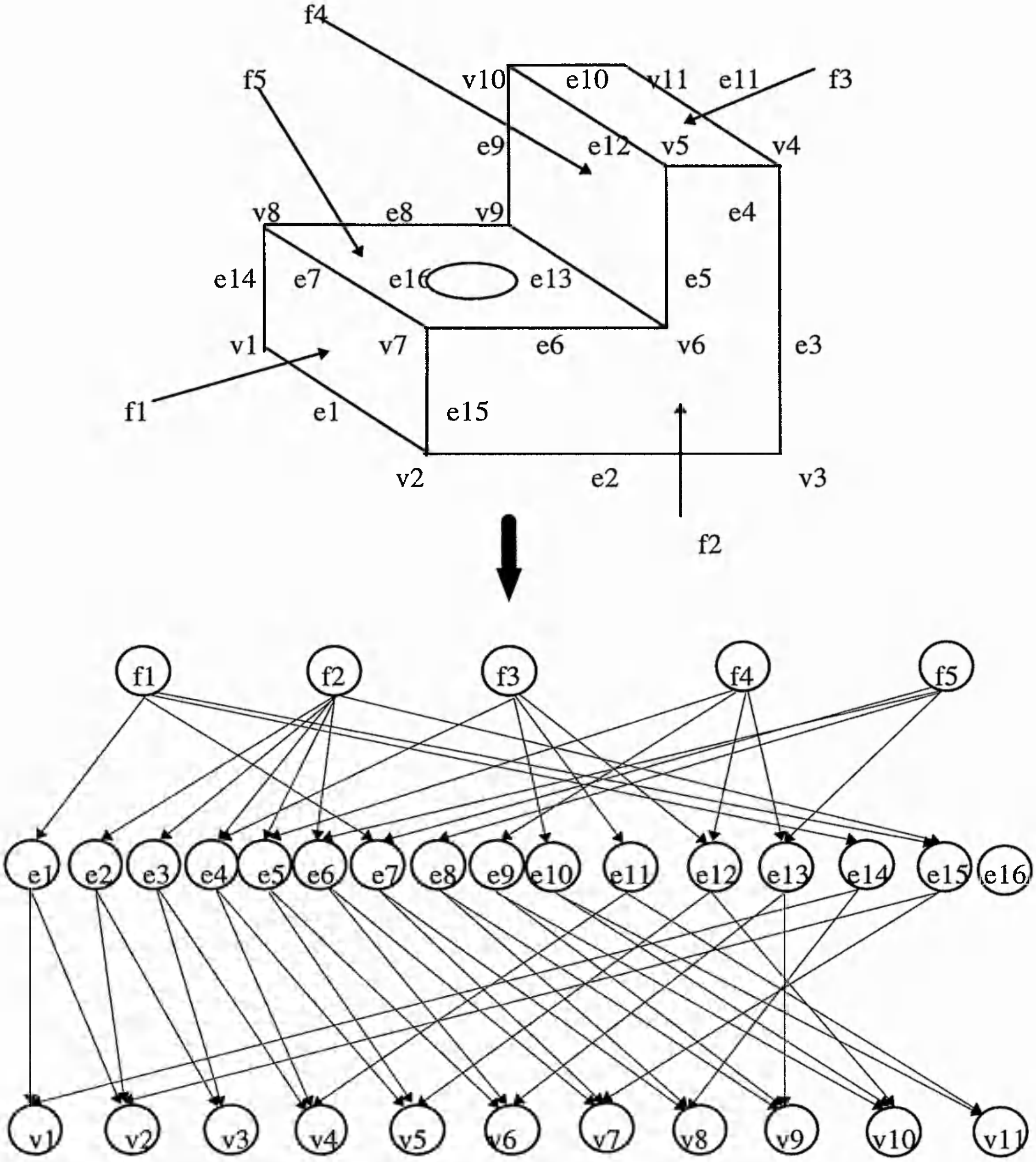


Figure 2.3: Boundary representation.

2.3 Advance System for Interpreting Engineering Drawings

This system starts with noisy orthographic line drawings and produce 3-D solid descriptions. Most systems fall into two very broad categories; the low level (line drawing) and the high level (dimensioning and textual annotations). The later carry more technical information about the object.

There are a few systems that incorporate the low and high levels of information for the reconstruction of 3-D solid descriptions [Lysak90], [Lysak91], [Nagasamy90], [Nagasamy91]. Those that are incorporating the two levels of information have to compensate by imposing some restrictions, such as solving the reconstruction problem from noise-free projections. Also only the dimension annotation are utilised with these systems.

Joseph and Pridmore [Joseph92] used a top-down approach in their 3-D CAD interpretation system called Anon. Anon takes as input an image from scanned engineering drawing with dimension annotation, and produces 2-D CAD descriptions. It does not handle multiple object assembly CAD drawings. It does not handle non-polyhedral objects either.

In the low level category, there is a large body of literature on feature extraction and grouping of features into larger symbolic structures. But most of the current systems are stuck at this phase and have not been able to successfully reconstruct the 3-D solid from the extracted primitives. The reason for this state of affairs is that the current systems do not take into consideration the geometry of line-drawing formation process. That is, they do not have a model of the line-drawing image formation process. Thus, although there is a lot of information the current systems could have exploited by using a model-based approach, they resort to *ad hoc* techniques, where they have a flat rule-based reasoning system that tends to become very slow due to the large number of rules.

At the other extreme is the high level category with many theoretical and implementation papers describing 3-D reconstruction from noise-free data. Although these papers capture the geometry of orthographic projections, which the image processing-type systems have not, these papers do not at all consider missing and extraneous lines or loops. Thus, these algorithms cannot handle real images.

The performance evaluation should be based on a reasonably large set of simulated and real engineering drawings, which the system is supposed to process and understand. Many experimental systems work well on a small selected set of trial drawings but perform poorly over a large set. Different noise levels of various possible types should be tested to determine the robustness of the system and the noise level at which their performance is unacceptable.

The automatic interpretation of engineering drawings is a difficult task due to the complexity and diversity of such drawings. The dual nature of engineering drawings, carrying information in both graphical and textual form, together with the fact that real engineering drawings may be of poor quality and that their many features depend on factors such as the type of drawing, make their analysis difficult. Thus, it is necessary to use as much a priori knowledge as possible in their interpretation. The knowledge has to guide the early vision tasks in the processing of the drawing making the low-level techniques more efficient.

Geometric information can provide knowledge of the association of the different entities describing the object. The recognition of line types provides information about the visible and hidden parts of the object as well as convention entities that may help identifying the location of the object or its features on the drawing.

To achieve an automatic interpretation of engineering drawings, all the information provided on the drawing must be identified. The interpretation process then must be guided into using the relevant information in order to produce a consistent interpretation.

Interpreting drawings which have annotations referring to geometric features of the object, requires a procedure that translates these annotations into geometric data. In this thesis an algorithm has been developed to parse textual information and generate geometric descriptions.

The requirement for an advanced system is basically to interpret as much information available on the drawing as possible. Thus to summarise the requirement for this system, the following need to be accomplished.

1 - Produce a bitmap image of the engineering drawing.

2 - Generate a CAD data file that contains a line-drawing, text (dimensioning and textual annotations) and symbols, these can be recognised by vectorisation and using OCR techniques on the bitmap image.

3 - Recognise dimension-sets.

4 - Use the dimension-sets to check the precision of the line-drawing, as well as using the textual annotation to correct any distortions and recover simplified features.

5 - Perform reconstruction on this CAD data file to produce a 3-D description.

Furthermore, other conventional information such as sectional views may be used depending on the complexity of the object. Using a back track or a reverse process of the reconstruction of 3-D description in order to evaluate the consistency of the 3-D model with its 2-D representation.

2.4 Comparison of Previous Work

Much of previous work in the reconstruction of 3-D objects has mainly concentrated on using only the low level (line data) information in their reconstruction. This information will not be sufficient to interpret a drawing having part of its geometry simplified or distorted. Keeping this in mind, all research in this area is dedicated toward producing a system that automatically interprets engineering drawings. This, therefore, requires the interpretation of all information it can use on the drawings in order to produce consistent 3-D models.

Annotations and dimensions are part of almost every engineering drawing. Dimension can define the actual length of the outlines. This information can be used to correct the actual size of the object. Obviously, this is very important for the design and manufacture of the part, where the precision is basically an essential element of these processes.

Modern engineering drawings can be generated by a CAD system. These can be easily manipulated and converted to different format. The main concern, is that for the vast number of already existing old engineering drawings. Since CAD systems were not available in those old days, these drawings were generated by hand. Knowing that designers' time is so precious, repeated features of the drawings are simplified and replaced by other information.

2.4.1 Low Level Approach

Over two decades ago [Idesawa73] attempted to construct a solid model from orthographic projections. He described how he thinks an engineer would interpret a drawing and pointed out the difficulty of constructing a system which generates a solid object from three views by machine analogously to the human case. He made the assumption that a straight line in 2-D view corresponds to an edge in the 3-D object. Thus, he described his method by matching 2-D nodes and lines in different views and generate their corresponding 3-D vertices and edges. He realised that so called ghost figures are generated, and set some constraints to eliminate them. He then identified hidden lines by testing the intersection of lines with the boundaries of faces.

Idesawa's approach is limited to polyhedral objects and only capable of dealing with drawings containing one object and producing only one interpretation.

A few years later [Wesley80], developed a method that finds all possible objects which have a given wireframe. They described their wireframe algorithm in a number of stages, where in the first stage the input data has been checked for the validity of the wireframe. The next two stages find all planes which contain at least two intersecting edges. Faces are then constructed which are defined as the closure of a non-empty, bounded, connected, coplanar whose boundary is the union of a finite number of line segments. They applied some tests for the validity of the input data and the data created by their construction. These tests ensure that every vertex belongs to at least three edges, and edges intersect only at endpoints. Other tests check the legality of intersection between faces, which have the property that faces belonging to them intersect only at boundary points of the faces. During the next stage, which involves assembling all faces into candidate blocks, faces with a common edge are combined. Finally, the blocks are assembled to generate all objects with a given wireframe.

Wesley and Markowsky later produce a method for reconstructing wireframe model [Wesley81] described by a set of two dimensional projections (such as engineering drawings). This method constructs the wireframe model by matching all nodes and lines in different projections and combining them into vertices and edges respectively. The limitation is that it can only cope with drawings containing polyhedral objects, but they have improved on Idesawa's method in one respect; it can cope with more than one object on the drawing and more than one interpretation if more than one exists.

K. Preiss [Preiss81] developed a method that interprets a drawing by finding all vertices, edges and faces of the object. He described his method by finding paths (contours) of solid lines in each view and identified their neighbouring contour. These contours are then matched with contours in a second view using the third view to confirm the matching. He indicates that for each contour of each view, if an interpretation exists, it will be found. After this stage, a 3-D interpretation will exist for the vertices, edges and faces on the three visible sides of the object. He interprets dashed lines separately, by comparing each line with lines in the other views and creating all possible edges. He then applies some form of scene analysis program, which postulate that each vertex and the edges attached to it are explicitly identified in order to assemble the object. This method is similar to that of Idesawa in that it can only cope with polyhedral objects, and can only interpret a drawing containing one object and produce only one interpretation.

Preiss later extended his method to include objects that have cylindrical faces [Preiss84]. Because of this, he introduced the idea of using more constraint relations, but the principle of his method remain unchanged. He used a method similar to that of Wesley and Markowsky's to generate vertices and edges with addition constraints to interpret "silhouette" edges. He described this method as a problem of constraint propagation. First, a set of candidate vertices is assembled using constraints applied only to the vertices. A set of candidate edges is assembled using constraints applied to single edges. Inconsistent edges are deleted from the candidate set by using a constraint applicable to pairs of edges. Finally, edges are related to generate faces, using constraints applicable to edges in a face and constraints applicable between faces.

This method has an advantage over Wesley and Markowsky's in that it has the capability of interpreting objects that have plane or cylindrical faces. Interpreting objects containing cylindrical faces introduce so called silhouette edges which indicate where two surfaces meet tangentially.

B. Aldefeld and H. Richter [Aldefeld83b], [Aldefeld84] have developed a method that uses a Constructive Solid Geometry on object of uniform thickness. Their approach is to view a complex part as consisting of a number of elementary objects and to recognise these from their specific structural pattern in the 2-D representation. This approach accommodates a diversity of complex structures that lend themselves to a volume-oriented way of geometric modeling. They describe their algorithm by using a Constructive Solid Geometry tree of the primitives and the set operators UNION, INTERSECTION and DIFFERENCE, to reconstruct their 3-D representation of the elementary volume.

B. Aldefeld [Aldefeld83a] has also implemented a basic concept to describe the structural information in terms of a semantic network composed of entities, relationships and attributes. Interpretation proceeds by establishing the relational structure of the primitives followed by a gradual building up of the higher level structures until all objects have been found.

Aldefeld and Richter's method is limited to interpreting objects of uniform thickness, and user guidance is needed to identify ambiguous elementary volumes.

O. Bergengruen [Bergengruen89] used a similar approach to Aldefeld and Richter in his reconstruction, searching for instances of primitive solids in the image. He starts reconstructing the 3-D model from pixel images rather than from a complete set of geometric primitives (lines, circular arcs and circles). The limitation to Bergengruens' method is that it relies on the user interaction. Each time an instance is found, a hypothesis is generated that the new primitive corresponds to a valid decomposition. To verify this hypothesis, the user is asked to confirm or deny the hypothesis. The class of drawings that his method is capable of reconstructing is constrained to some assumptions.

Sakurai and Gossard [Sakurai83] developed a method that construct a wider range of objects. They extended Wesley and Markowsky's method by interpreting objects containing planar, cylindrical, conical, spherical and toroidal faces. Their interpretation of curved faces introduced a number of complications, for example by generating "silhouette" and "tangency" features. These features are not present in all views which make the matching process more complex. This method has the advantage over previous methods in that it can deal with a wider range of objects.

An approach similar to Sakurai and Gossard has been used by [Lequette88], with possible reconstruction of objects from two views.

M. H.Kuo [Kuo96], in a recent PhD thesis, has developed a method for automatically converting 2D graphic elements to a 3D wireframe model. This wireframe model then translated to a boundary representation (solid). Initially, B-rep oriented approach is used to reconstruct a wireframe model. CSG oriented approach uses a translational sweep operation to construct the primitives, then combines them using the set operators to form the final solid object. This method detects and deletes pseudo elements in the wireframe.

The principle of this method does not differ much from the method developed by [Shaw95]. However, it deals with objects having conic edges and quadric surfaces, and gives multiple solutions. It also gives a mathematical description of the reconstruction procedure.

2.4.2 High Level Approach

Unlike the low level approach, little work has been done to explore the textual information on engineering drawings. Almost all engineering drawings contain dimensioning and annotations, this information complements the geometric information of the drawing.

The assumption made by previous research of interpreting noise-free images may not require this technical information, since the geometry completely describes the object on the drawing. However, the goal of many researchers is the automatic interpretation of engineering drawings. This requires the interpretation of real engineering drawings and therefore, avoiding any unnecessary restrictions imposed on processing the drawings.

Scanning and vectorising of engineering drawings may produce an incomplete geometric description of the object. Thus, textual information is required to correct ambiguities and complete the geometric description.

Previous research was devoted to produce a complete 3-D representation from a complete 2-D drawing. At first, drawings containing single object with only polyhedral shapes are interpreted [Idesawa73], next drawing comprising multiple objects are considered [Wesley80], [Wesley81], then the interpretation is extended to include objects containing curved edges [Sakurai83]. The research carried on into interpreting more complicated objects.

Dimensioning and annotation are simply part of every engineering drawing, and interpreting this information can extend the interpretation process to include a wider range of engineering drawings. In addition, this information can add an extra dimension into the interpretation process. Such procedures would need some degree of intelligence in understanding simplification on drawings, and will then be able to identify and correct some distortions. Thus, interpreting textual information can help to convert existing drawings directly into a CAD database.

Hiroshi Yoshiura [Yoshiura84] implemented a method using a top-down approach which deals with interpreting the textual information on drawings prior to the reconstruction of (3-D) objects. His method requires an expert user to enter the sentences which are then analysed by a computer program. Natural language processing techniques are used to interpret these sentences and to deduce the existence of geometric features of the object. A 3-D model is then reconstructed using a Constructive Solid Geometry by matching features in different views. This approach is very important to the interpretation of engineering drawings that contain text. However, Yoshiura in his interpretation used sentences represented in the general domain, although most text on the drawing is represented as phrases rather than full sentences.

Later, researchers have ignored the presence of annotations which are present in practically most engineering drawings. Annotations are aimed at providing the engineer with additional information about the object which qualifies its geometric description. While geometric lines describe the shape of the object, annotations assert the presence of this information. The simple text expressed by dimensions and are used by a number of researchers [Bergengruen89], [Dori90], [Lai94], [Shaw95] and others. Seiichi Nishihara [Nishihara94] identifies the difference between recognition and understanding the drawing using geometric knowledge not text. His understanding of the drawing is basically to reconstruct objects as human would by combining the faces of the object. The reconstruction process described in his method divided into two parts: the pre-processing phase (to recognise drawings) and the main phase (to understand drawings).

2.4.3 Previous Work done in the Department

M. Waite [Waite89] developed a method that uses Plane-sweep algorithms to derive his data structures. The data structures used by Waite to represent engineering drawing is a development from the 'Modified-Edge Data Structures' described in [Weiler85]. Waite extended Plane-Sweep algorithms which were required in his work to enable them to cope with the variety of general geometric conditions appearing in line-drawings. The approach used by Waite to reconstruct objects is similar to that of B. Aldefeld and H. Richter. The primitives used in his method are limited to uniform thickness parts with polygonal cross sections, and combined by placing them adjacent to each other. Hidden lines are not distinguished from solid lines and the algorithm relies on the user interaction to select the outlines of the uniform thickness parts.

G. Shaw [Shaw95] developed a method which uses a similar approach to that of Markowsky et al. and Sakurai et al. in his reconstruction of 3-D models. He extended their work to allow reconstruction of objects from imperfect images. His reconstruction method can deal with objects containing planar, cylindrical and conical surfaces. He developed a method that corrects errors such as gaps in lines, rotation and stretching of the drawing and possible missing lines. His algorithms also make use of the written dimensions on the drawing to correct the positions of nodes in the 2-D structure and vertices in the 3-D model. False features generated during association of views are eliminated by imposing 3-D constraints. Shaw's method relies on the values of certain parameters which determine the tolerances used for example when matching features in different views. Apart from the dimensions, his method does not use other more complex, descriptive textual information, but concentrates on the low level information.

2.5 Summary

The work of Sakurai et al and Lequette is the most advanced and interprets a wider range of objects. [Shaw95] has extended their work by including dimensions to correct the actual geometry.

Hiroshi Yoshiura [Yoshiura84] used Natural Language Processing technique to analyse engineering sentences. His method requires an expert user to enter the sentences after studying the drawing. He did not give any indication of the performance of his method.

It would not be practical to use Yoshiura's method to interpret drawings that already exist which have notes on them. This approach would throw away all the already existing notes and relies on the expert user to supply them. It would be more appropriate, therefore, to use a Character Recognition Technique to recognise annotations on the drawing and store them as additional information. This makes the approach depend upon itself in accessing this information rather than expect the expert user to enter them

Chapter 3

Low Level Processing of Engineering Drawings

3.1 Introduction

This chapter deals with the recognition of straight line and curved segments. The lines on engineering drawings can be classified as either outlines or interpretation (convention) lines. Outlines define the visible (solid) and hidden (dashed) border lines of an object. Hidden lines are outlines which are obscured by a face of the object. Interpretation lines are used to define dimensions and auxiliary information. Other aspect of convention lines is the indication of the position of a feature on the object. These maybe centre (dashed-dot) lines and pitch circles (dashed-dot) indicating the centre of a *feature* or leader lines associating a phrase with a particular part on the drawing, Figure 3.1. The identification and classification of these lines is vital to the interpretation process.

The reconstruction of 3-D models can only correlate the information of the outlines on the projections. On the other hand interpretation lines assist the reader to give a complete description of parts of the outline. The work in this thesis, attempts to use textual information as well as these interpretation lines to achieve a consistent and complete outline of the object on the drawing. Almost all engineering drawings contain interpretation lines, it is a convention which engineering drawings cannot do without. An engineering drawing in itself is a convention which describes a physical object on paper. Thus, scaling the object to fit on paper, requires the specification of the scaling factor, or the dimensioning of an individual side of the object.

Interpretation lines are very important in understanding drawings. Textual information may replace parts of the geometry in describing the object on drawing which also have a great contribution in understanding the drawings. This chapter describes algorithms for identifying the projections of a drawing, and recognising the different types of lines. The use of textual information and convention lines are explored in the next two chapters.

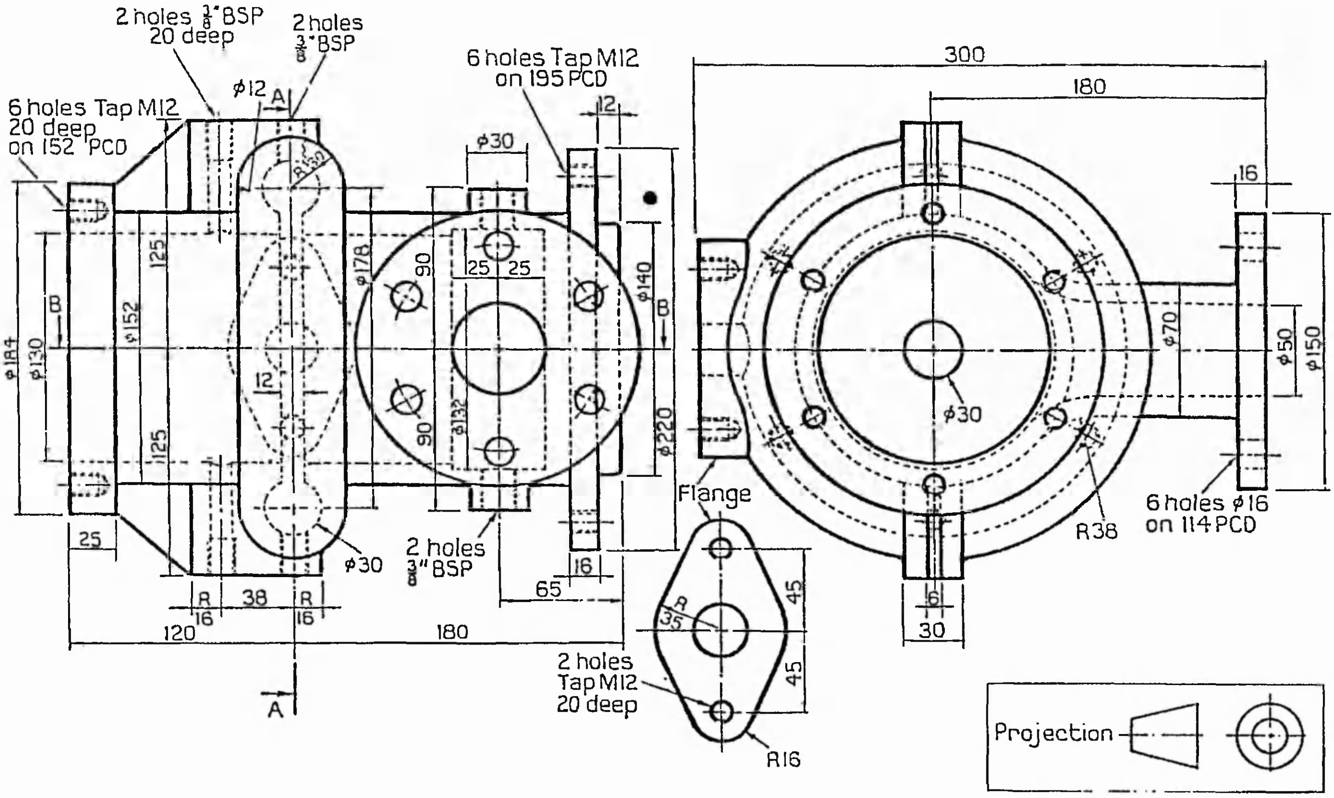


Figure 3.1: Example of Engineering Drawing (taken from [Hewitt75]).

3.2 Scanning and Vectorisation of Engineering Drawings

The conversion of an engineering drawing on paper to a suitable computer format is accomplished by a scanner that accurately measures the density characteristics of each individual picture element and converts it to a digital compressed data file. Scanning of a drawing creates an image of several million black and white pixels that represent the original drawing with varying accuracy, depending (among other things) on the optics of the scanner and the quality of the original drawing. However, this bitmap image is not directly suitable for processing by a CAD system, which operate with structures such as lines and curves as well as text. Therefore, many commercial systems have been developed to convert bitmap images to vector representations suitable for vector editing called vectorisation [Joseph92], [Manesh90], [Nagasamy90], [Vaxivier92].

These vector representations are more suitable to work with and require considerably smaller memory than its pixel representations. Each vector represents a group of pixels, and is described by the two end points and line type. And a number of these vectors may be constituted into a different type of lines, this therefore, identifies visible and hidden lines of the outlines and interpretation lines. This process of line type identification is described in section 3.6.

3.3 Correcting the Geometric information for Scanning Errors

Imperfections may occur due to the *Scanning* or *Vectorising* of a drawing on paper. These processes may produce different type of erroneous data which may produce an ambiguous 3-D model. Therefore, some measures are needed in order to rectify these errors. In section 3.3 an algorithm for global rotation and stretching of a drawing has been described which reduces the chance of errors in other low level processing such as views labeling and line identification.

Scanning and vectorising of the engineering drawing have been produced using software developed at The Nottingham Trent University. A scanner will read every thing on the drawing and convert the information into a bit map image. The format of this image is not a suitable representation for drawing interpretation. Therefore, the image is vectorised and lines and arcs are fitted where possible.

Some imperfections may occur during scanning and vectorising the drawings. For example rotation and stretching affect the whole drawing and need to be corrected globally before further processing. The recognition process starts by correcting for any rotation of the image. Obviously, if the paper drawing was not correctly aligned during scanning the resulting data may be rotated relative to its proper orientation. It is therefore, essential to correct this defect before identifying the views as even small inaccuracies can interfere with the recognition of geometric information within each view.

Algorithms developed by the author are described in the following sections. Section 3.5 describes a method to identify and label the views of engineering drawings. Section 3.6 describes a method which groups the segments that may represent a single line and in section 3.7, line types are identified which will distinguish between outlines and convention lines. Other information recognised is dimension lines which has been described in section 3.8. Section 3.9 deals with vectors which at this stage of the interpretation process has no apparent relationship to the geometric information. Finally in this chapter, all the geometric information recognised are presented in a suitable format for further processing as described in section 3.10.

3.4 Global Rotation and Stretching of Drawings

The first stage in processing the vector data file is to check whether the drawing has been rotated, an algorithm which was first described [Shaw95]. Rotation of the drawing occurs due to misalignment of the paper drawing during scanning. This rotation would interfere with the recognition process which looks for geometric information that are lined up. Therefore, it is necessary as a first step to detect any rotation and correct it. The way to correct for rotation is to find the longest approximately horizontal line. This has been achieved by finding a node (x,y) of an outline in the Front view and a corresponding node (x,y) of an outline in the side view. Such that the y values of the two nodes are approximately equal and the distance between them is the longest, as shown in Figure 3.2. Using the angle between this line and the horizontal axis, the drawing can be corrected for rotation.

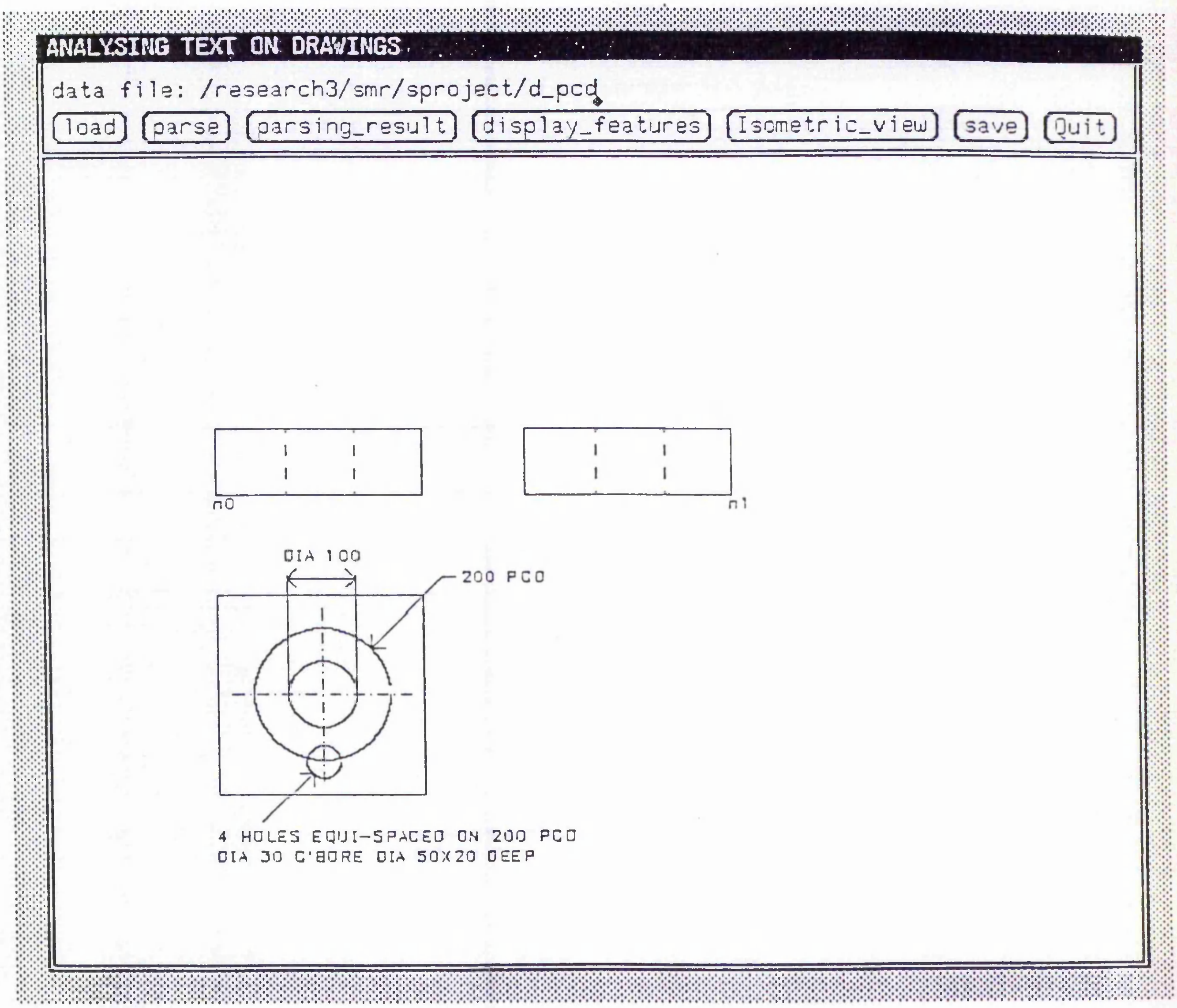


Figure 3.2: Scanned image of an Engineering Drawing, the line between n0 and n1 is the longest approximately horizontal line.

Correcting for stretching is done after the views and the line types are identified but it is appropriate to describe it here since stretching may occur due to scanning of the original paper drawing. Stretching can be detected by comparing two different representations of the width of the object. The width is measured vertically in the Top view and horizontally in the Side view, Figure 3.3. Any discrepancy between these two lengths indicates a stretch. A stretching value can be obtained from these two lengths and the whole drawing is corrected accordingly. The stretching value is the ratio of the horizontal length of the Side view over the vertical length of the Front view, equation (i). Obviously, if the stretching value equal to 1.0, the drawing is not stretched and processing the *y* co-ordinates for stretching will not change their values. Stretching value other than 1.0 means that the drawing has been stretched and therefore all the *y* co-ordinates of the drawing can be altered by multiplying by the stretching value, equation (ii).

$$\text{Stretching value} = \frac{\text{horizontal length of end elevation}}{\text{vertical length of plan elevation}} \tag{i}$$

$$\text{New y value} = (\text{Old y value}) \times (\text{Stretching value}) \tag{ii}$$

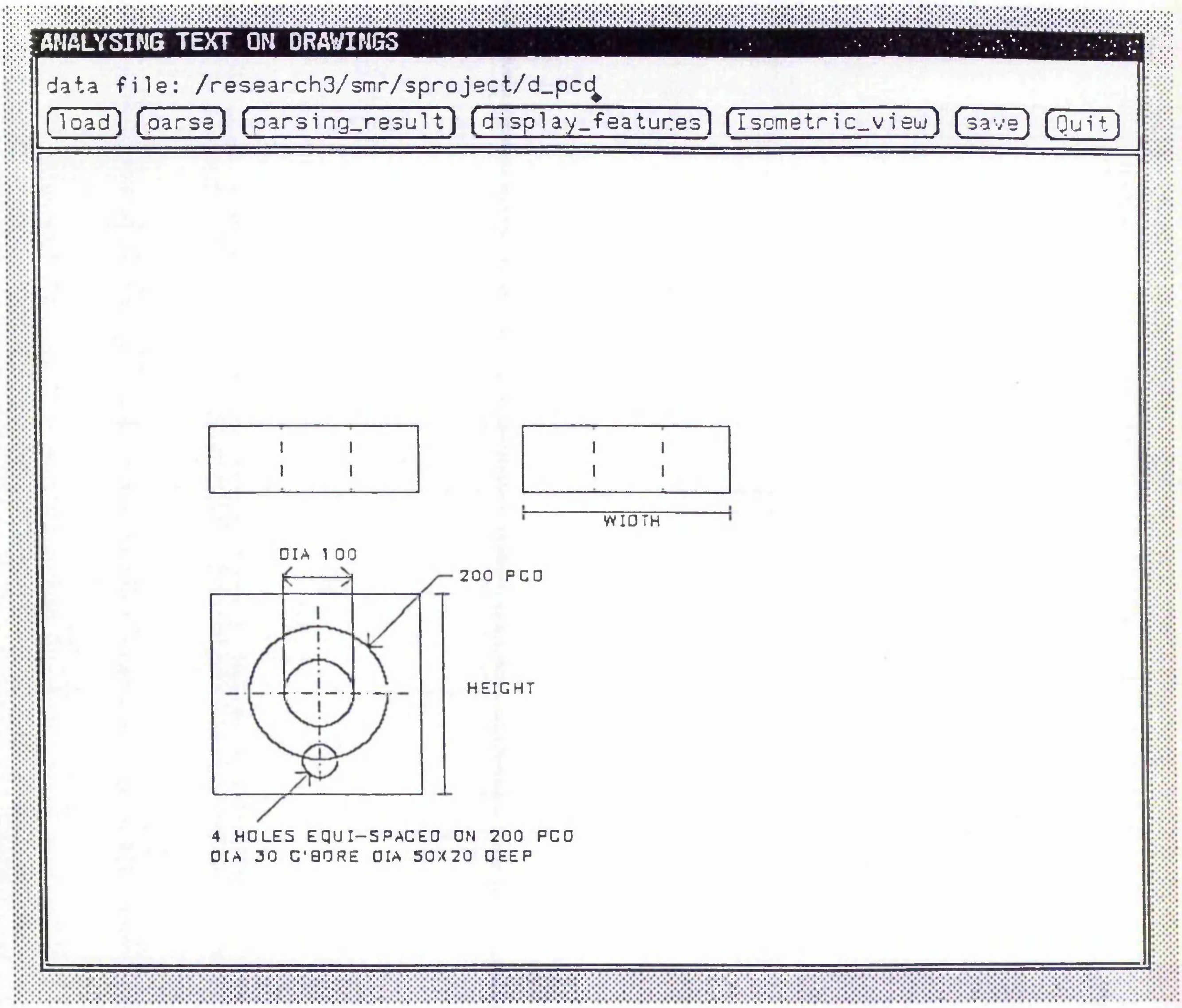


Figure 3.3: Obtaining the Stretching value from views.

3.5 Labeling Views of Engineering Drawings

An engineering drawing consists of several, usually three views. These views are normally chosen to be orthogonal, or at ninety degrees, to each other in terms of viewing direction.

The vectors produced by the vectorisation process are in no particular logical order, and no explicit information is presented regarding their connectivity. Interpretation of lines on the whole drawing may create errors in which lines from one view are considered to be in other views. Therefore, it is necessary to identify the centre point of the views to separate the vectors into groups, each group pertaining to one of the three views. Textual information and dimensioning must also associated with the view to which they belong. Line types are not identified at this stage. Thus, hidden and centre lines are represented by many short segments and some of the solid outlines may also be segmented during the scanning process.

In order to separate the views, a frequency histogram for all line segments on the drawing was calculated. The region represented by the minimum number of segments on the histogram is determined. Imposing a histogram along the x-axis determines the x co-ordinate of the centre of views, in the specified region, as shown in Figure 3.4. Similarly a frequency histogram for line segments along the y-axis yield another region which can determine the y co-ordinate of the centre of views. Using these centre co-ordinates, the drawing can be divided into four quadrants, three of which contain the projections while the fourth may contain a title box or be empty. The algorithms developed in this thesis based on the first-angle projection, although a third-angle projection has the same views but are positioned differently on the drawing. Therefore, the same principle can be used for labeling views in third-angle projection. The next stage is to process each view individually.

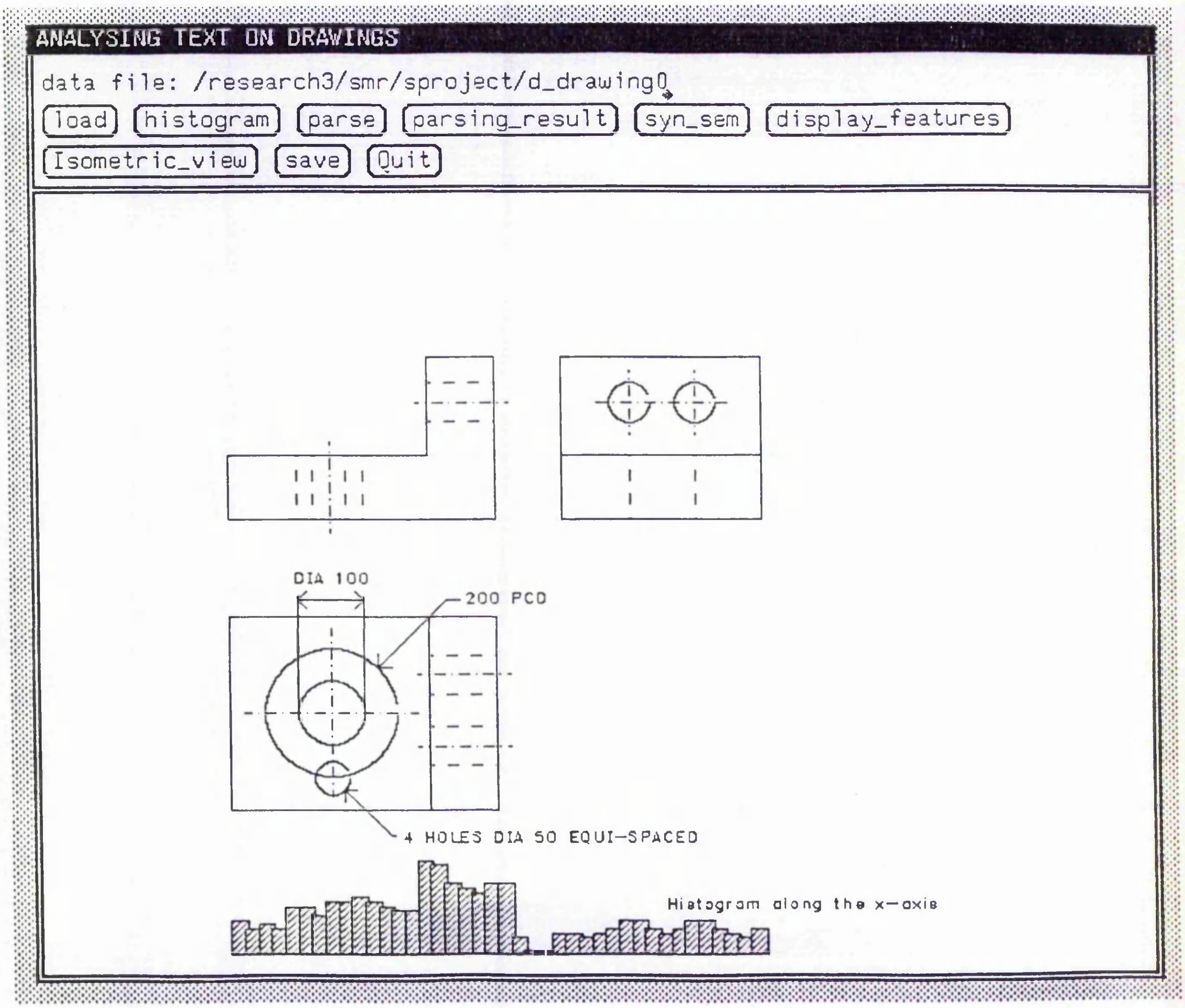


Figure 3.4: Frequency histogram of line segments showing the region of separation of the projections.

3.6 Grouping of Line Segments

In a particular view, all nearly vertical and all nearly horizontal line segments are grouped. The ends of these line segments have an equal or approximately equal common coordinate (common x-coordinate for vertical line segments and common y-coordinate for horizontal line segments). Furthermore a group of continuous vertical or horizontal line segments which have an equal or approximately equal common coordinate value are identified as collinear lines. Thus, all horizontal and vertical line segments are first clustered together and sorted in order. Each collinear group is then split at this stage, based on only one condition: If the separation between the two adjacent nodes of neighbouring lines is greater than some threshold value, then the group must be split into two. Groups containing single line segments are identified to represent solid lines. Other groups extracted from the initial one with the condition set above represent broken lines. Figure 3.5 shows the vectors data representation of the dashed line segments (A, B, C and D) as well as the vertical lines of the two square holes of Figure 3.6.

View	X1	Y1	X2	Y2	Length	Separation between adjacent nodes	
2	314.00	488.00	314.00	492.00	4	3	
2	314.00	495.00	314.00	498.00	3	3	
2	313.00	501.00	314.00	505.00	4	4	
2	314.00	509.00	314.00	513.00	4	3	
2	314.00	516.00	313.00	520.00	4	3	
2	314.00	523.00	314.00	527.00	4	36 ←	Split group here
2	314.00	563.00	313.00	635.00	73	152	
2	349.00	487.00	350.00	490.00	3	3 ←	Split group here
2	350.00	493.00	350.00	497.00	4	4	
2	350.00	501.00	351.00	504.00	4	3	
2	350.00	507.00	350.00	510.00	3	4	
2	349.00	514.00	350.00	518.00	4	3	
2	350.00	522.00	350.00	526.00	4	72 ←	Split group here
2	349.00	562.00	350.00	634.00	73	225	
2	522.00	489.00	522.00	493.00	4	4 ←	Split group here
2	521.00	497.00	522.00	501.00	4	4	
2	522.00	505.00	521.00	509.00	4	3	
2	522.00	512.00	522.00	515.00	3	4	
2	521.00	519.00	522.00	522.00	3	4	
2	522.00	523.00	521.00	528.00	3	36 ←	Split group here
2	522.00	562.00	522.00	634.00	72	55	
2	558.00	487.00	559.00	491.00	4	4 ←	Split group here
2	557.00	495.00	558.00	499.00	4	3	
2	558.00	502.00	558.00	505.00	3	4	
2	558.00	509.00	557.00	513.00	4	4	
2	558.00	517.00	558.00	521.00	4	3	
2	557.00	524.00	558.00	527.00	3	36	Split group here
2	557.00	563.00	557.00	634.00	72	45 ←	

Figure 3.5: Shows the positions of separating the groups.

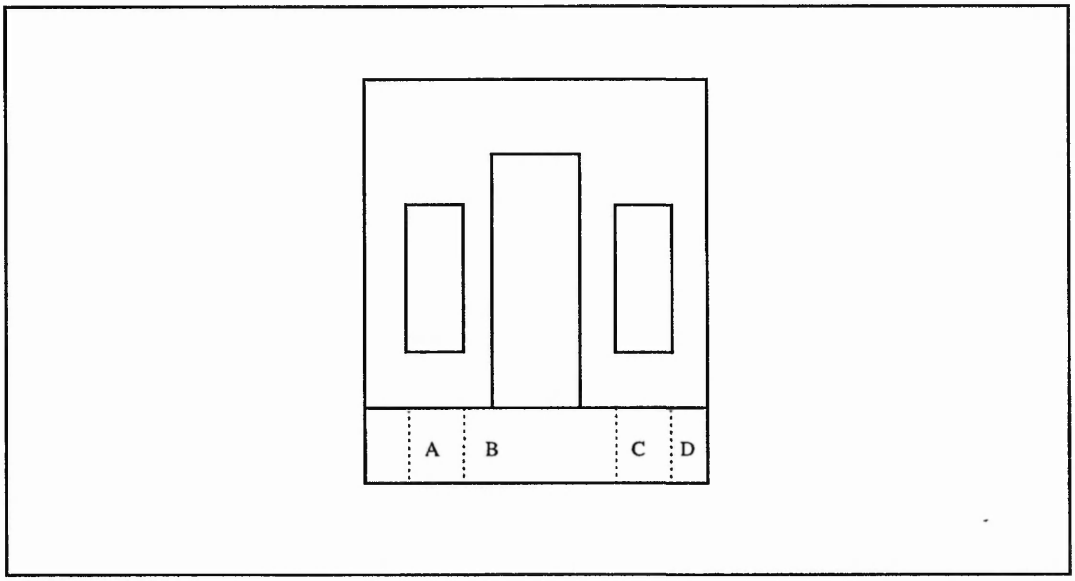


Figure 3.6: Grouping of hidden line segments A, B, C and D.

Extracting other groups that might represent dashed or centre lines, require additional conditions. Dashed lines groups are extracted by testing the collinear groups for the average separation and the average lengths between neighbouring lines. If the average separation is approximately equal to one separation and the average length of line segments within a group is approximately equal to an individual length, then the group is considered to represent a single hidden line as will be explained in the next section. Further, if the average separation of adjacent line segments within a group is approximately equal to an individual separation and the average length is not approximately equal to an individual length, this group is tested for centre line.

Groups should consist of lines which have approximately equal values for their separation. If the gap between two lines is much greater than the average separation, they most likely belong to different groups.

The group splitting routine is implemented sequentially. Since lines are stored as structured records, therefore, groups are stored having a root line pointing to the lines which are contained within that group. Figure 3.7 gives a step by step example of how a group might be split. The example is based on the data of Figure 3.5 which represent the hidden lines (A, B, C and D) of the view in Figure 3.6.

Lines that are not belong to any groups are considered to be individual visible lines and given the appropriate line identifier. Figure 3.8 shows the data from a drawing and line grouping which has occurred as a result of applying the rules stated above.

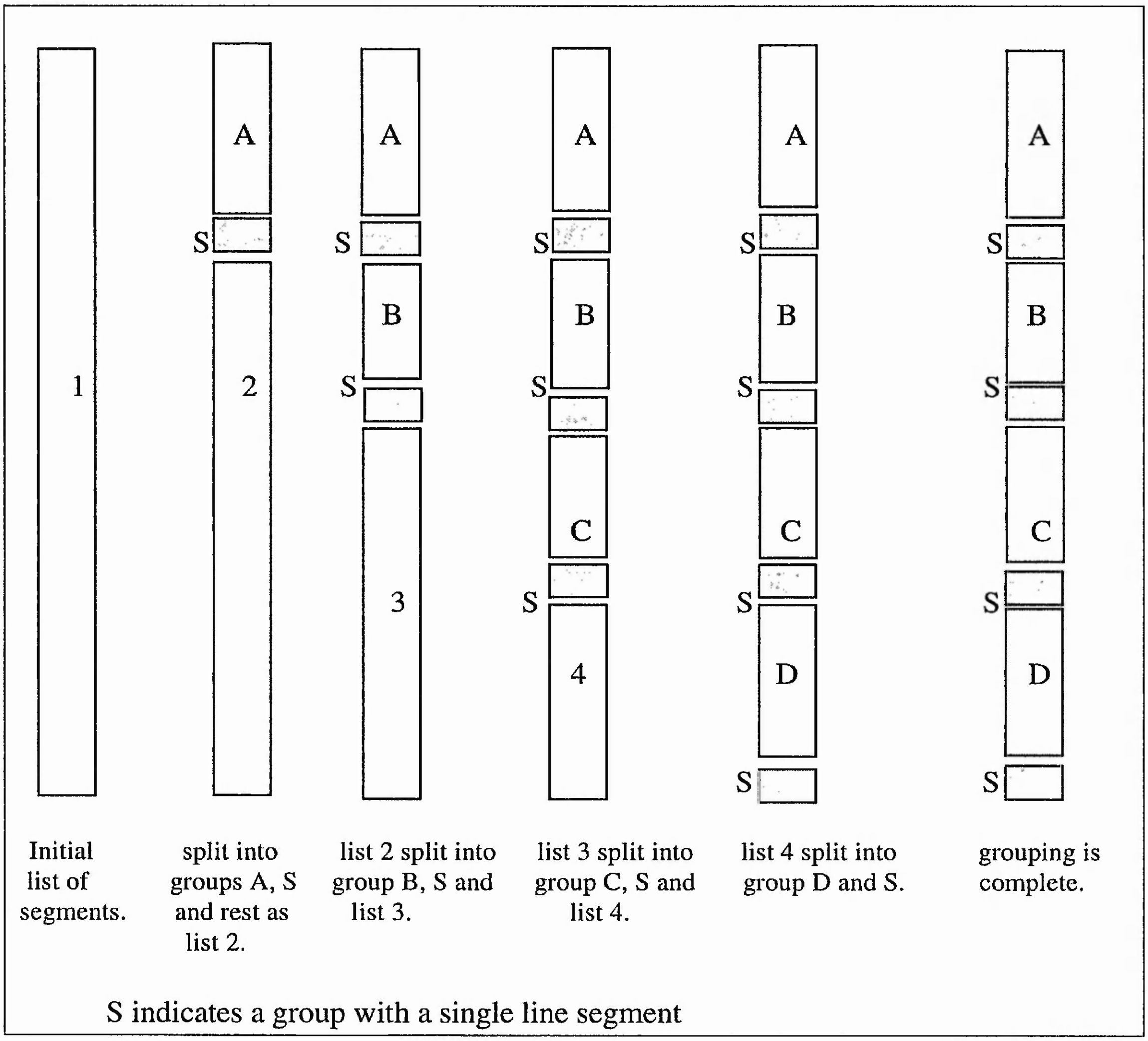


Figure 3.7: Grouping Algorithm.

View	X1	Y1	X2	Y2	Length	Separation between adjacent nodes	
2	314.00	488.00	314.00	492.00	4	3	Group A
2	314.00	495.00	314.00	498.00	3	3	
2	313.00	501.00	314.00	505.00	4	4	
2	314.00	509.00	314.00	513.00	4	3	
2	314.00	516.00	313.00	520.00	4	3	
2	314.00	523.00	314.00	527.00	4	36	
2	314.00	563.00	313.00	635.00	73	152	Solid Line
2	349.00	487.00	350.00	490.00	3	3	Group B
2	350.00	493.00	350.00	497.00	4	4	
2	350.00	501.00	351.00	504.00	4	3	
2	350.00	507.00	350.00	510.00	3	4	
2	349.00	514.00	350.00	518.00	4	3	
2	350.00	522.00	350.00	526.00	4	72	
2	349.00	562.00	350.00	634.00	73	225	Solid Line
2	522.00	489.00	522.00	493.00	4	4	Group C
2	521.00	497.00	522.00	501.00	4	4	
2	522.00	505.00	521.00	509.00	4	3	
2	522.00	512.00	522.00	515.00	3	4	
2	521.00	519.00	522.00	522.00	3	4	
2	522.00	523.00	521.00	528.00	3	36	
2	522.00	562.00	522.00	634.00	72	55	Solid Line
2	558.00	487.00	559.00	491.00	4	4	Group D
2	557.00	495.00	558.00	499.00	4	3	
2	558.00	502.00	558.00	505.00	3	4	
2	558.00	509.00	557.00	513.00	4	4	
2	558.00	517.00	558.00	521.00	4	3	
2	557.00	524.00	558.00	527.00	3	36	
2	557.00	563.00	557.00	634.00	72	45	Solid Line

Figure 3.8: Completed group splitting results.

3.7 Classification of Line Segments

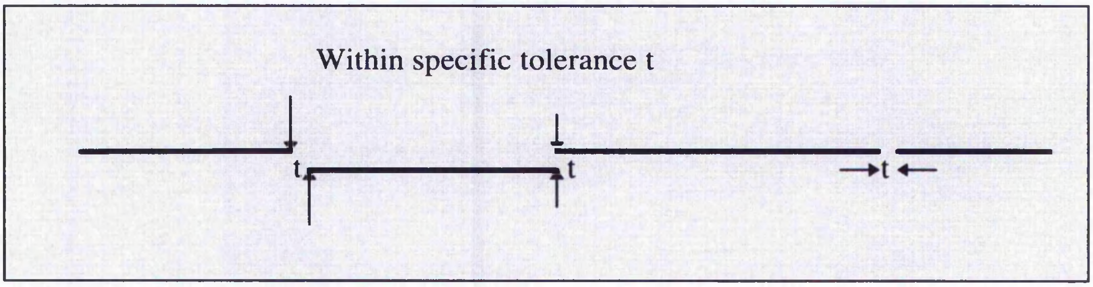
Line segments within each of the identified group were sorted in ascending order starting with the minimum y co-ordinate for vertical and the minimum x co-ordinate for horizontal line segments. The purpose of this sorting, is to test the identified groups for one of the following classifications:

(a) - Line segments which are approximately collinear and their lengths are not equal and have very small gaps between them (broken Visible lines), as in Figure 3.9a.

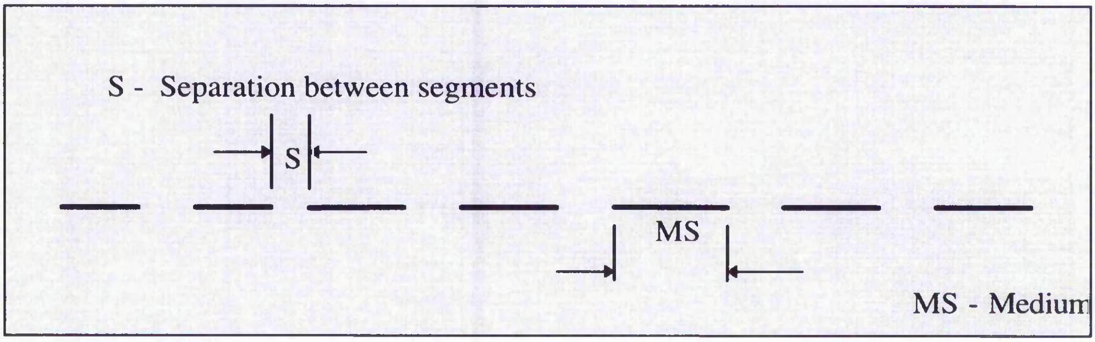
(b) - Line segments which are approximately collinear satisfying the conditions of approximately equal lengths and at an approximately equal separations from one another (Hidden lines), as in Figure 3.9b.

(c) - Line segments which are approximately collinear satisfying the conditions of approximately equal separations and the average length is not equal individual length (Centre lines), as in Figure 3.9c.

Line segments are not distinguished by either the density or the width in this work. It might be the case in general where convention lines are drawn thinner than outlines as described in [Nagasamy90], [Vaxiviere92]. The line recognition method implemented in this work totally depends upon segment lengths, and the separation between segments as well as the collinearity of these segments.

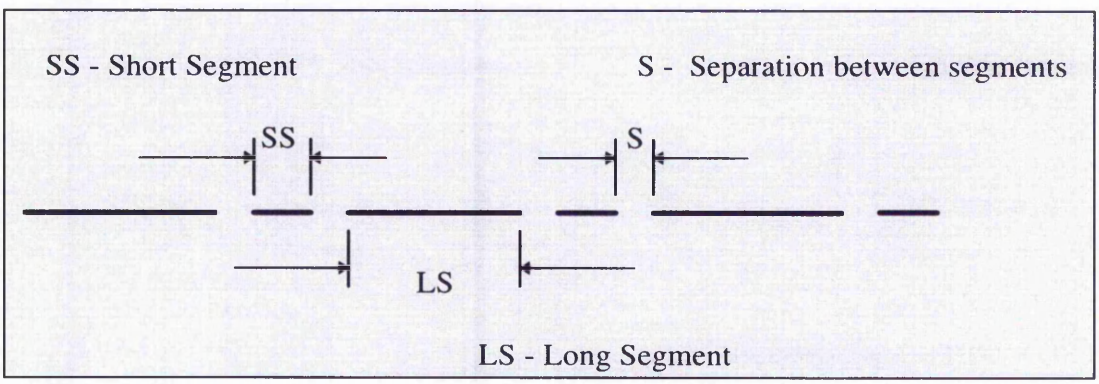


(a) - Joining segmented outlines of the drawing.



Segment

(b) - Identification of hidden line in a group.



(c) - Identification of centre line in a group.

Figure 3.9: Illustration of the identification of line types.

3.7.1 Hidden Line Segments

The identification of segments that might be part of "Hidden" line begins by the identifier picking a group containing segments that are approximately have equal length values to that of the average length of all the segments in the group. Also the separation between consecutive segments should have an approximately equal separation values to that of the average separation of all the segments in the group. If these two conditions (1) and (2) are met, the set of all segments replaced by a single line entity with the attribute "Hidden" line.

$$\text{Average Length} = \frac{\sum_{i=1}^{k} \text{Segment's Length}}{k} \qquad (1)$$

for (i = 1, ... , Number of segments in the group(k)).

$$\text{Average Distance} = \frac{\sum_{i=1}^{k-1} (\text{Distance Between Segments})}{(k-1)} \qquad (2)$$

for (i = 1, ..., Number of separations between segments(k-1)).

The co-ordinates of this hidden line will be assigned the extreme co-ordinates of the end segments of the group. Most vertical and horizontal hidden lines can be found using this approach. The geometric information is constantly reduced in size every time a line type is recognised.

3.7.2 Centre Line Segments

Similar approach to that of identifying of "Hidden" lines has been used to recognise "Centre" lines. The difference in this case is that, there are two different sizes of segments that can be part of a "Centre" line, long segment and short segment. Identifying any group containing a sequence of segments having two different lengths would initialise the recognition procedure to test the segments for the condition that the length of each segment in the group, be either approximately equal to the average of all short segments or approximately equal to the average of all long segments in the group equations (3) and (4). Other condition required to be satisfy is the separation between the segments which should be approximately equal to the average separation of all the segments in the group, equation (2).

Long and short segments are separated by starting with the initial segment in the group, and search the rest of the segments in the group for approximately equal length segment and add to the list. Any segment that are not approximately equal to the initial segment, is added to another list. A comparison between an individual segment from each list can identifies which list contains the long segments.

$$\text{Individual Segment} \approx \frac{\sum_{i=1}^{l} \text{Long Segment}}{l} \qquad (3)$$

for (i = 1, ... Number of long segments in the group).

and;

$$\text{Individual Segment} \approx \frac{\sum_{i=1}^{m} \text{Short Segment}}{m} \qquad (4)$$

for (i =1, ... Number of short segments in the group).

Initially, segments that are collinear and within very small tolerance are tested for broken lines and if this is satisfied then they are jointed to represent a single line with attribute "Visible". This special case needs to be treated with care as dimension lines close to the outlines can be wrongly connected.

Identifying any one of the classes (a-c) of Figure 3.9 will replace the whole group by a single line entity with the appropriate attribute (visible, hidden or centre lines). Centre lines are very important because they symbolise symmetric lines on which many features such as holes are centered.

The problem of hidden lines detection is not only concerned with engineering drawings but it is a common problem in many computerised document analysis applications [Kasturi90], [Agam95].

3.8 Oblique Line Segments

Line segments that are meant to represent hidden or centre lines but are at an oblique angle are grouped by testing for the collinearity and gradient. By recognising vertical and horizontal dashed lines, this can give clear indication to the size of the segment that comprised in the dashed lines. This value can be used as threshold to test the remaining segments which might represent a dashed lines but are at an oblique angle. The test for collinearity between two lines involves the calculation of the angle between them as follows:

$$\text{Cos } A = \frac{\mathbf{a1.a2}}{|\mathbf{a1}||\mathbf{a2}|}$$

Where **a1** and **a2** are the vectors representing the lines and **A** is the angle between these lines. If **A** is less than small threshold value, then further tests need to be applied for collinearity. First a test for the comparison between their gradients is calculated which should be less than a small threshold value. Finally, lines are considered to be collinear if the line joining their centre points has approximately the same gradient as that of the lines.

$$g1 = \frac{y_{a12} - y_{a11}}{x_{a12} - x_{a11}} \qquad g2 = \frac{y_{a22} - y_{a21}}{x_{a22} - x_{a21}}$$

and; $$| g1 - g2 | < \text{Threshold}$$

Where **g1** and **g2** are the gradients of the vectors representing the lines and **Threshold** is very small value. To compensate for any errors occurred on the vectors due to the scanning and vectorisation processes, a small threshold value of 0.4 has been used.

Once grouping the segments is complete, a test similar to that of vertical or horizontal dashed lines can be applied to identify the whole oblique dashed line shown in Figure 3.10.

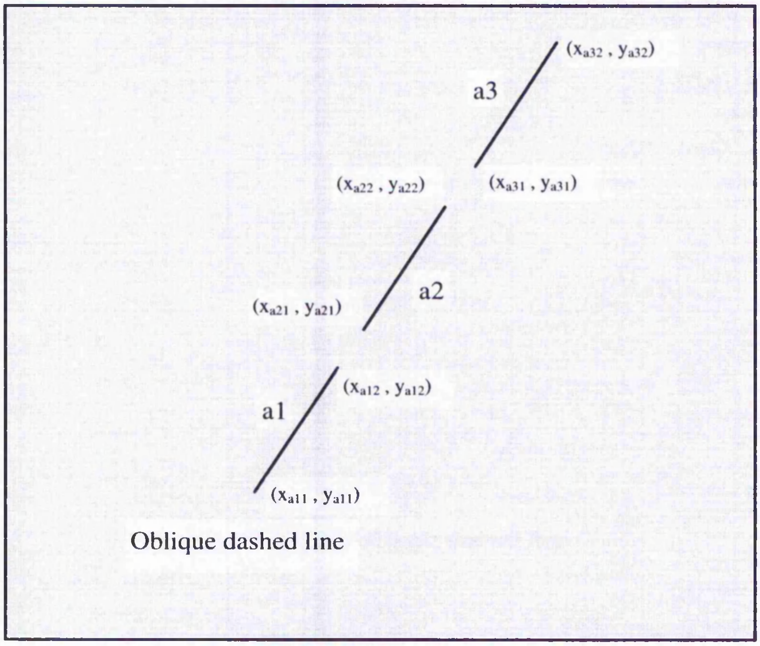


Figure 3.10: Illustrates the oblique segments representing a hidden line.

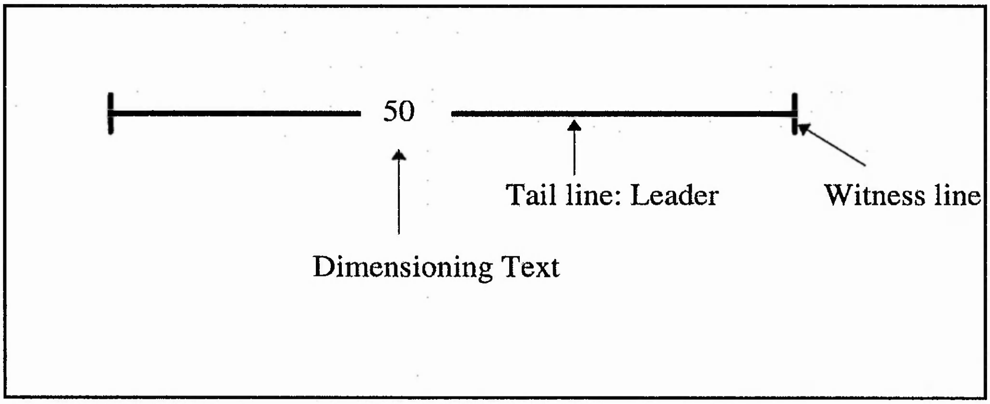
3.9 Dimension Outlines

In order to interpret engineering drawings, it is necessary to separate text from graphics. There are many algorithms described in the literature [Fletcher88], [Whal28] for text-graphics separation. The method described in [Fletcher88] is for omni-directional text-graphics segmentation. A new rule-based text/graphics separation algorithm and a model-based procedure for detecting arrowheads in any orientation have been developed in [Lai94]. The algorithm uses arrowhead tracking and search methods to extract leaders, tails and witness lines from segmented images containing only text. Dimension text extracted from the segmented images are then associated with their corresponding leaders to obtain complete dimension sets.

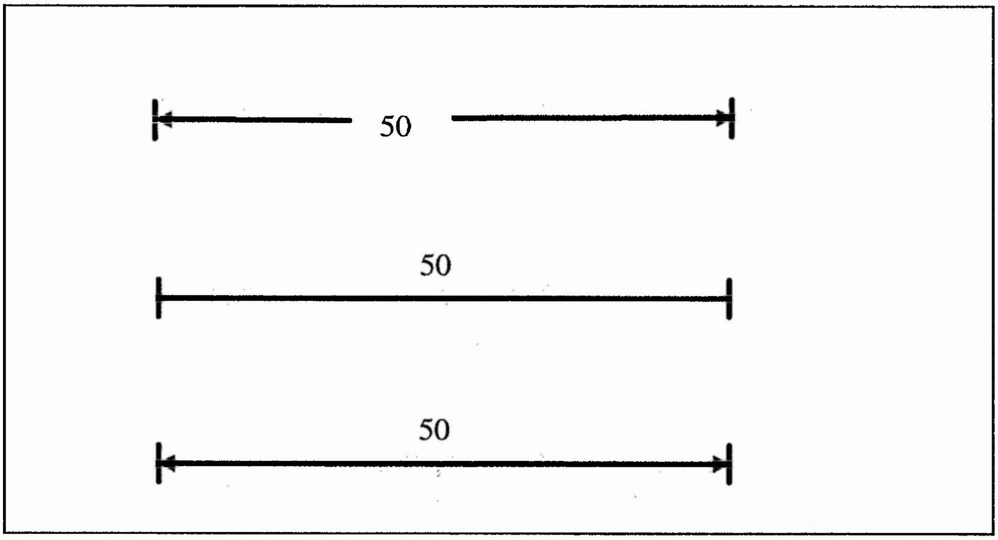
Chai and Dori [Chai92], proposed an algorithm for textbox extraction, that is preceded by Orthogonal Zig-Zig (OZZ) vectorisation, arc segmentation, and arrowhead recognition. The textbox extraction is done by clustering the remaining short lines that are close to each other through a region growing process. This algorithm is designed only for detection of text areas without string extraction, and may lead to some detection errors. Later D. Dori and Yelena Velkovitch [Dori95a], described a sub-system for dimension text recognition from engineering drawings produced according to either ANSI or ISO drafting standard, which is integrated into the Machine Drawing Understanding System (MDUS). The complete recognition is performed beginning with text segmentation and ending with verification of OCR results.

There are also a number of other work which has been reported to use the dimensions to determine line length [Gopin79], [Hillyard78], [Sutherland77], [Requicha77].

For linear dimensioning, it is relatively easy to locate dimensioning lines (leader and witness lines), when dimensioning text is known Figure 3.11. Thus, the position of numbers representing lengths of outlines on drawing, indicate that dimensioning lines must be very close to the text position. For further simplification of drawing information, dimension lines are removed once the co-ordinates associated with dimensioning text are found. However, these can easily be put back on the drawing. Dimensioning text, end co-ordinates of dimension and a dimension identifier are stored away as the information describing the precise size of the outlines of the object on the drawing.



Case (a) - The dimensioning text "50" lies between dimension lines .



Case (b) - Show three other ways of representing a horizontal dimension.

Figure 3.11: Some representations of horizontal dimensions.

3.10 Residual Information

The information presented by the vectorising process so far has been categorised and each category is stored in a separate linked list for further processing. Thus, we have identified the necessary information needed to describe the drawing. Furthermore, the remaining information (in particular, line segments) is stored in a linked list (residue). This list may be checked in cases where ambiguous line situations occur.

Because the way line segments are connected to other lines which share a common node, the aim is therefore, to examine all line segments that are not connected at both ends. A threshold value should be specified for the length at which line segments can be considered as redundant information.

The procedure therefore, looks for the line segments that are unconnected at their ends with other segments and smaller than the specified threshold. These segments are then tested if they belong to other segments. This can be achieved by checking the separation between these line segments and neighbouring segments. If the separation between them fall within a specific threshold, then they can be combined to represent a single segment. Other segments that do not satisfy this condition can be assumed to be redundant information and placed into the residual list.

These small segments maybe the result of some dirt or crease on the paper drawing which the scanner has read during scanning. This therefore, can be distinguished from the drawing information, since these segments have no apparent relationship with the drawing data. Larger creases and other extraneous lines are found using their connection with the geometric information on the drawing, algorithm described in [Shaw95].

3.11 Suitable Representation of Drawing Information

All the entities identified in the previous section (lines, dimensions and so on), need to be presented in a suitable format for a CAD system. Any one of these entities is represented by a type identifier followed by numerical entries specifying its geometry. The number of these numerical entries in each entity varies depending on the type of the entity and best described by these entries. A line is best described by two sets of co-ordinates representing its ends, in addition an entry for line thickness and an entry to which a line could be one of the following type; visible, hidden or centre lines. Circles and circular arcs have two entries for the centre co-ordinates, one for radius and two for start and end angles as well as entries for the thickness and types. Linear dimensions have an entry for dimension length and two sets of co-ordinates representing the start and end of the length.

Views are described by an identifier and an integer entry representing the number of view. As mentioned previously, first-angle projection has been used, where Front view will be represented as V0 which situated in the top left corner, Top view will be represented as V1 which situated in the bottom left corner and Side view will be as V2 and situated in the top right corner, Figure 3.12 shows the geometric data along side the view in Figure 3.6.

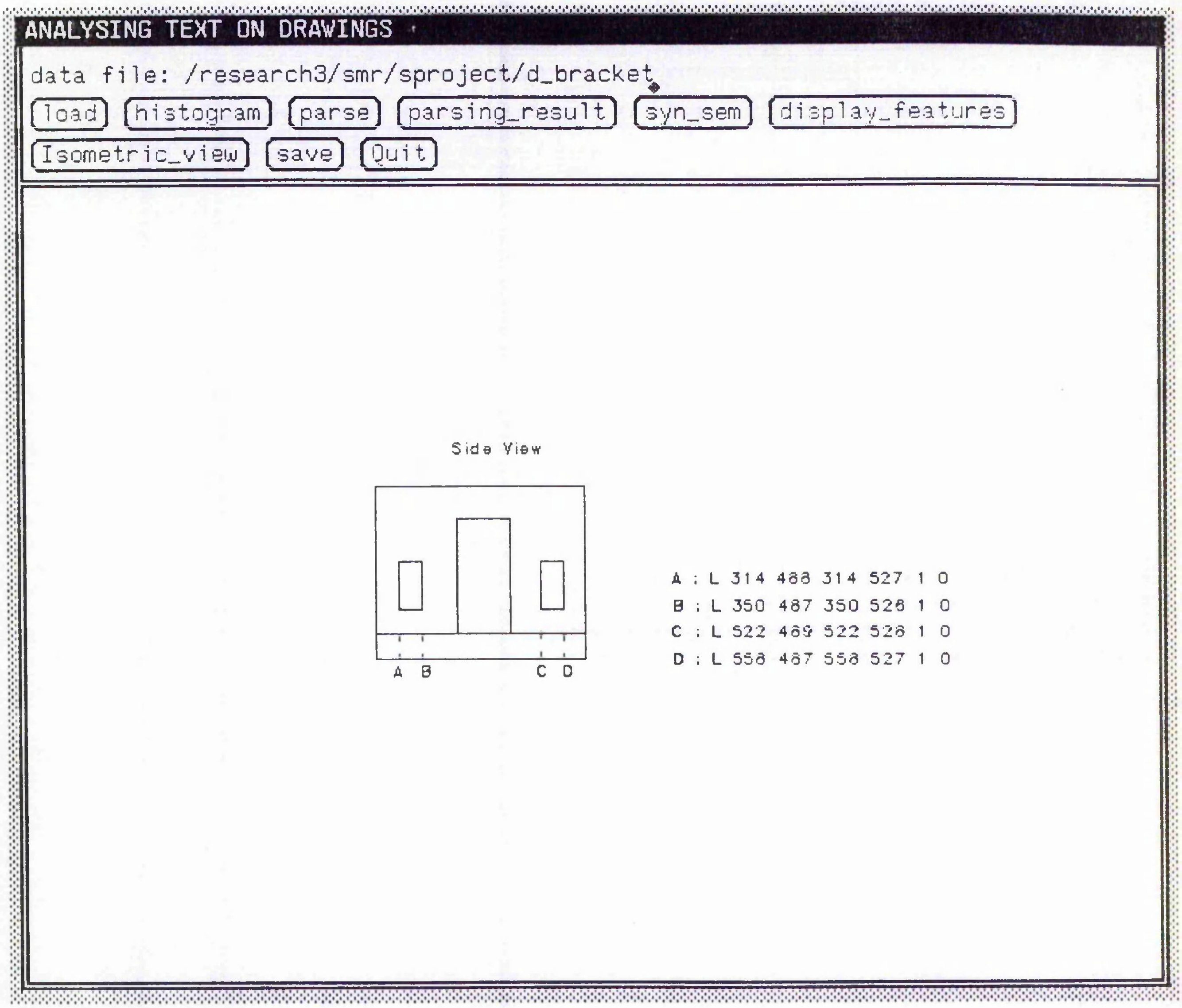


Figure 3.12: Part (side view) of Engineering Drawing with geometric details of hidden lines.

3.12 Summary

The ultimate goal of many researchers and designers is the automatic reconstruction of 3-D objects from a paper drawing. The processes of scanning and vectorising of a paper engineering drawing provide a representation of the original paper drawing on computer. This chapter has reported methods of using this computer representation to identify the main elements of the drawing.

An algorithm using vectors frequency histogram has been used to identify the orthographic views of the drawing. Each group of vectors resulted from this method has been subjected to further recognition of the geometric entities describing the object. Groups of collinear vectors have been recognised to be visible, hidden or centre lines which are then removed and replaced by a single line and a line type.

No attempts have been made by this work to recognise textual information. Therefore, dimensioning text and textual annotations have been added to the data representation. But the positions of dimensioning text have been used to recognise axial dimension lines which are then removed from the data representation.

The main area described in this chapter is the recognition of the geometric information of the data representing the engineering drawing. Unrecognised data were stored in a residual data file which can then be used in cases of ambiguities. This has another advantage which is the cleaning of extraneous data occur due to dirt or crease on paper drawing during scanning.

Chapter 4

High Level Processing of Engineering Drawings

4.1 Introduction

Textual information on an engineering drawing is regarded as high level information because it carries technical information about the object. On the other hand outlines are regarded as low level information. The high level information is used by the interpreter to understand the drawings, while low level information is used to recognise them. Although much information about the object in an engineering drawing is carried by the outlines, it is incomplete without dimensioning and annotations. Text not only provides the dimensions and tolerances of a part, but also the machining reference surfaces. When we are dealing with the complete engineering drawing, we must consider everything on the drawing, dimensioning and annotations as well as outlines.

Some text describes features which are also shown in the outlines, whereas other text provides information not represented in any other way such as machining process, tolerance and so on. Text is important because it usually provides information not available in any other way. Sometimes it adds to information already available, for example by providing a dimension of a *feature* shown in the outline. Other *features* described by text may be omitted partially or totally by the designer or may have been distorted or lost during scanning. Some textual information does not help from the recognition sense, but some text is very useful and can improve efficiency or in many cases achieve things impossible without it.

4.2 Text Information: Dimensioning and Annotations

Dimensioning in engineering drawings provides an exact definition of the geometry which is represented approximately by the geometric outline information. Therefore, recognition of dimensions [Dori90], [Lai94] and association with the geometry is essential to the interpretation process. It has been described in [Dori90] how the basic building blocks of dimensioning are the dimensioning text, arrowheads, leader, and witness which together constitute a single dimension-set. A collection of dimension-sets is used to accurately and completely define the geometry of each view. Based on both explicit and implicit conventions of the major drafting standards, it has been shown in [Hewitt75] a view of an object is properly dimensioned if and only if its dimensioning completely and non-redundantly defines its geometry. The main motivation for the proper dimensioning requirement by the standards is that it compels the designer to specify those dimensions in the product which are the most important from a functional point of view. Proper dimensioning thus minimises the number of specified dimensions, which, in turn, reduces the possible occurrence of contradictions, especially when tolerances are involved.

Dimensioning and annotations constitute the text, the recognition of either of which will accompany dimensioning or annotation lines that add extra information about the object. Some annotation lines accompanying text, for example as a leader line, indicates the part of the drawing that is associated with the text. This information is very important to the interpretation of textual information. For instance if a phrase refers to a *feature*, then the leader line will indicate the position of the *feature* on the drawing. For example in the case of a hole then leader line will either indicate a point on the edge of a circle or points at the intersection point of the centre line with the end of the cylinder representing the side projection of the hole. In Figure 4.1 Front View, the phrase "2 *HOLES M10 - 6H x 12 DEEP*", has leader line pointing to circumference (Top view of hole), whereas phrase "2 *HOLES M10 - 6H*" pointing to the middle intersecting point of rectangle (side view of hole). Therefore, the leader lines are used to associate with the text a particular position on the drawing. This position is important to the interpretation of the text as opposed to the position where the text is written which merely depends on where there is space to put it.

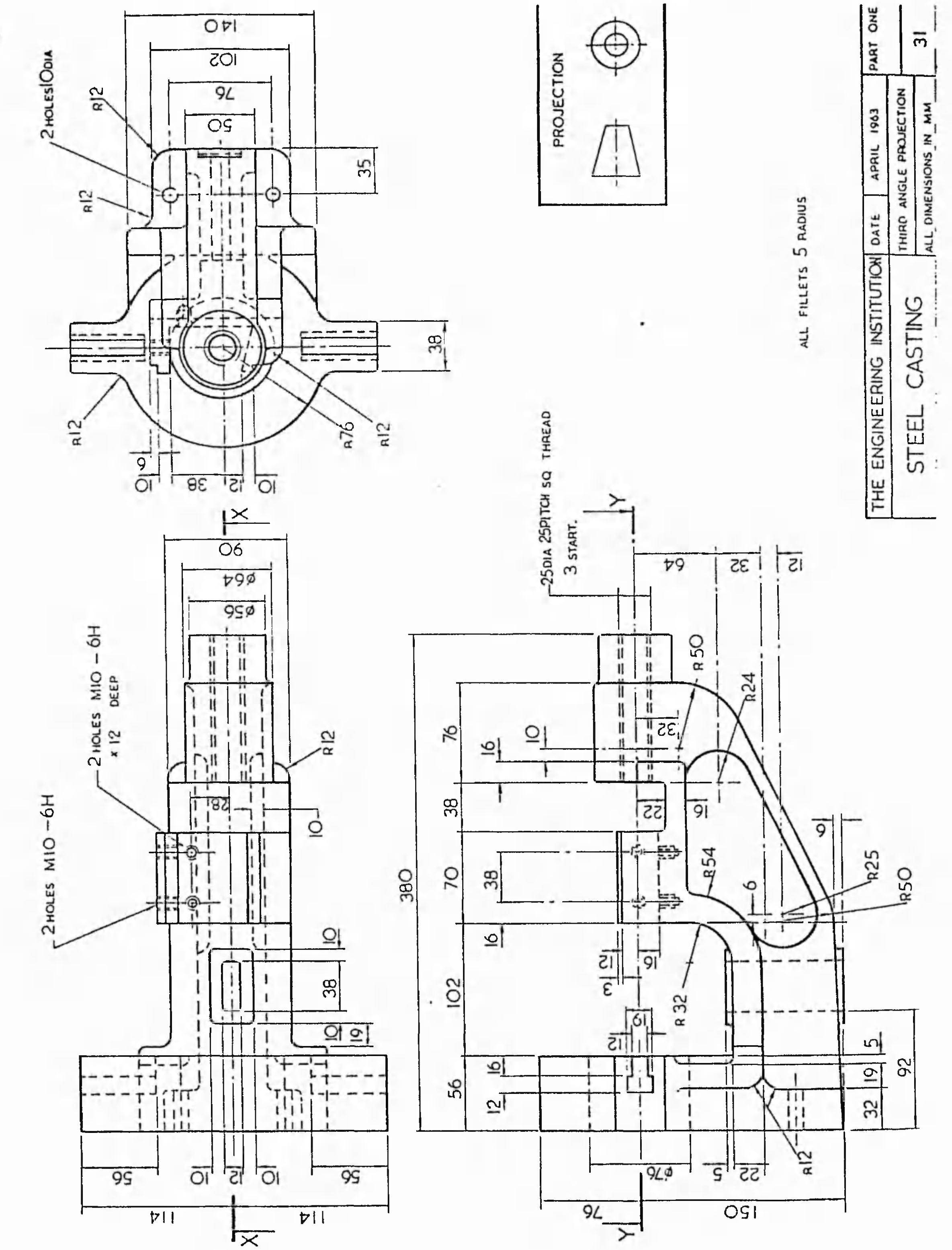


Figure 4.1: An example of an Engineering Drawing showing a number of common phrases. Many linear and radial dimensions, text in a title box and a number of annotations referring to holes (taken from [Hart75]).

4.3 Textual Information for 3-D Reconstruction

Many technical drawings contain annotations which describe a simplified part of the object in one of the projections. Some of these comments are very useful to the reconstruction process, and may not only save a lot of processing time, but most importantly help produce a consistent 3-D model from the original 2-D drawing.

Hiroshi Yoshiura [Yoshiura84] has treated and interpreted complete sentences on a drawing, but in fact most text on drawings is in the form of phrases rather than complete sentences. His method also required an expert user to enter these sentences.

Annotations describing *features* simplified by the designer are most important to this work and also *features* that may have been distorted by pre-processing the geometric data.

Figure 4.2 shows the overall process of interpreting the low and high levels of engineering drawing information. Figures 4.3 and 4.4 show the respective processes of recognising and understanding of engineering drawings. The automatic interpretation of engineering drawings can be achieved by integrating these two processes, Figure 4.5 shows an overview of the method.

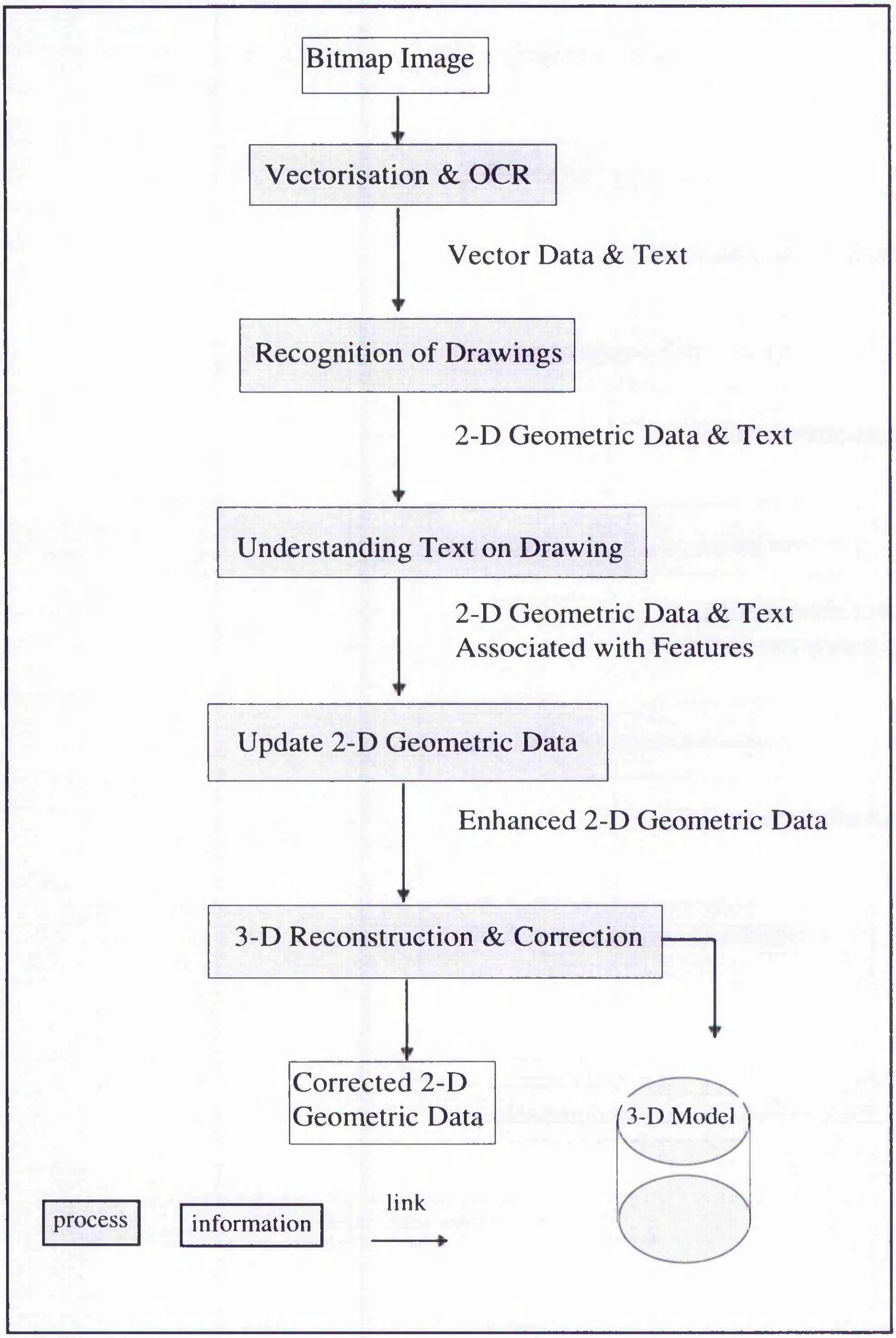


Figure 4.2: Overview of Processes of Recognition (line interpretation) and Understanding (text interpretation) of Engineering Drawings.

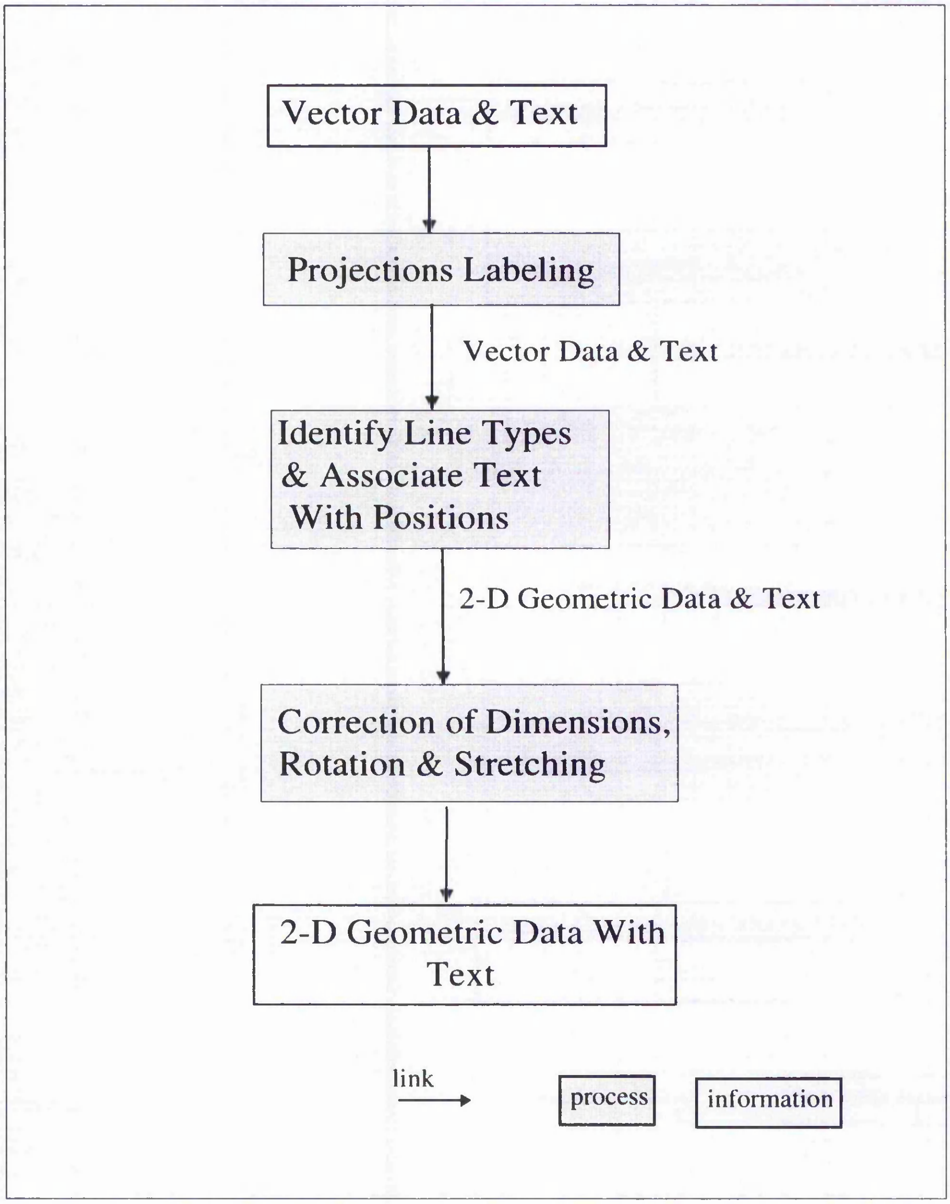


Figure 4.3: Overview of the recognition process.

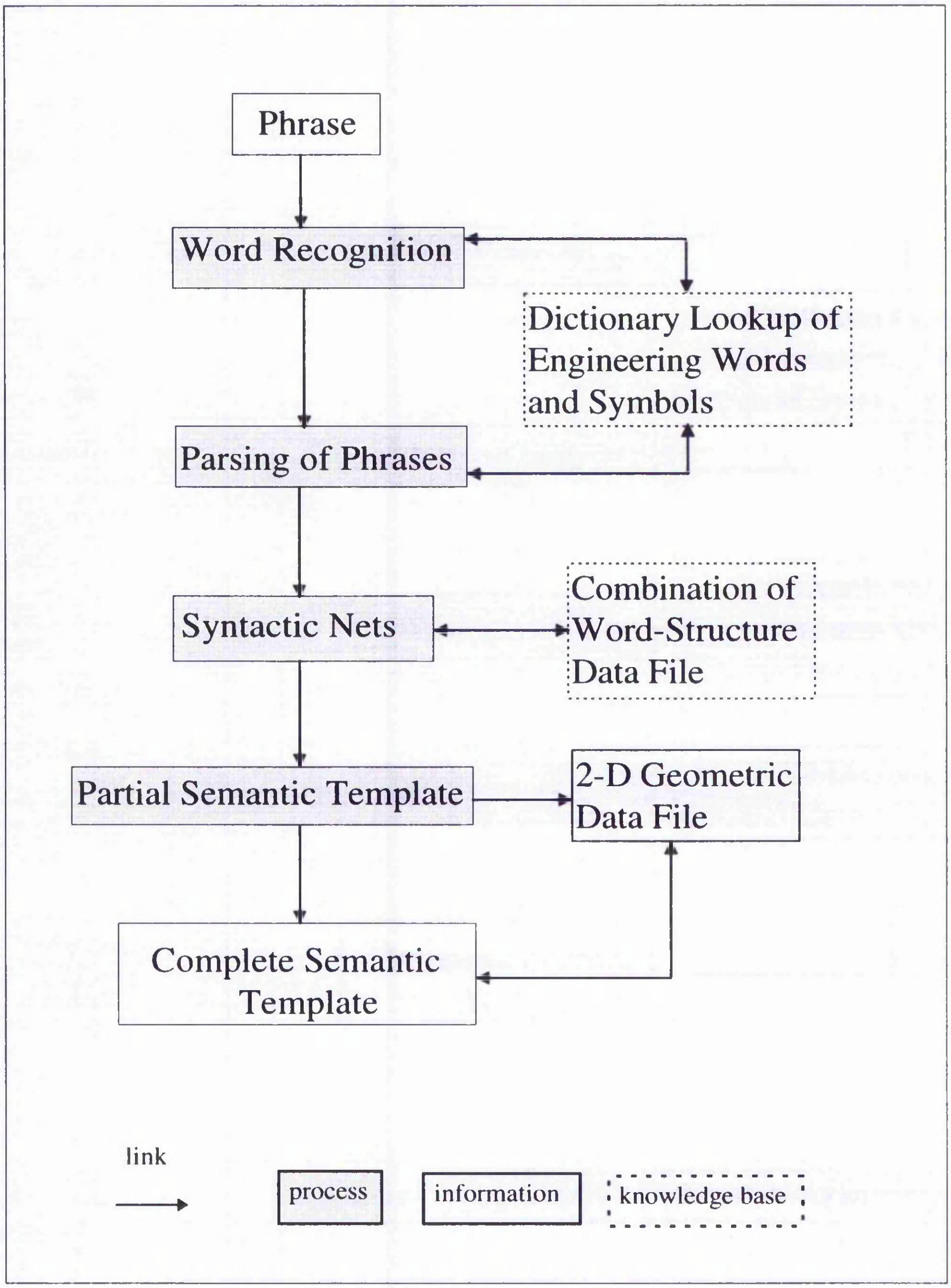


Figure 4.4: Overview of the procedure for Understanding Text on Engineering Drawings.

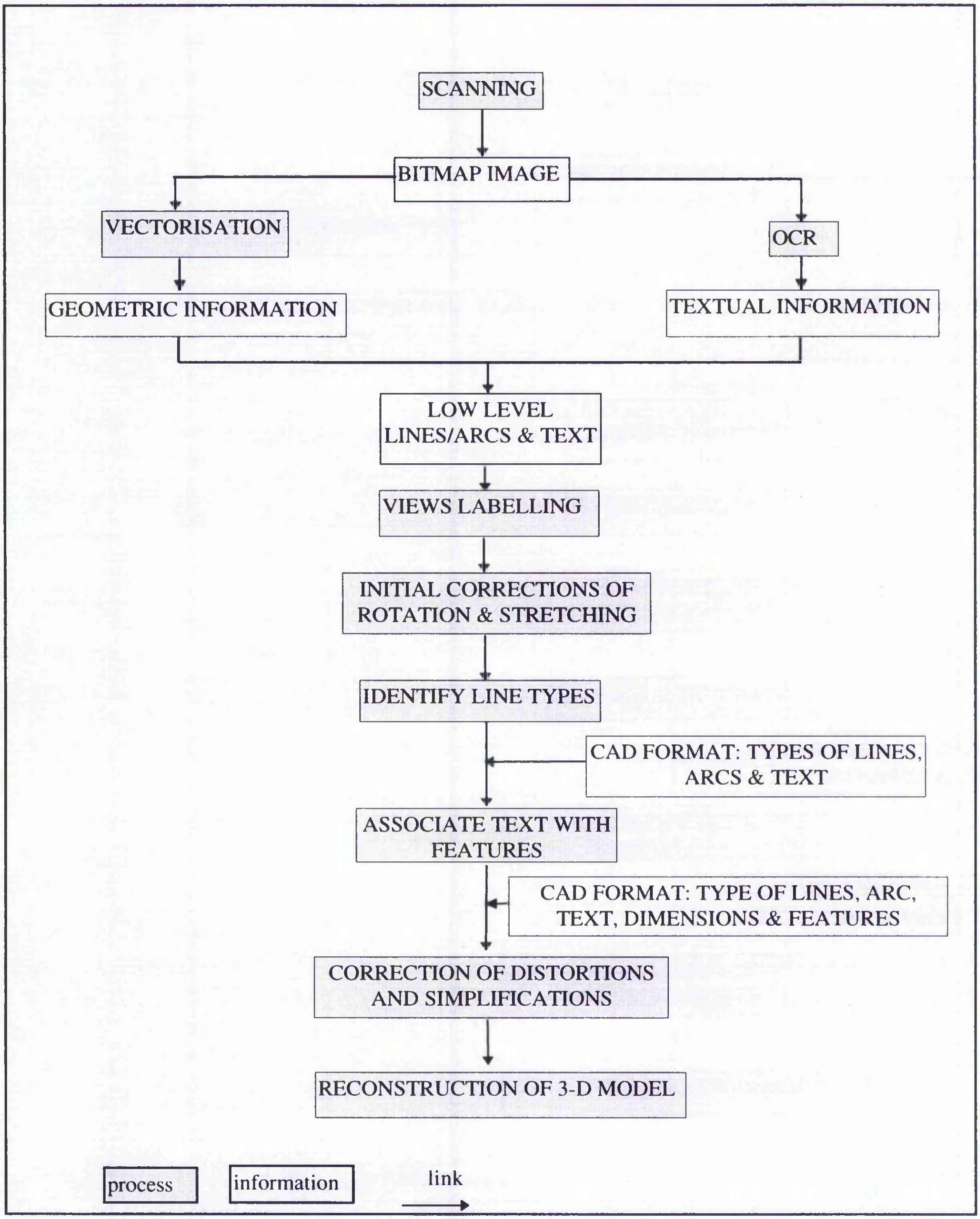


Figure 4.5: Overview of the integration of recognition and understanding of text on drawings for the automation of 3-D reconstruction.

The majority of previous researchers have used a direct (without text interpretation) bottom-up procedure in reconstructing a 3-D model [Er83], [Preiss81], [Preiss84], [Sakurai83], [Wesley80], [Wesley81]. This approach involves matching features in different views. A 2-D node (x,y) in one view is matched with 2-D nodes in the other two views, and when this is successful, the three nodes are combined into a 3-D vertex (x,y,z). Other successful matches will produce other 3-D features. Similarly, lines in one view will be matched with lines in other views, and a successful match will produce 3-D edges. Drawings that have a perfect geometric description, generally produce a consistent 3-D model. Unfortunately, this is not true for many technical drawings, where some *features* will be difficult to resolve. When the drawing has been scanned, the image produced by the scanning process may have been corrupted. For instance, small *features* may be distorted during scanning process. In other cases part of the drawing may be simplified and described by a note.

When the information obtained from the outline alone is incomplete, textual information often provides some additional clues about the *feature* described by the drawing. Therefore, an approach is needed that uses the text to provide the information omitted by the simplifications or lost during scanning process. The author has found that, interpreting the textual information on the drawing can improve the chances of the reconstruction process in producing a consistent 3-D model.

4.4 Text on Engineering Drawings

Text on drawings comprises of dimensioning and annotations. Annotations can be divided into a number of different categories depending on their relationships with other information on the drawing. Some annotations are very important to the interpretation process, such as those describing *features* of the object, "*4 Hole Dia 10*". Others may be less important and need not be included at this stage in the interpretation, such as those describing a general element on the drawing, "*Partial view in direction of A*". Obviously, the recognition of these different categories can save processing time by directly interpreting annotations which are most important to the interpretation process.

Annotation describing the geometry of the object is either an emphasis on the presence of features on drawings, such as the note "*4 Holes Dia 10",* or an indication to the precise size of the geometry of the *feature*, such as the note "*Dia 20".* On the other hand a note may indicates to a general element on the drawing. This can be either indicating to the labels of the drawings such as "*FRONT VIEW*", or indication for additional views, "*ISOMETRIC VIEW*". There are also many other general notes on the drawing such as the note, "*ALL UNDIMENSIONED RADII 5".* These different categories of text need to be recognised, the interpreter then decides on those pieces of text that can help recognise and understand the geometry of the object.

4.4.1 Text attached to Outlines (dimensioning and annotations)

Draughtsmen lay great stress upon correct dimensioning in line with the requirements of [BSI]. An engineering drawing is not complete unless it is fully dimensioned, and it is not fully dimensioned unless the item could be manufactured from the information given, without making reference to other drawings or data sheets.

It is relatively easy to ruin a well drawn, correctly projected drawing, by placing dimensions carelessly or in an incorrect style on the drawing. It is imperative that dimensions should be placed clearly and in a uniform manner, so that there should be no doubt about the size, shape and *features* of the object in question. Not all industrial organisations use the same system of dimensioning, but all are expected to conform to [BSI], in which several alternatives are approved.

Dimensioning on drawing usually are represented in three different types, linear dimensions, such as textual dimension "*380*", in the Top view of Figure 4.1. Radial dimensions are associated with curved (circular arc) segments, "*R50*", in the Top view of Figure 4.1. In some drawings angular dimensions are associated with line segments which are at an angle.

Annotations attached to an outline are mainly assertive notes, which are an alternative way of describing the geometry or *features* on the drawing. These are very important for 3-D reconstruction, where errors due to scanning or simplifications of *features* may lead the reconstruction process to inconsistent 3-D model. The annotations can then be used to correct the distortions and may recover any simplified *features*.

Annotations on the drawings frequently refer to holes, as in Figure 4.1, but many others refer to boss, slot, pitch circle and section line. Together they complete the description of the object on the drawing.

4.4.2 Text not attached to Outlines (general notes)

There are many other phrases which are not associated with the outlines of the object on the drawing. These phrases are declarative, which may be useful in declaring sections or parts on the drawing. For example notes for labels of the projections, *"Top View", "Front View", "End* View", others may indicate the presence of details views such as *"Sectional View", "Half Section", "Partial Views"," Auxiliary View*", and so on. The interpretation of these annotations may identify additional information by recognising the additional views. These additional information can then be extracted and utilised to clarify the complex parts of the geometric description.

4.4.3 Text in Title Block

Text in a title block includes additional information about the object. Measurements and tolerances are indicated in this block as well as the object's name or reference number, draughtsman or the manufacturer name. Information within the title block may give a general idea about the object on the drawing but not an explicit information related to the geometry of the component.

4.5 Listing of some Phrases

Generally, there are many different phrases on engineering drawings, but the majority can be grouped into categories because of some commonality between them. Phrases associated with the same *feature* can be grouped into one category. This categorisation reduces the size of the engineering domain which quickly help the interpretation process identifying the phrase type or the *feature* indicated by the phrase. Also these categories may have a general formula comprising all the possible sub-phrasal type included in each category.

In this section, a small number of common phrases are mentioned, and in the next section an example of a category is described. Table 4.1 shows some common simple phrases on engineering drawings.

Simple Phrase	*Described Feature*
R20	Radial Dimension
DIA 20 *ϕ 5 DRILL* *8 HOLES DIA 10 EQUI-SPACED* *DIA 16x19 DEEP* *HOLE ϕ 15x20 DEEP*	Normal Hole(s)
ϕ 10 C'BORE ϕ 16x8 DEEP *DRILL DIA 44 MAJOR DIA 48 DIA 14 DEEP*	Counterbore Hole
ϕ 10 S'FACE ϕ 20 *DIA 56 SPOTFACE*	Spotface Hole
ϕ 5 C'SK AT 90° TO ϕ 12 *2 HOLES 8 DIA C'SK 13 DIA*	Countersink Hole(s)
8 BOSSES ϕ 20 DRILLED ϕ 10 *BOSS TAPPED M2 - 6H*	Boss(es) Tapped Boss
SPHERE R20 *SPHERE ϕ 25*	Sphere
12 DIA SQUARE THREAD *HOLE 12 DIA SQUARE THREAD* *ϕ 20 SQUARE THREAD 5 PITCH* *ϕ 22 ϕ 6.5 PITCH RH SQ THREAD* *SQUARE THREAD ϕ 20 SINGLE START 3 PITCH* *25 DIA 25 PITCH SQ THREAD 3 START*	Screw Hole
2 HOLES TAP M8 10 DEEP	Tapped Hole(s)
ϕ 20 CORE	Nut

Table 4.1: Some simple common phrases and features associated with them.

Details of holes include information about the depth and the type of thread, phrases including the letters M and/or H refer to tapped screw hole and boss, where M indicates the nominal size of the diameter and H preceded by a digit indicates the class of the tolerance required (coarse/fine series).

There are also many general phrases on engineering drawings which may need to be interpreted in order to obtain further information about the drawing. The listing of a number of these phrases gives an indication to the different engineering drawings information which can be described by them. The interpretation of this type of phrases may also be included in future work. Table 4.2 lists some of the general phrases found on engineering drawings.

Clearly, looking at Tables 4.1, the most frequently occurring phrases are associated with holes. This has inspired the author to concentrate his interpretation on phrases describing the hole feature. The principle of text interpretation can be established with this commonly occurring *feature* which will then be extended throughout the engineering domain.

General Phrase	*Description*
PLAN VIEW *FRONT VIEW* *END VIEW*	Principal views
AUXILIARY VIEW *ISOMETRIC VIEW* *PARTIAL VIEW IN DIRECTION OF A* *SECTION ON ZZ* *SECTION XX*	Additional Views
RECTANGULAR BASE SHARP CORNERS *RADII NOT SPECIFIED TO BE TAKEN AS 3* *ALL CORNERS 5 RADII UNLESS SHOWN OTHERWISE* *ALL UNSPECIFIED RADII 3* *ALL RADII NOT NOTED TO BE 3* *ALL UNDIMENSIONED RADII 5* *SMALL RADII NOT SPECIFIED TO BE 6*	Further details of dimensions

Table 4.2: Shows some general phrases and descriptions.

4.5.1 Phrases for Holes on Engineering Drawings

In engineering drawing, holes are usually described by textual annotations. There are a number of different phrases that describe a hole on the drawings. In general, *features* either explicitly indicated by the phrase, for example "4 holes", or by some characteristics of the *feature*, for example "∅ 10", "Dia 10". A phrase showing this characteristic may also refers to *features* such as Bosses. Both the name of *feature* and some of its characteristics may be presented in one phrase. Some phrases describe the presence of several *features* on the drawing, for example "4 Holes 10 Dia Equally Spaced". Since there are many different phrases indicating the presence of the same *feature*, it is most convenient to have a general formula for all phrases representing one *feature*. For instance, a general formula for the different types of holes may have the following form;

(Number) + **Dia&Size** + **(Dia&Size)** + **(Angle_value)** + **(Hole_type)** + **(Deep&Value)** + **(ADJ)**.

This formula has been represented by a structure and the terms represent the elements of this structure, and the "+" sign represents concatenation of phrases. Thus, terms within parentheses are optional. The description of each term of this general form is described below:

(Number) : Indicates the number of repeated hole in a phrase, this is an optional value and the default value is one.

Dia&Size : Standards for the diameter of the hole, and in most phrases a value for the size of the hole also present otherwise obtained from the drawing. The second (**Dia&Size**) is optional, because it is not needed for a normal hole.

(Angle_value) : This term stands for the value of the angle at which countersink hole is defined.

(Hole_type) : it includes Spotface, Countersink, Counterbore and Normal holes.

(Deep&Value) : Optional term that indicates to the depth of the hole which also associated with a value.

(Adj) : This is an optional term which can indicate to the relationship between repeated holes, for example "equi-spaced", "inline".

The order of this format is not fixed, and as mentioned earlier any number of these terms may appear in a phrase.

The different types of holes together with their relationships can be represented by a Syntactic Net as shown in Figure 4.6. Cross section of the different types of holes is also shown in Figure 4.7, which gives additional information about the different holes.

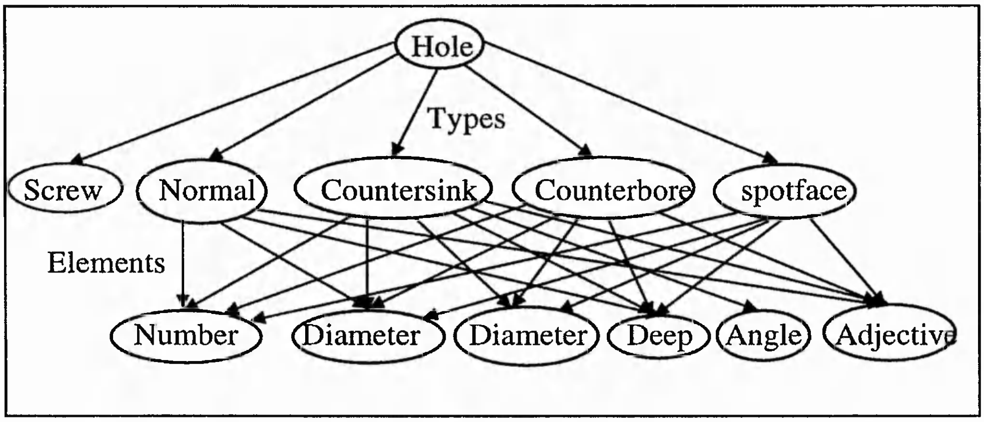


Figure 4.6: Syntactic Net for different types of Holes.

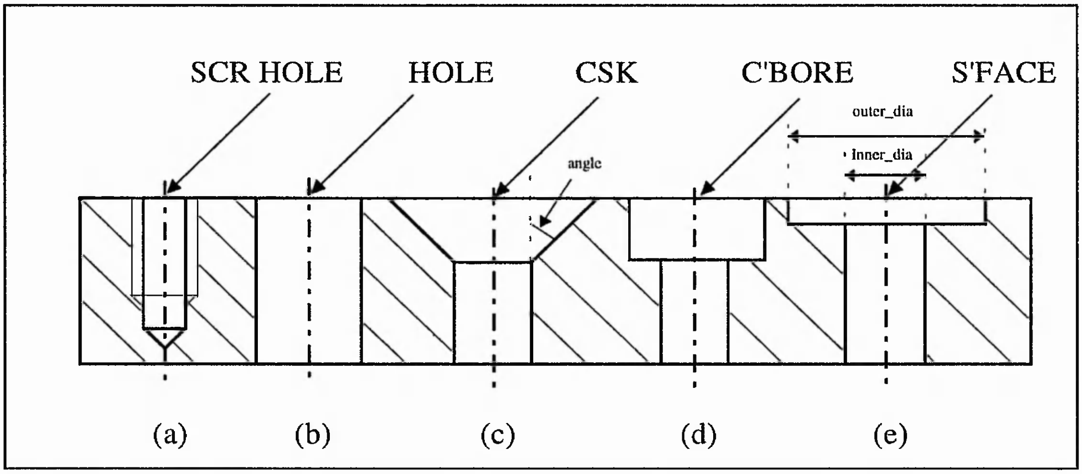


Figure 4.7: Cross section of different types of Holes: (a) Screw hole; (b) Simple hole; (c) Countersink hole; (d) Counterbore hole; (e) Spot face hole.

4.5.2 Phrases for Threads on Engineering Drawings

Other phrases may be related to threads which are used for screw and screw holes. The simplified symbol is the most frequently used and also the most easily confused with exact projection primitives. The other two types are more like a symbol than a convention. The structure of the screw hole phrase is clear and fixed and can represented as follow:

MD + ND + N + ANGLE + UNC/UNF + AB + LR.

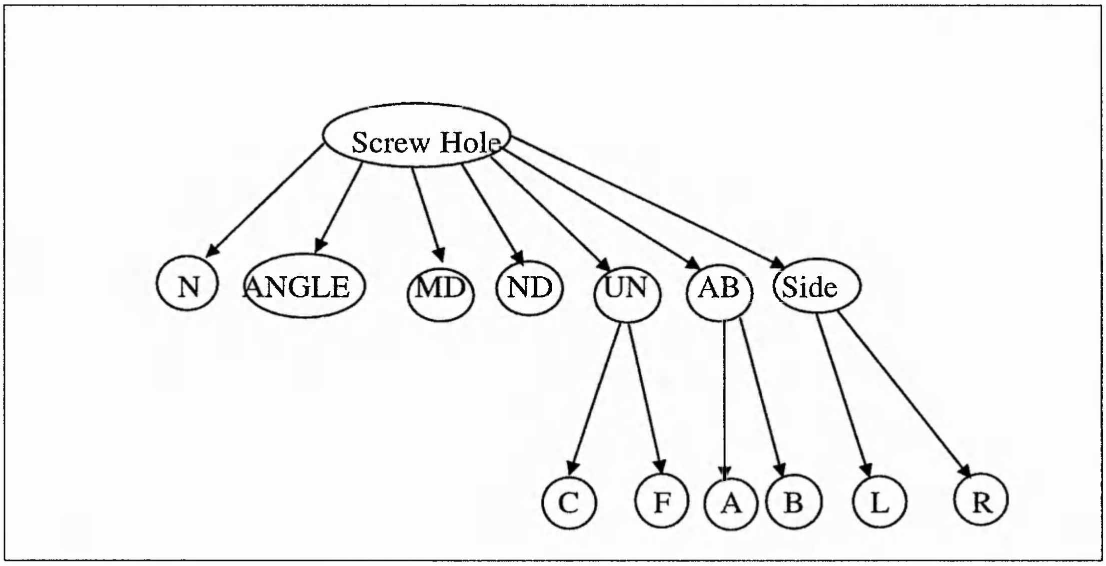


Figure 4.8: Syntactic Net for Screw type Hole. **MD** is the major diameter. **ND** is the minor diameter. **N** is the number of threads per inch. **ANGLE** at which the end of the thread runs also the angle at which the threaded portion runs. **UN** is the thread series (Unified National). **C** stands for Coarse and **F** for Fine threads. **A** is external threads and **B** is internal threads. The element **LR**, indicates it is left hand or right hand thread.

4.6 Text Analysis

In general, textual information on an engineering drawing consists of phrases not whole sentences as has been considered in [Yoshiura84]. Although processing all phrases would be a difficult problem in general, because it requires a vast amount of knowledge, our method is restricted to a specific domain, that is the *engineering* application. Resolving ambiguities and correctly interpreting abbreviations requires specific knowledge about the domain, as well as common sense. Therefore, the work described in this thesis attempts to model an aspect of human drawing interpretation for this particular application.

The interpretation process starts by splitting the phrase into separate tokens (sub-phrases, usually words or numbers), which are then stored for further processing. By matching each token in the phrase to an entry in a specially compiled engineering dictionary, the token is associated with a syntactic (phrasal) type.

The phrasal parsing block then uses the syntactic type to associate a Syntactic Net (SN) with the phrase. Next the Syntactic Net is linked to a Semantic Template (ST) to provide a partial meaning of the phrase. Finally, the projections of the drawing are searched for the *feature* described by the phrase. An overview of the method used in this work is shown in Figure 4.9.

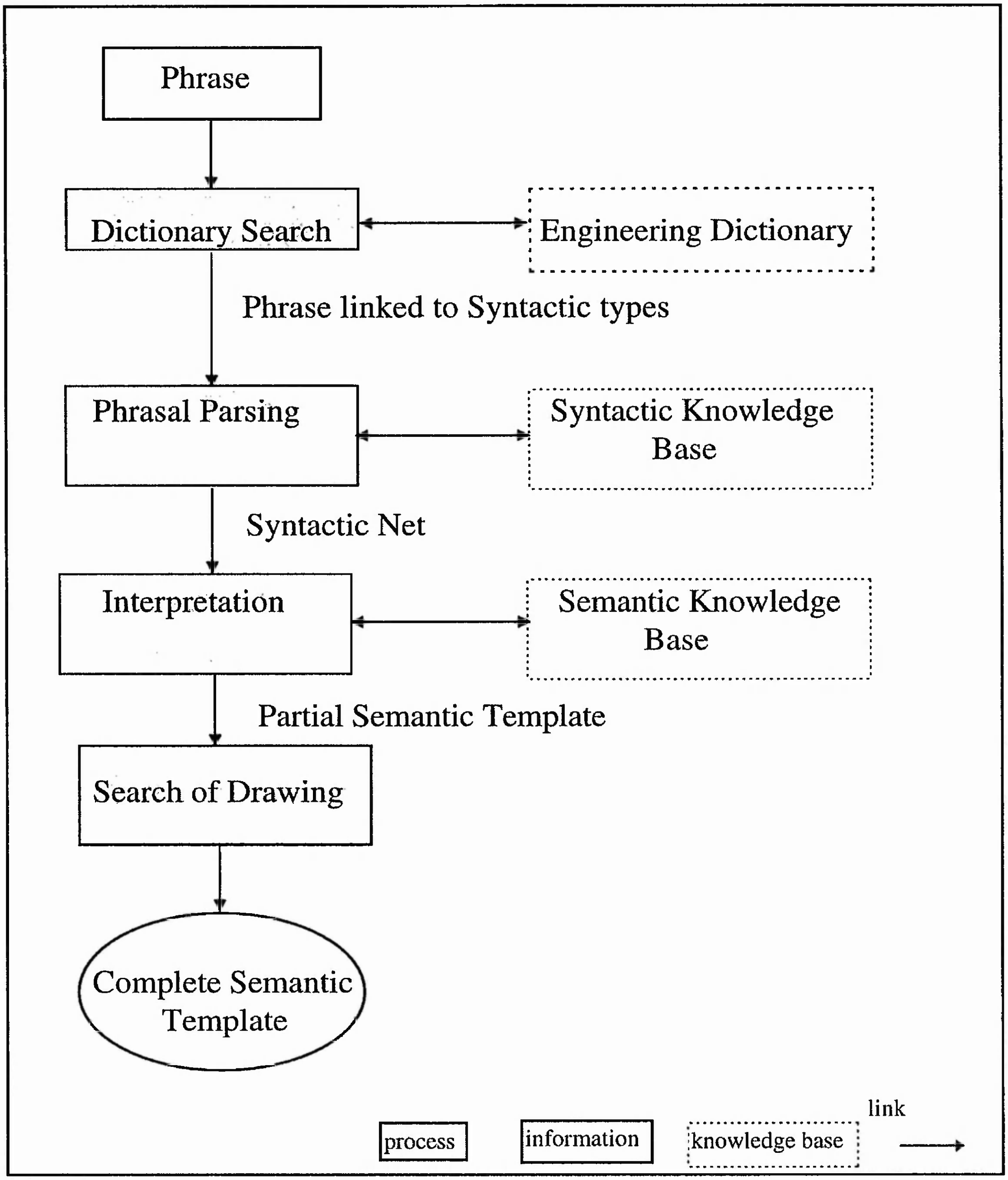


Figure 4.9: An overview of the process of parsing and interpretation of a phrase on Engineering Drawing.

The main concept of this algorithm is to break the phrase into lower items, i.e. tokens, and gradually by recognising and relating these tokens a template can be constructed representing the meaning of the phrase.

Syntactic Net and Semantic Template are the two main phases of the algorithm. In the first phase (Syntactic Net), tokens are recognised and relationships of the net are sorted, then a phrasal type is assign to it. Phrases are categorised by this phrasal type which indicates whether the phrase may be useful for the reconstruction process, and whether the phrase describing feature or general text. Depending on the phrasal type, a particular Semantic Template is created to accommodate for the information derived from the phrase.

Text interpretation described in this thesis based on phrases associated with *features* on the drawing. The principle of the interpretation can be extended to include other text on the drawing. Phrasal type can be used to categorise text associated with other text. The interpretation process can then decide on interpreting text which can be useful for the reconstruction process.

4.7 Constituency

If we look at a note, we can see clearly that it consists of larger units made up out of smaller units. These smaller units, in their turn, are made up of units that are smaller still.

These units are what we call phrases, words and letters. A note on an engineering drawing consists of a phrase, which consist of words, which consist of letters. This relationship can be diagrammed as in Figure 4.10.

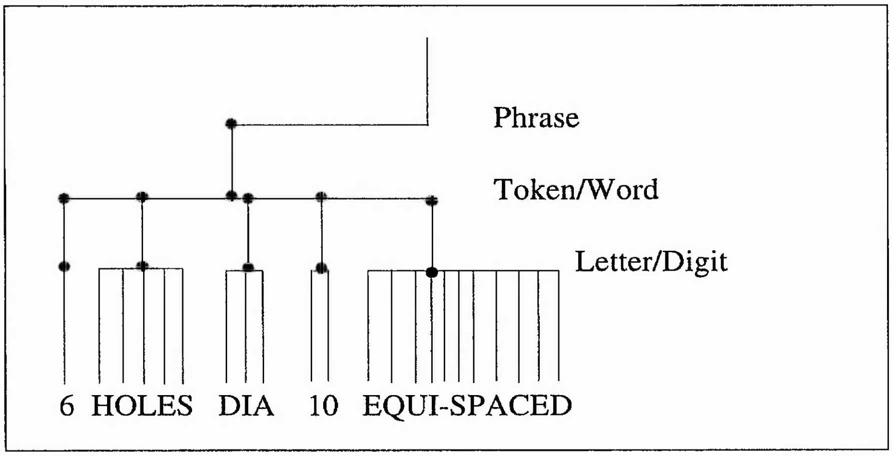


Figure 4.10: Units of Phrase.

The reason this has been diagrammed in this way because each unit begins where the previous one ends. A phrase may follow another phrase, tokens follow tokens, and letters follow letters in a simple sequence; they do not overlap, nor does anything else occur in between. The space that separate them, narrow spaces between letters, wider spaces between tokens, and still wider spaces, with may be accompanying full stop, between phrases. This kind of layered part-whole relationship which occurs among the units of text is referred as *constituency*.

4.8 Dictionary Search and Phrasal Parsing

The tokens (or sub-phrases) are stored in a doubly linked list in order to be recognised individually and to establish their relationships with other tokens in the phrase. They are then assigned syntactic types which are placed into a syntactic sequence of elements. These syntactic types are defined with respect to the domain, verbal types such as "F" for "Feature" (or "P" for a "Parameter" or "A" for an "Adjective" associated with the Feature), or numerical types, such as a "V" for "Value" or "N" for "Number". Also these elements are defined in terms of restricted syntactic categories rather than general grammatical categories. The parsing process, therefore, leads more quickly to interpretation meaningful in the domain.

It has been recognised that some text on engineering drawings is very useful, particularly phrases describing *features*, where phrases can refer to the *feature* by name. Figure 4.1 shows a number of these phrases where a hole *feature* has been described. Some phrases may describe some characteristics of a *feature* without mentioning its type for example where the phrase describes the size or tolerance of the *feature*, "*∅76*" in Figure 4.1. Knowing the type of the feature referred to by the phrase leads the interpretation process to the feature more directly. When the name of the feature is not present in the phrase, the identification of the *feature* has to be deferred until a later stage in the process.

The next few sections describe text analysis algorithms using the example phrase *"6 HOLES DIA 50 EQUI-SPACED*".

4.9 Syntactic Net (SN)

A Syntactic Net (SN) is a set of element nodes attached to a root node. Initially, a pattern of the engineering phrase is generated in which tokens of the phrase are represented by a single letter to denote each one, as described in section 4.5. Most commonly, apart from simple dimensions, phrases refer to *features* which are part of the whole object. All of the engineering drawings which are considered in this work, represent three dimensional objects. Therefore, a *feature* such as "HOLE" is defined as "F" type and assign it as the highest ranking type in the syntactic net. The characteristics of the *feature* are then defined by the elements constituting a particular net. Other syntactic types which occur in relevant phrases include:- Parameter "P", representing a *feature*'s parameter such as DIA; Adjective "A", which gives additional information about the *feature* such as "EQUI-SPACED"; Preposition "R", which describes the relation between two *features* such as "on". In addition, the phrase may contain numbers which could refer either to multiple *features* or to a value associated with the parameter of that *feature*. These two terms are numerical syntactic types, either a "Number", "N", where a number of *features* are described ("6" in the example), or a "Value", "V", for a value associated with a parameter ("50" in the example), depending on their positions in the phrase, and so no further processing is needed to determine their meaning. Thus, as shown in Figure 4.11, for the example phrase "6 HOLES DIA 50 EQUI-SPACED", the Syntactic Net (SN) contains Number (6), Feature (HOLES), Parameter (DIA), Value (50), and Adjective (EQUI-SPACED). The corresponding Syntactic Net (SN) therefore contains the letters A, F, N, P and V, with "A" linked to "F" and "V" linked to "P". Thus, the relationships between the tokens or sub-phrases are made explicit. This is especially important when processing more complex phrases which may involve several *features* and many parameters and their values which must all be associated correctly with the *features.*

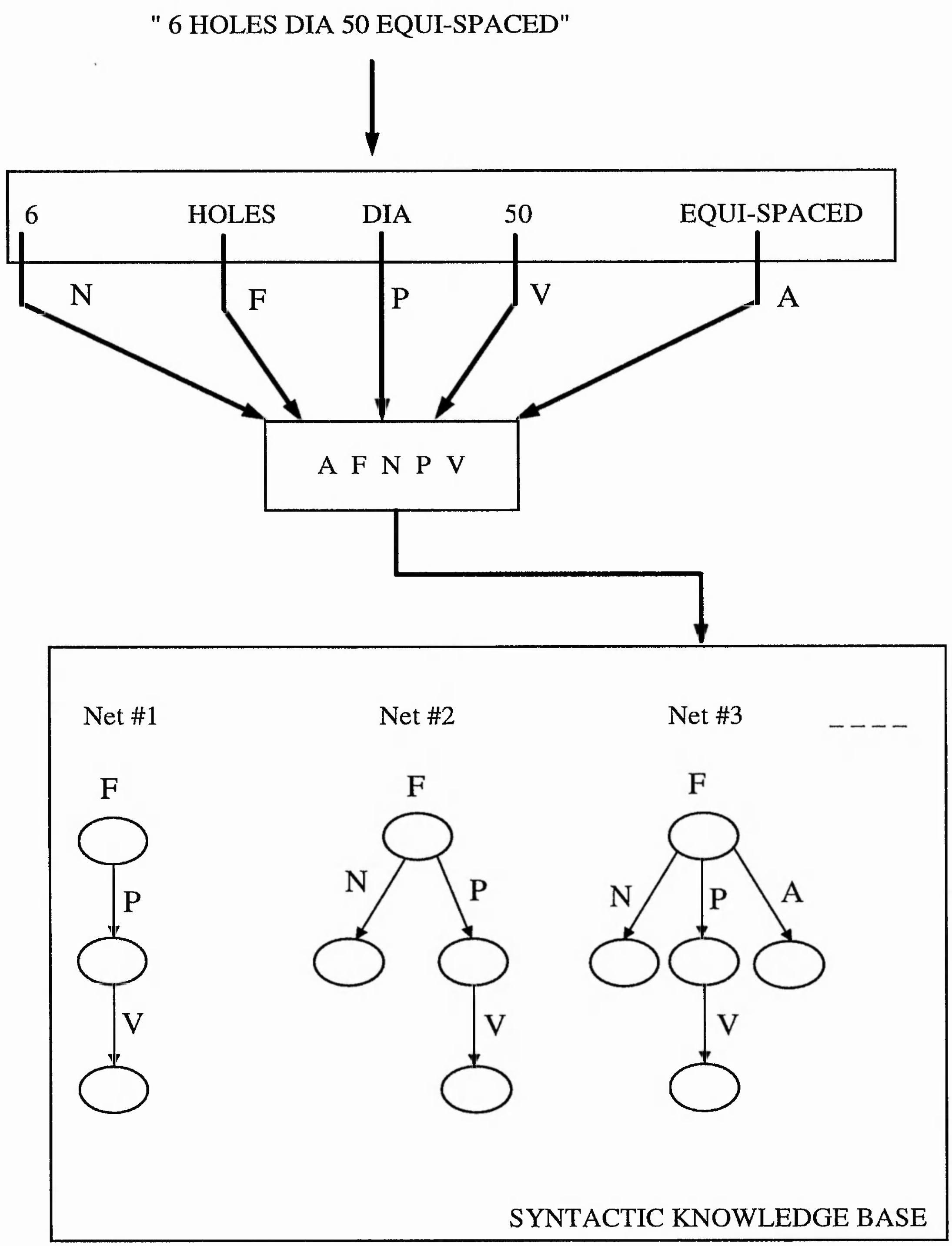


Figure 4.11: An illustration of the linking of the phrase "6 HOLES DIA 50 EQUI-SPACED" to a Syntactic Net (SN).

4.9.1 Phrases Containing name of Feature

The interpretation process of a phrase containing a feature can take a more direct approach. As described in the previous section, the phrase is associated with a Syntactic pattern, using the sequence of syntactic types of the tokens within the phrase. Then the Syntactic pattern is linked to a corresponding Syntactic Net.

Features represent top ranking elements in the net and when the phrase contains a feature, we can directly associate the feature with the top element of the net. Each token is then assigned to the appropriate relation associated with this net as shown in Figure 4.12

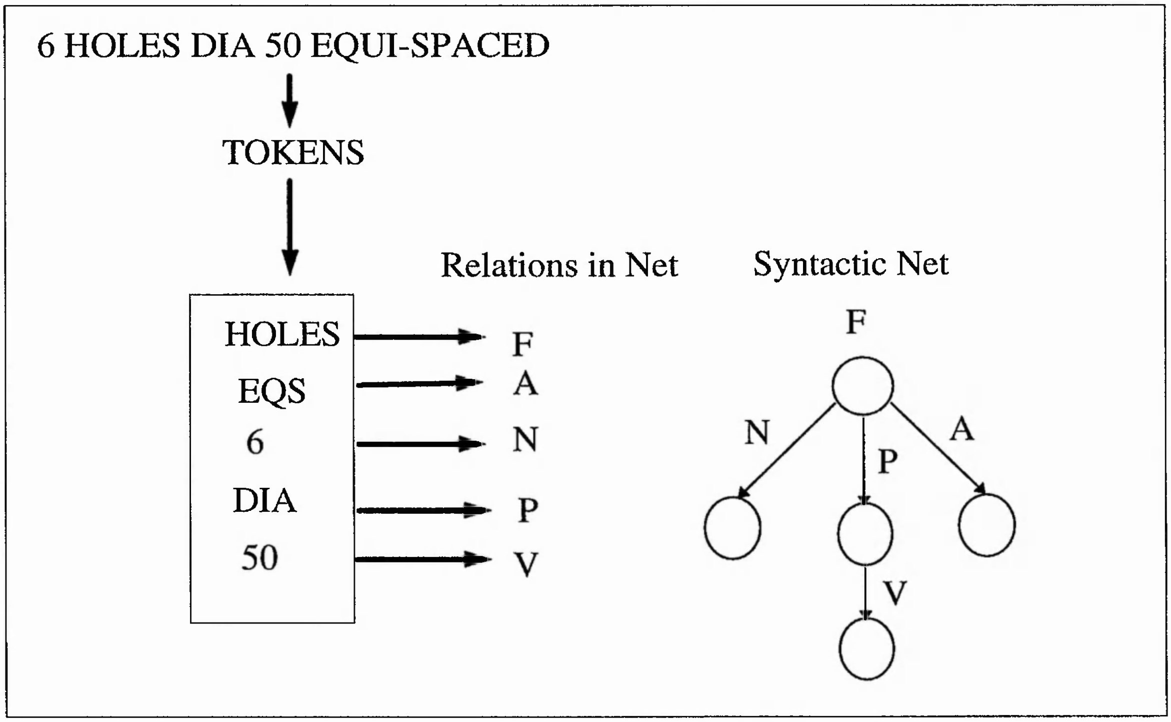


Figure 4.12: Association of tokens to the relationships of the Syntactic Net of the phrase "6 Holes Dia 50 equi-spaced".

4.9.2 Phrases without the name of a Feature

The other approach for interpreting text is used when there is no feature present in the phrase. For instance, the simple phrase " *ø76*", as in Figure 4.1, describes the size of a circular *feature*. The phrase is then associated in the knowledge base with the general circular *feature*.

The realistic approach in this case is therefore, use the relationship *parameter* to identify a general *feature* which has a *parameter* relationship that match with that indicated by the phrase. Searching of the drawing or other parts of the phrase help to recognise which *feature* has the same relationship. A hole is a possible candidate that the phrase indicated to, but there are other *features* such as a Boss which has its parameter as *Dia*. Thus, additional information from the phrase or the drawing can identify which *feature* is indicated by the phrase. Figure 4.13 shows the Syntactic Net of the phrase which leads to the identification of the *feature*.

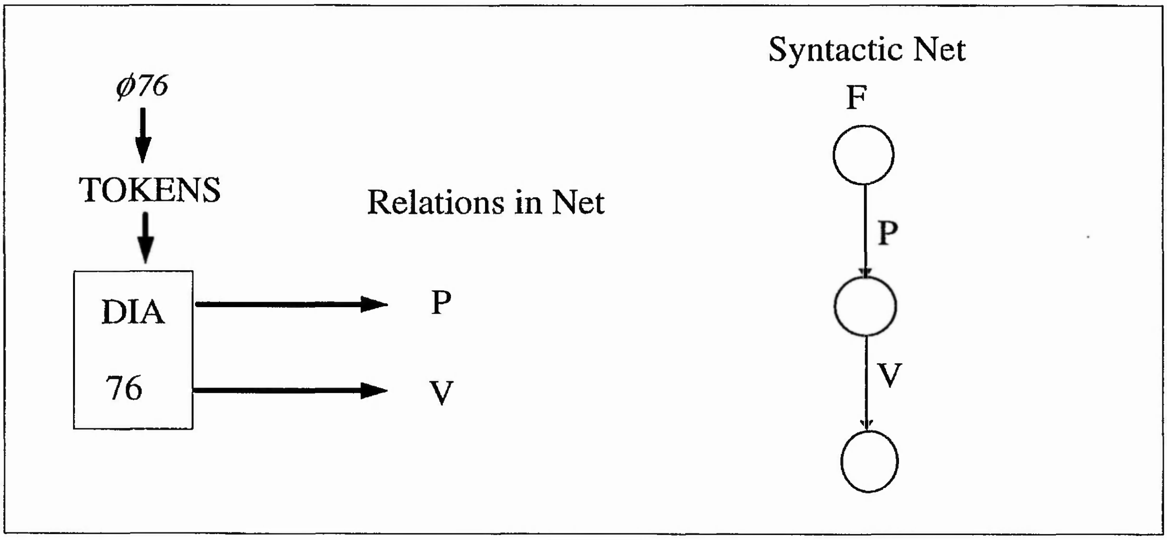


Figure 4.13: Association of tokens to the relationships of the Syntactic Net of the phrase " $\phi 76$".

4.10 Semantic Template (ST)

Once a phrase has been associated with a Syntactic Net, a Semantic Template can be generated. The Semantic Template is associated with a particular *feature* (such as a simple hole) and specifies all the information required for this type of *feature* (such as diameter, depth, position, orientation, etc.). In the example described in previous section, the Hole Template will now indicate that there is a simple *hole* with *diameter 50* and that this hole is repeated *6* times and their centres at an equal spacing. This information is not sufficient by itself to identify the six holes on the drawing, the procedure needs to know the position of at least one of these holes. The information obtained from the phrase is then passed to a searching function associated to holes, to enter the corresponding projection. This searching is simplified by the fact that, in our system, phrases are stored together with reference to a particular projection where the phrase was originally displayed, so allowing the search to be restricted initially to the projection where the phrase has appeared. The search is further simplified by exploiting information from the *leader* lines on the drawing.

The Hole searching function is specifically developed to identify holes on drawings. On the projection where text has appeared, the shape of the hole could be represented either by a rectangle with some hidden lines or a circle. The circle may also be represented by either hidden or solid line depending on whether the hole was obstructed by other parts of the object or not.

In Figure 4.14 the phrase "4 holes Ø 26" in the top right view (on the side projection) describes *4 holes* and two of the four holes are obstructed and therefore, represented by dashed lines. Also the side projection of the example of Figure 4.1, has the phrase "2 holes 10 DIA", indicating the presence of holes. These different representations can be expected by the searching function in order to be able to deal with them. By processing this information together with the information obtained from the phrase, the holes can be located on the projections. The centres of holes are usually indicated by the intersection points of two centre lines or intersection points of the pitch circle with a centre line. These intersection points can be identified and tested to ascertain whether they represent the centre points of the holes. All information obtained from the phrase "6 Holes Dia 50 equi-spaced" and by the search which help describe the *feature* are added to the Semantic Template, as shown in Figure 4.15.

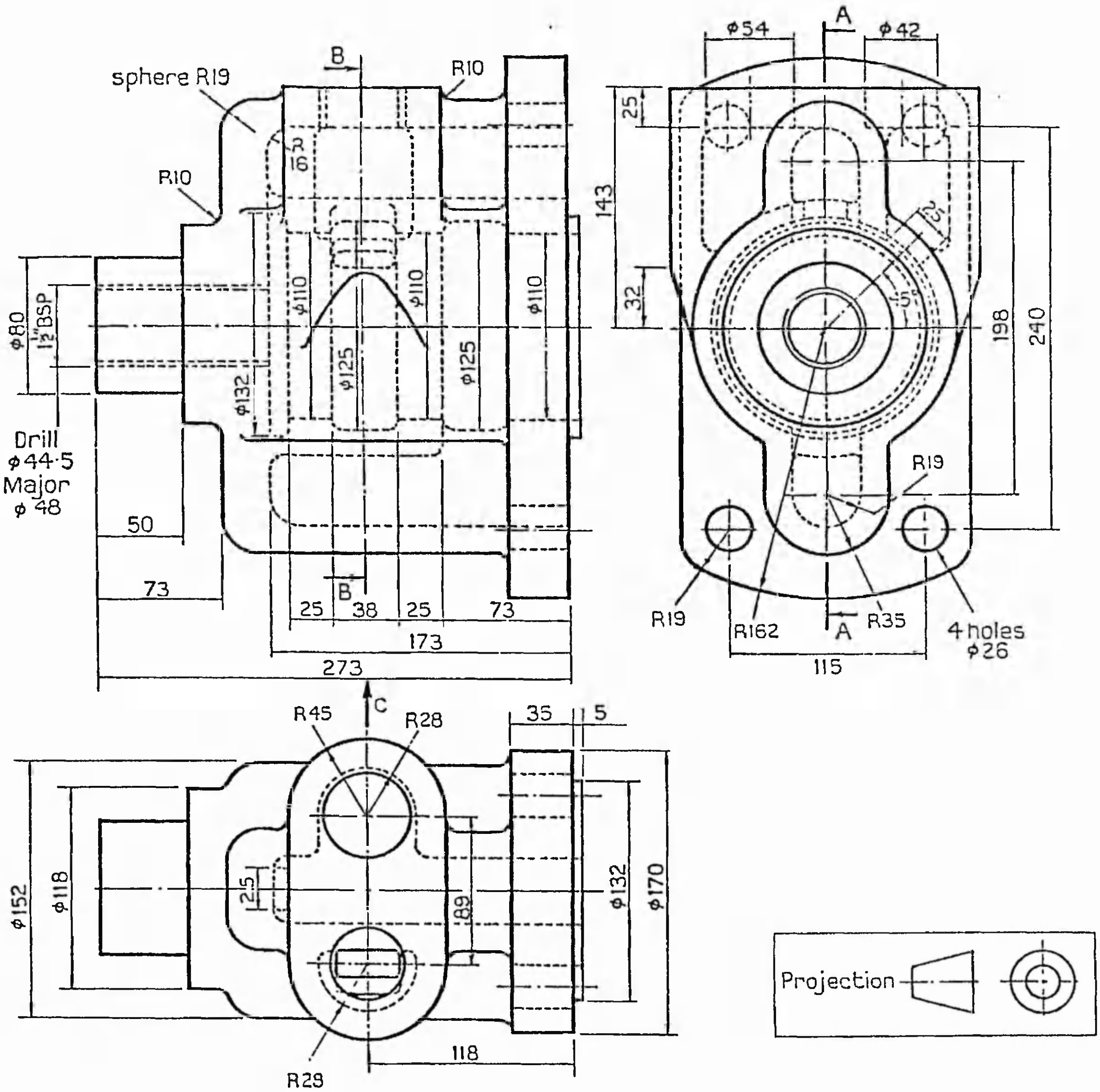


Figure 4.14: Example of Engineering Drawing, (taken from [Hart75]).

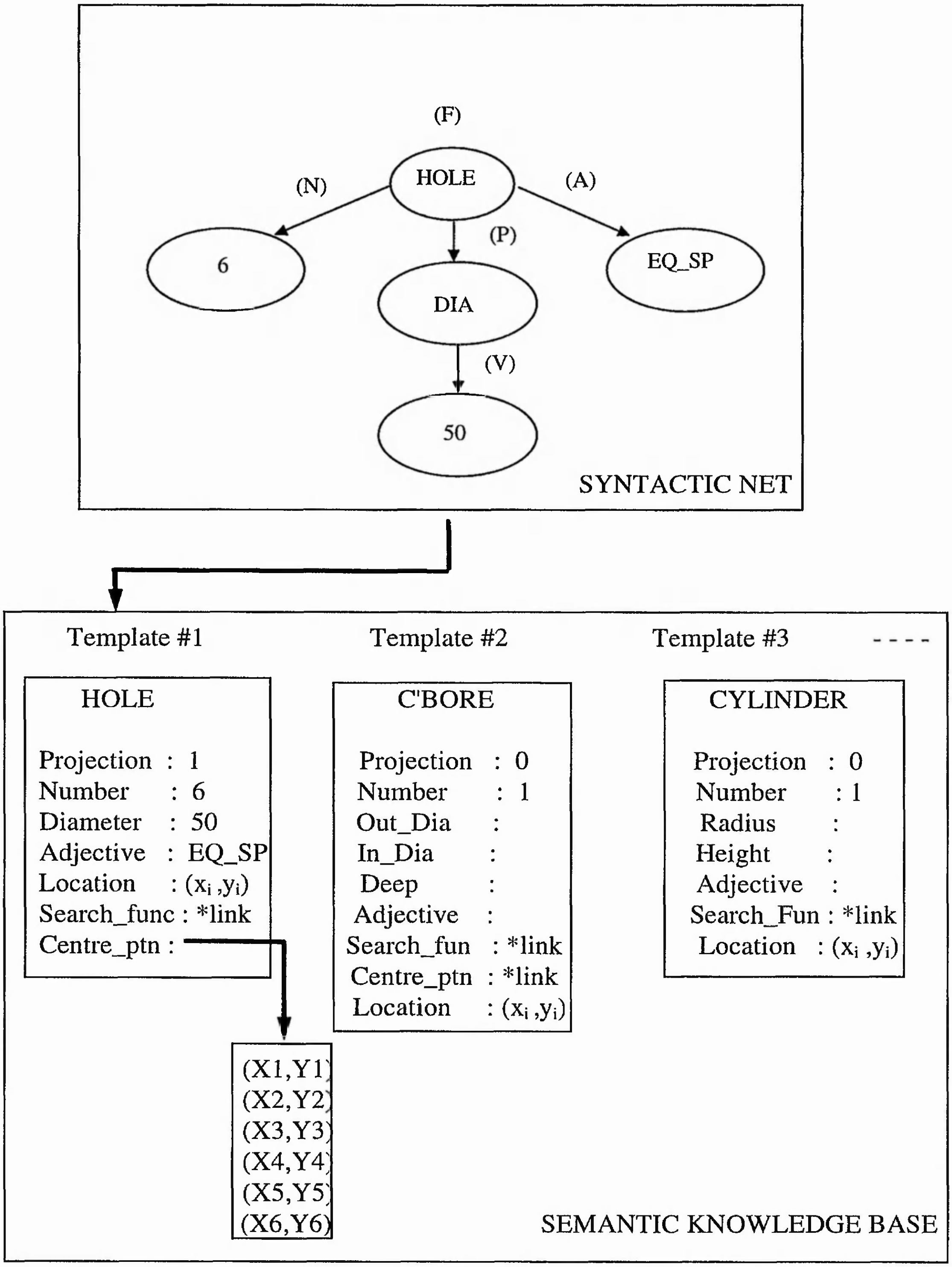


Figure 4.15: An illustration of the association of the Syntactic Net of the phrase "6 Holes Dia 50 equi-spaced" to its Semantic Template.

4.11 Template Representation

Templates are semantic networks in which *features* and their relationships are described. A record of the *feature* is created once it has been identified. Elements of the record can be associated with values identified from the phrase, others maybe determined from the drawing.

The HOLE template can accommodate all the characteristics which fully describe the hole on the drawing. Normal hole is the parent *feature* of the holes family, other types of holes inherit the characteristics of the parent in addition to their own. This grouping of *features* minimises the number of templates used to describe *features*.

Knowing the shapes of *feature* in all projections helps to identify and recover the information about the *feature*. With this knowledge, the identification of one shape of a *feature* leads to the identification of the other shapes of the *feature* in other projection Figure 4.16.

The shape of holes in two of the three projections are the same, and it looks as if one shape has been copied from the other and translated a distance. The analogy of the shapes in the two projections will not exactly be the same if the features exchanged positions on the projections. The shape of the feature on the two projections obviously will not change but one has rotated at right angle in relation to the other. This analogy between projections can also happen for the whole object represented by the projections. The restriction in this case is that the drawing should be composed of one simple object not an assembly of objects.

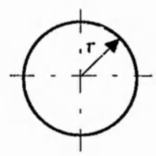
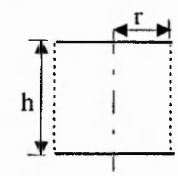
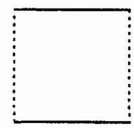
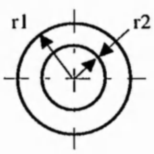
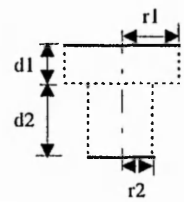
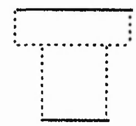
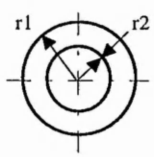
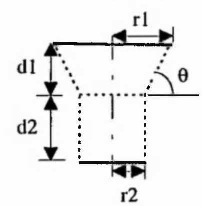
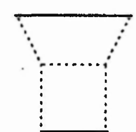
PROJECTIONS	Projection1	Projection2	Projection3
NORMAL_HOLE			
COUNTERBORE_HOLE			
COUNTERSUNK_HOLE			

Figure 4.16: Illustrates the shapes of three types of holes in the projections of a drawing (Projections 1,2,3 can be any one of Front, Top and Side views).

4.12 Searching Other Projections

Once the *feature* has been located in one projection, the possible representations of the *feature* which may appear in the other projections are derived from the stored knowledge base. Thus, having located the projection of the *feature*, such as a circle representing the hole labeled “DIA 50”, the algorithm “expects” to find the remaining two representations of the hole depicted in the other two projections (which in this case are rectangles made up of hidden lines. The two projections are then searched for the appropriate representation of the *feature*. This search yields additional information about the *feature*, in this case, the depth of the hole. Thus, additional information about the *feature* can be derived from the three projections to complete the Semantic Template. When no representation is found in other projections, sufficient information can still be derived, for example the depth of the holes, since holes are assumed to be penetrating unless specified otherwise in the text.

To illustrate this procedure with the example of Figure 4.17, the front projection has text indicating the presence of 4 holes. Processing this text will generate a hole template that have 4 identical holes of size *50*. Following leader line and using the fact that centre of holes coincide with the intersection points of centre line and a pitch circle, it is possible then to identify the positions of these *holes*. The size of the holes is known from the phrase, therefore, for each possible location, a test of recognising whether the hole is complete, distorted or omitted. The latter possibility requires further tests to confirm the existence of one of the holes that the algorithm is trying to locate and therefore stores it away to be tested last.

Distorted holes (incomplete circles due to scanning errors) are rectified, all segments within the small circular search area around the presumed circumference of the circle are removed and replaced by a complete circle. For each complete circle found, a search for the silhouette lines is directed vertically and horizontally by extending imaginary lines from the sides of the circle. These imaginary lines are formed by a small strip of width *d* (small tolerance), rectangles in other projections are searched for. Because holes are penetrating, the expectation is to find hidden lines within the extended strip in other projections, otherwise presumed missing and therefore have them created. Finding the first hole in all three projections (circle in one projection and hidden lines in the other two) will obviously make the task a little easier for locating the remaining three, because the circles will be all in the same view and are equally spaced. Since all of these holes are identical, the height of the first hole can be added to the hole structure and therefore can be use to assert the height of the remaining holes.

Holes are common *features* that can appear in almost all engineering drawings, they may also appear in any one or more of the three projections. Generally, holes are found by first searching for the circles in the projection where the text appears, then searching other projections to find the rectangles. But this is not always the case, because text indicating the presence of holes may also be associated with hidden lines (side view of the hole) on the projection. The consistency of other holes found by the searching algorithm is always checked with the information stored in the hole structure (Semantic Template).

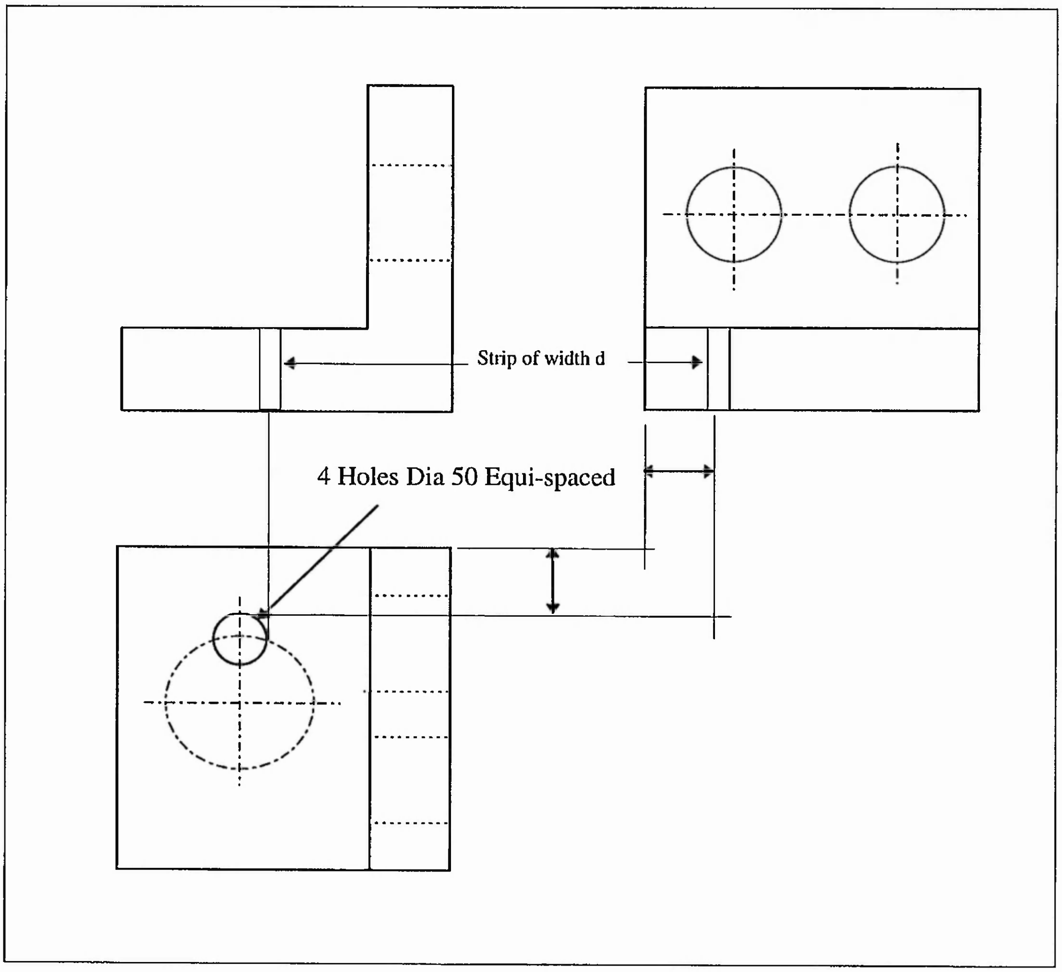


Figure 4.17: Illustrates searching of other projections for the complete description of the hole feature. Using a strip of small width hidden lines of the hole in other projections can be found.

4.13 Feature Description

Simple *features* are described in terms of lines and arcs, as shown in Figure 4.18. The description consists of the three orthographic projections representing the Front, Top and End projections of the *feature*. *Features* are only described by their general shape, they have no specific information about their size or position on the projections.

Features such as simple holes and counterbore holes are not difficult to determine on drawings, a hole is simply represented by a circle in one projection and rectangles in other projections. Also by engineering convention, the centre of this circle should coincide with the intersection points of two centre lines or the intersection points of a centre line with a pitch circle. This convention also applies for the counterbore hole, but its characteristic described in one projection by two concentric circles, and two rectangles in the other two views. Using the combination of facts and engineering conventions, these *features* have been easily derived and described in a knowledge base. However, holes can be described in the knowledge base as having one closed arc in one view, and hidden lines for rectangles in other views.

A phrase referring to the presence of a hole on a view can be associated with any view of this *feature*. This therefore, requires the searching procedure to look for either a circle, or a rectangle. The three descriptions of the *feature* in the knowledge base can then be associated with a projection if part of the feature matched with any one description in the base. The other two descriptions are used to match the *feature* in other views.

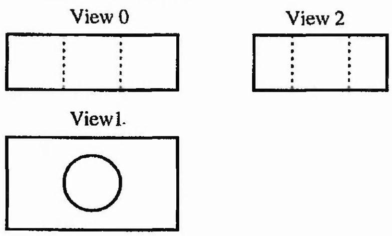
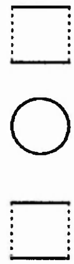
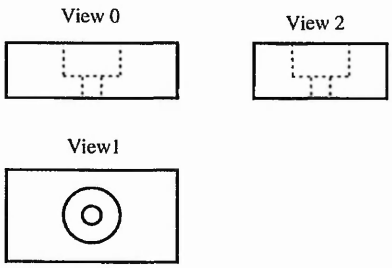
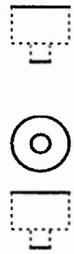
Feature Name	Features on Engineering Drawings	Feature Descriptions
SIMPLE HOLE		
COUNTERBORE HOLE		

Figure 4.18: Stored descriptions of features.

The description of counterbore hole would have two closed arcs of different sizes having the same centre node, in one projection, and two rectangles comprising of hidden lines for each of the other two projections.

Features other than holes may not be so easy to recognise. It is therefore, necessary to implement an algorithm to recognise the *feature* indicated by the phrase and match it with descriptions stored in the knowledge base. Chapter 6 describes a method of *feature* recognition which can be incorporated with text interpretation.

All described features are associated to functions which can be used to update the Semantic Template with all the information found on the drawing. These functions are activated to search the drawing for their positions. The next section will explain the action of these functions with an example.

4.14 Geometric Representation of Textual Information

In general, the work described in this thesis, is devoted to the design of a compiler to translate textual information on drawings into geometric representation. These translated geometric *features* are added to the data file to give a complete geometric description of objects on engineering drawings.

Assertive textual information refers to *features* or their relationships to other *features* on the drawings have been interpreted. Therefore, this work is mainly concentrated on the identification of *features* from the phrases, and the retrieval of the stored descriptions of these *features* from the database. Templates associated with these *features* can be updated with the relevant information extracted from the phrase and the drawing.

The information required to satisfy this objective, is to associate values to the characteristics of a particular *feature* which will be sufficient to describe and display it on the drawing. Each *feature* has a different number of characteristics which best describe them. Thus, *features* are described as structures with their characteristics as elements of that structure.

Features in the database are described in terms of lines and curves which define their shapes. Their sizes and positions on the drawing can be obtained from the phrase and the drawing.

The most important property of text interpretation is that *features* and their characteristics are linked by pointers to further processing. This strategy generates a chain of actions on the recognition of a particular *feature*. The procedure for *feature* recognition is to interface with the knowledge base. *Features* and their descriptions are stored in this knowledge base. Thus, *features* indicated by the phrase can directly be matched with this base.

The indication of a hole in a phrase, means one and only one representation of physical shape. Stored description of the matched *feature* can then be loaded to the *feature* structure. This can then be used to build up all the necessary details about the *feature*.

The identification of the *feature* also activates other processes, such as to retrieve the structure of a “HOLE”, functions to search the drawing and the phrase to associate values to its characteristics. Some characteristics of the *feature* may be identified straight away from the phrase and inserted into their slots. Other characteristics may require searching the drawing to find their values. Projections can be searched by activating specific *searching functions*.

These *searching functions* are associated to some hypotheses to which human would visually identify a specific part on the drawing. Other criteria such as matching the database description of *features* with that on the drawing which have been used to some extent but described in more detail in chapter 6.

The information extracted from the phrase " 6 HOLES DIA 50 EQUI-SPACED ", in the previous section will activate a *searching function* that searches for holes in the specified projection. Simple holes are represented by circles in one projection, and by rectangles in the other two projections. At least two sides of the rectangles will be represented by dashed lines denoting the profile or the hidden part of the hole. According to British Standard for Engineering Drawing [BSI], holes should be drawn on centre lines. The intersection of these lines should coincide with the centre of the hole. This procedure would easily determine the position of any hole on the drawing, but:

(a) - Is it the one which is referred to by the phrase? and,

(b) - Does it have a complete representation on the drawing?

These valid cases need to be checked and an appropriate action will then be taken. In case (a), the sizes of all holes on the drawing are compared with the one stored in the Semantic Template (recognised size of *feature*). Holes that do not match are discarded from the test, and those that do match are processed and searched for in the other views.

Information in Semantic Template will be updated to have the position of the one located on the drawing. Holes that are distorted due to the scanning process or abbreviated by simplification, are then checked for case (b). Again the focus point is the intersections of centre lines, which represent the centre of the hole. Checking for vectors around each intersection point within an annulus to determine whether this point represent the centre point of any of the holes that we are looking for. The area inside the annulus should cover the presumed outlines of the identified *feature*. Frequent line segments or arc segments found in the specified annulus indicates the presence of a hole. Therefore, these line and arc segments are discarded from the list and a circle is created at the intersection point of these centre lines.

Other indication which associates text to a particular part on drawing are leader lines. A leader line is so called because it leads the reader to the part of the drawing which is referred to by the text. Leader lines have been described in Chapter 3, section 3.8, in relation to dimensioning. A piece of text referring to holes on the drawing normally has a leader line touching at least one of the holes indicated by the phrase. Therefore, this can lead the interpreter to the location of the first hole in one projection, while the others can be determined by further understanding of the phrase. Generally, identifying them will lead the process to the part of drawing associated to the text. Leader lines associated with holes may either point at the centre of one side of the rectangle (projection of hole in one view) or touching the circle (projection of hole in other view). Therefore, identifying the leader line can also determine the position of the hole in the referred view.

All geometric information recovered by interpreting the text is then added to the geometric data file. The data file then passed to the bottom-up procedure [Shaw95] to reconstruct the 3-D wireframe model.

4.15 Summary

The algorithm described in this chapter is capable of interpreting a number of simple phrases on engineering drawings. This algorithm uses syntactic nets to relate phrases to their specific domain. Tokens (numbers and words) in a phrase are assigned a syntactic type and the collection of these syntactic types is linked to a Syntactic Net.

The Syntactic Net is then matched to a Semantic Template which in turn identifies the *feature* in the phrase. Also this Semantic Template activates searching functions to search the projections for matching *features*, which determine the actual location and size of the *feature*.

The advantage of using this method is that it is flexible, other concepts can be added to the knowledge base and so enable it to interpret further phrases. Text is read from a data file and passed to the interpretation process without the user's help. Thus, this interpretation is generated automatically on the detection of text in the data file.

Text on Engineering drawings are simple phrases, therefore, it was decided to develop this algorithm which identifies tokens as the main components of the interpretation. This approach, therefore, was not concerned with interpreting grammatically correct phrases but interpreting meaningful representations within the specific domain. With this approach, even partial understanding of the phrase may still generate a meaningful interpretation.

The method described in this work is an improvement on the method used by [Yoshiura84]. The interpretation process treats text as phrases which are read from the drawing data file. The approach implemented in this work is most appropriate in order to have an automatic interpretation. The principle has been established for a small range of phrases. It can be extended to directly interpret many notes on many engineering drawings.

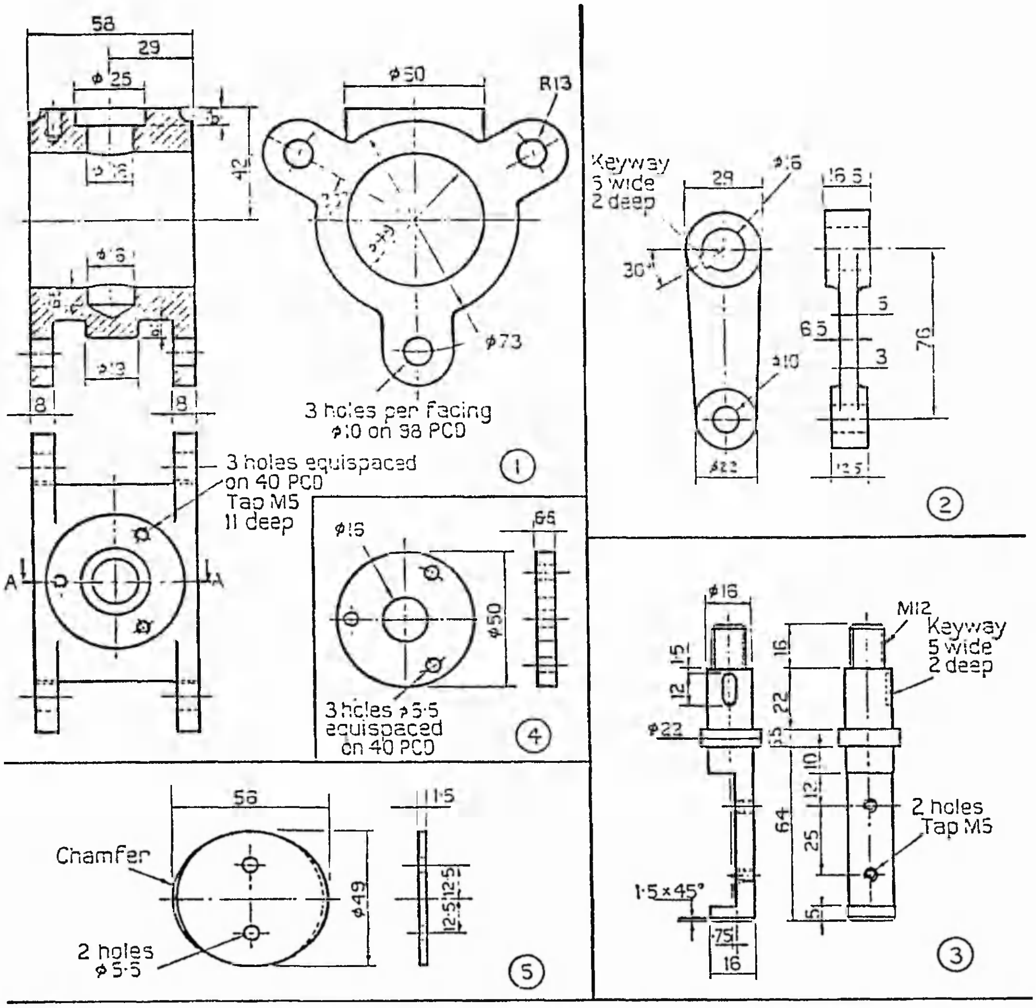
Chapter 5

Interpretation of Prepositional Phrases

5.1 Introduction

The interpretation process for text on drawings described in chapter 4 has been extended to include some prepositional phrases. Many prepositional phrases in the engineering domain consist of two or more simple phrases related by a prepositional term, Figure 5.1. A critical point is how to break down large concepts into their component parts. Usually, there are so many branchings that it is very difficult or impossible to do an exhaustive search of all the appropriate breakdown paths without any supplementary information.

This problem is manageable with the algorithm described in this thesis, because the number of primitive *features* to be considered, in both two and three dimensions, is usually quite small. Also applying common-sense knowledge concerning mechanical parts and their relationships as well as knowledge of conventions in engineering drawings can simplify the task of searching projections for these parts.



ITEM	TITLE	MATERIAL	Nº OFF
1	Valve housing	Cast iron	1
2	Arm	Cast iron	1
3	Shaft	Steel	1
4	Cover	Steel	1
5	Valve	Steel	1
6	M5×16 long CH HD screw for ④	Steel	3
7	M5×6·5 long CH HD screw for ⑤	Steel	2
8	M12 hex nut for ③	Steel	1
9	M12 plain washer 3 thick	Steel	1
10	5×4×12 long key	Key steel	1

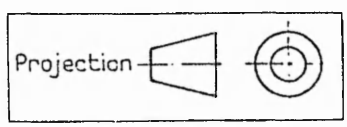


Figure 5.1: An assembly drawing showing a number of prepositional phrases (taken from [Hewitt75]).

The method of extracting tokens used in Chapter 4 is used to identify the prepositional terms. These can then be used to help recognise parts of the phrase. To simplify the interpretation of prepositional phrases, the author has used an approach to categorise the phrases into sub-phrases and assign a phrasal type to each sub-phrase. This kind of categorisation can be used to define facts about a prepositional phrase. Before describing the algorithm for interpreting prepositional phrases, a little should be said about the knowledge needed to run the interpretation process, since knowledge is the fundamental part of the algorithm.

5.2 Knowledge Engineering

The use of the word 'Engineering' clearly implies the existence of some raw material which can indeed be engineered. In this sense of the word, what do we mean by 'knowledge'? Clearly, we mean the representation of human knowledge, gathered from an acknowledged expert in the field under study (the 'domain expert') by the 'knowledge engineer' - the developer of the expert system.

There seem to be little agreement on how knowledge is represented in the human mind. Without an agreed objective model, it is inevitable that many competing and complementary formalisms will be proposed and used with varying degree of success in the endeavor to create adequate computerised replication. One of the more frequently employed formalism is that of the 'production rule'.

There are two important questions about the representation of knowledge. First there was the question of how knowledge is represented in the human or animal brain, and now what structures may be used for computer representation. The first question is the concern of cognitive psychology and psychoanalysis. However, the theories of the cognitive psychologists have much to offer in the way of ideas and cognitive simulation has been an important issue in research on expert systems.

Artificial intelligence is regarded as research into human psychology via the analogy with computers, while knowledge engineering is the practical application of the techniques from all parts of computer science to solving a range of problems in the construction of better decision support system.

The inter-disciplinary subject of artificial intelligence has indeed been defined as "the study of mental faculties through the use of computational models" [Charniak85]; exactly the reverse of what the designers of knowledge-based system trying to achieve. Perhaps this is why there is such a considerable overlap and confusion between the fields of artificial intelligence and knowledge engineering today. One important point to make categorically is that no one knows how the human brain works and no one can give a prescription for the best computer knowledge representation formalism. Until some pretty fundamental advances are made, the best approach as an algorithm designer is to use pragmatically whatever formalism best suits the task at hand.

Knowledge is usually seen as a concept at a higher level of abstraction, and there is a sense in which this is true. For instance, *'4 Holes on 100 pcd'*, is knowledge about the hole and the pitch circle. Information, on the other hand, always has a context, for instance, *'Dia 30'* indicates information about the size of a circular feature.

The realisation that much knowledge is expressed in the form of heuristic descriptions or rules is what gives rise to the conception of knowledge as more abstract than information. In our application (interpretation of text on engineering drawings), much of the information is obtained from the text and the geometry on the drawings and knowledge is obtained from humans which is incorporated in a knowledge-base. Quite apart from its ability to be abstract at various levels, *knowledge is concerned with action*. Knowing about conventions in engineering domain, helps us to determine the location of a hole on the drawing. Knowledge is a guide to informed practice and relates to information as processor; that is, we understand knowledge but we process information. It is no use knowing that holes are present on drawing but do not know the shape or where to search for them. Effective use of knowledge leads to the formation of plans of action and ultimately to deeper understanding.

There are several evidently different types of knowledge at hand: knowledge about *features*, events, task performance, and even about knowledge itself. If we know something about holes we will probably know that holes on engineering drawing are represented by a circle in one projection and rectangles of hidden lines in other projections. The methods used for encoding the knowledge are closely bound up with methods used to manipulate knowledge in the presence of facts.

There are two expert systems that can be considered for an intelligent application; one of which relies on data structures and the other relies on knowledge. Decision support systems (DSS) are richer in data structures which permit higher level of user interface. In contrast, knowledge engineering systems are richer in knowledge structures which permit more intelligence, Figure 5.2, shows the kinds of knowledge and techniques used for this project.

Expertise = inference + knowledge

*** Richer knowledge structures permit more intelligence systems**

Features knowledge	**- Structures**
Causal knowledge	**- Rules**
Process knowledge	**- Functions**
Knowledge of relationships	**- Logic**

Figure 5.2: Knowledge Based Programming.

The expertise of the algorithm depends on how much knowledge it has about the domain and the judgment at which an outcome is selected, Figure 5.2. Knowledge about the geometric shapes is described by structures. Facts or events of geometric relationships are described by rules. Logical operators are used to combine a number of rules in a relation. Finally, process knowledge is defined by functions to enter the drawing to process the geometry.

For many years, and even to the present day researchers are still working to produce a fully automatic interpretation method for engineering drawings. The ultimate goal is within sight but not sufficient for the method to be fully dependent upon itself. The work in this project therefore, has been implemented on the same foundation, by applying a method which uses knowledge engineering embedded within the algorithm. The information can be extracted from the text which can then be used by the rules and the knowledge to generate an appropriate action.

The essential part of this approach is to present the knowledge in the whole complex of conventional algorithms which were being engineered. It is not difficult to see that human problem solvers rely on a vast range of representations and are likely to switch between them during problem solving. A knowledge engineering environment has to be equipped with all possible situations on engineering drawings which will make it capable to deal with.

5.3 Relational Template

The interpretation method uses parsing rules which are defined in terms of restricted semantic categories rather than general grammatical categories. This therefore, can only produce interpretations meaningful in the domain. Many phrases may be ambiguous in a general context, but using the algorithms developed in this project, these phrases are correctly analysed to have a unique meaning in the technical domain.

The method of syntactic types of a phrase described in chapter 4 uses a pattern of the tokens extracted from the phrase. The collection of these tokens can then be related using the knowledge base and pointers which associate them with the appropriate Syntactic Net and Semantic Template.

There are a number of prepositional phrases in the engineering domain and some examples are shown in Table 5.1. These phrases have a relation format which relate different engineering *features* or situations. Prepositional phrases are identified by recognising the prepositional term included in the phrase such as 'on', 'in', ‘inside’, ‘outside’ and many others. The rules then generate a relational structure representing a geometric relationship between parts of the phrase. For example the prepositional phrase *"2 holes 14 dia on 95 pcd*", will be segmented into two sub-phrases. Upon the recognition of the prepositional term "*on*", the phrase is partitioned into "2 *holes 14 dia*", as the first sub-phrase and the remaining of the phrase "*95 pcd*", as the second sub-phrase. The prepositional term "on", links both sub-phrases by a process. In addition, each sub-phrase can be categorise to have a sub-phrase type, "2 *holes 14* *dia*", has *Feature* type, and "*95 pcd*", has also a *Feature* type. These sub-phrases are interpreted as simple individual phrases. But in this category, they are interpreted within a relational structure which links the sub-phrases by a relation. This kind of strategy therefore, identifies the sub-phrases of the original phrase and associate it with a particular process (pre-defined relational functions). These functions can be activated on the satisfaction of a number of conditions. The conditions are for example, the situation of holes placed on a pitch circle, which can then be linked to the specific function that describes this particular situation. All relational situations between *features* must be included in the algorithm in order to make it capable of interpreting a large number of prepositional phrases.

Prepositional Phrase	*Related Features*
6 HOLES EQUI-SPACED ON 200 PCD DIA 30 C'BORE DIA 50X20 DEEP.	Counterbore Hole and Pitch Circle
2 HOLES 14 DIA ON 95 PCD	Normal Hole and Pitch Circle
3 HOLES ϕ 13 REAMED ON 140 PCD	Normal Hole and Pitch Circle
3 HOLES 5 DIA SPOTFACE DIA 10x2 DEEP EQUALLY SPACED ON 82 PCD.	Spotface Hole and Pitch Circle

Table 5.1: Shows some common prepositional phrases and the related features they referring to.

Using the fact that in the engineering domain holes (c'bore, c'sk and others) are some of the *features* that can be placed on a pitch circle, the rules for identifying this type of phrase is shown in Figure 5.3. The relational structure will points to the two *feature* templates.

BEGIN

IF both sub-phrases type = "*feature*"

Identify *feature* in sub-phrase1 and create its Template.

Identify *feature* in sub-phrase2 and create its Template.

IF relationship = **[preposition]**

THEN correlate the two Templates by the relationship term

PROCESS Procedure (***feature1 preposition feature2***);

ENDIF

END

Figure 5.3: Shows the set of rules for identifying the prepositional relation between two features.

When the above rules are satisfied, a function *holes on pitch circle* is activated, it first searches the drawing for the pitch circle as described in Chapter 4. The algorithm has been made more efficient, it deals first with situations which are most likely to occur on engineering drawings.

The Syntactic Net of the *feature* "hole" in the above example, has two elements attached to it as it is clear from the initial phrase. One link points to the number of repeated *features* and the other points to the parameter associated with the *feature* which in itself has one link pointing to the size of the *feature*. On the other hand the pitch circle has only one link which points to the size of the circle, see Figures 5.4 and 5.5.

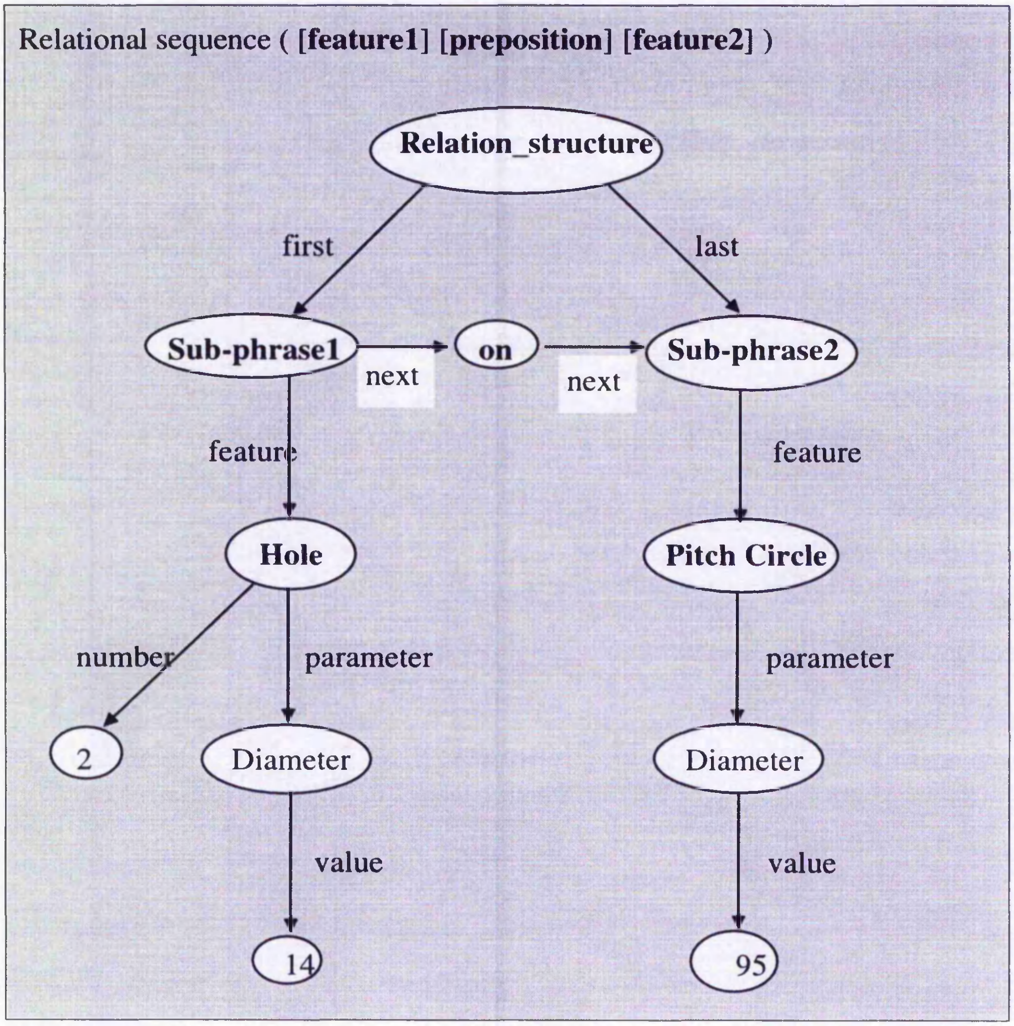


Figure 5.4: A relational structure of the prepositional phrase "2 Holes 14 Dia on 95 pcd".

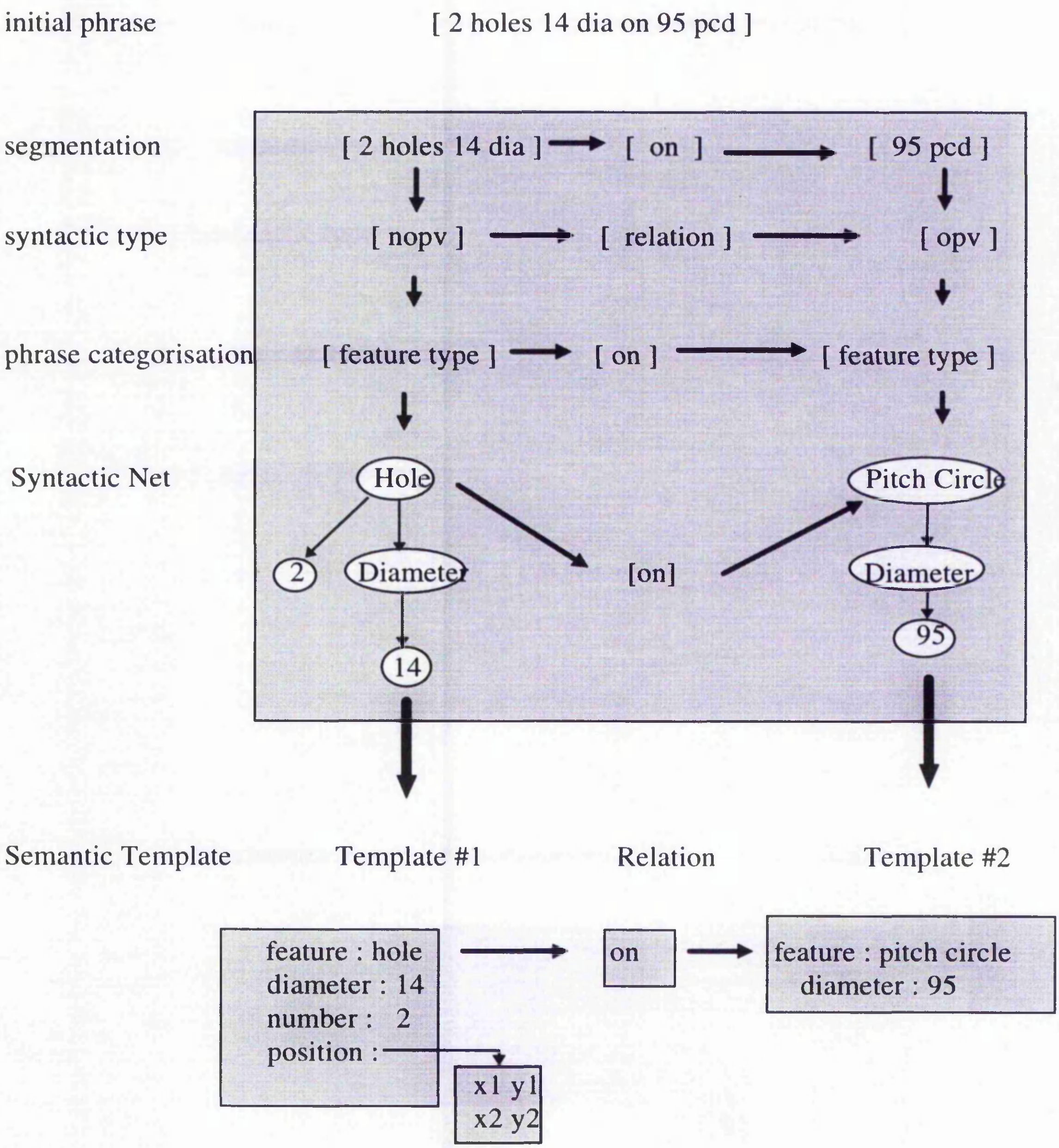


Figure 5.5. Syntactic and Semantic representations of the prepositional phrase " 2 Holes 14 Dia on 95 pcd ".

5.4 General Interpretation of Prepositional Phrases

The engineering domain is not a very large domain and text on engineering drawings normally are simple phrases. This text may not be grammatically correct but it serve the purpose of describing different components on the drawing. As described in chapter 4, the interpretation method identifies tokens of the phrase and by syntactic and semantics links a meaning is generated for the phrase. In general prepositional phrase consist of a preposition plus a nominal group, for example the phrase "*hole in the centre*". The algorithm developed here will interpret the above phrase as prepositional of two features. Sub-phrase1 will be consist of the token "Hole", and sub-phrase2 will consist of the token "centre", which is the nominal group indicating to the centre of a *feature*. The token "in" is the preposition.

The advantage of the method of text interpretation developed in this work is that it uses only the tokens which have meaning to the interpretation. Token "the", from the above example will be ignored. The example of Figure 5.6, taken from Chou and Wozny [Chou88] have the following text about Hole; "*6 HOLES DIA 10.5 EQUI-SPACED ON 120 PCD. BOLTS OR STUDS AND NUTS ARE NEEDED TO SECURE THIS COMPONENT TO THE BODY*". This text comprises of two separate phrases, the first describes a relationship between two geometric *features*. While the second describes additional information related to the assembly of *features* but has no relation to the geometry. The algorithm has no Syntactic Net or Semantic Template to accommodate for the second part of this text. Thus, the second phrase will be eliminated from the interpretation.

The algorithm as described in the previous section, will recognise the first phrase which comprises of two sub-phrases with sub-phrase one "6 HOLES DIA 10.5 EQUI-SPACED" and subphrase two "120 PCD". Sub-phrase one has a *feature* type and sub-phrase two has also *feature* type.

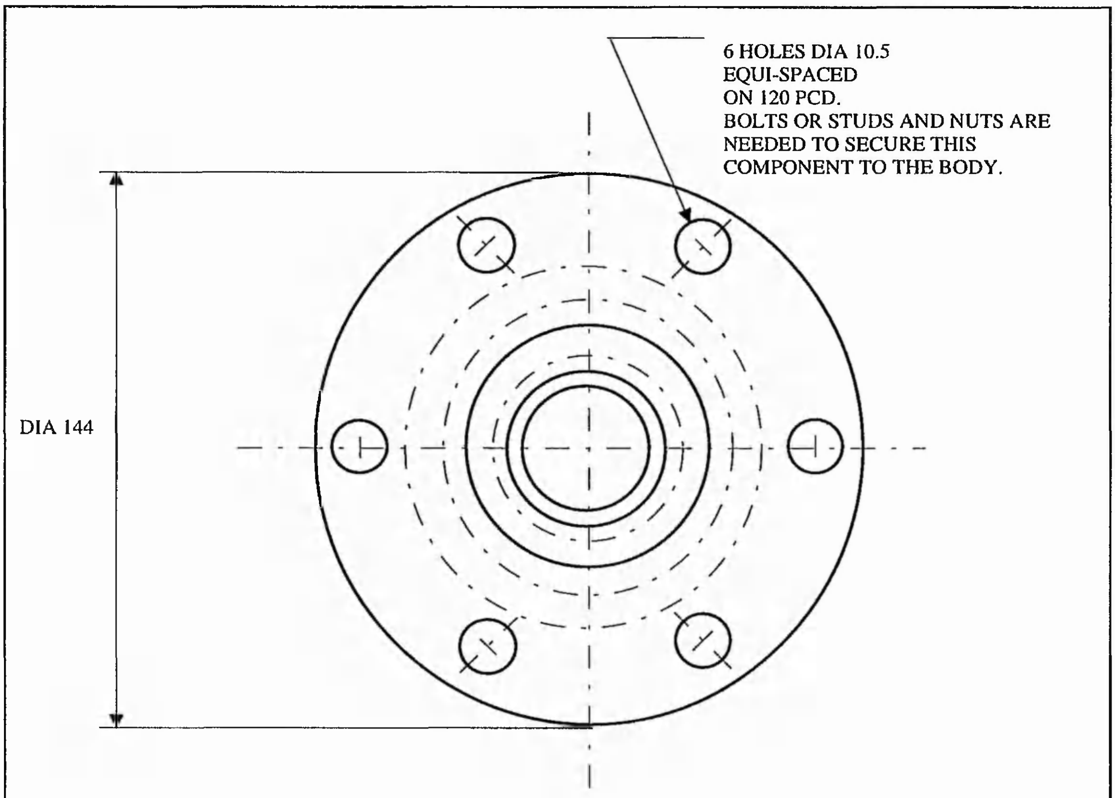


Figure 5.6: An example for a Hole note (taken from [Chou88]).

Some prepositional phrases may require further processing for example the phrase; " *4 holes equi-spaced on 100 pcd dia 30 c'bore dia 50x20 deep* ", Figure 5.7. However, the algorithm will recognise this phrase as a prepositional and therefore, segment it into two sub-phrases. The segmentation take place at the preposition term, where text prior to this term can be allocated to sub-phrase1 and text after the preposition term will be allocated to sub-phrase2. Also the algorithm will categorise sub-phrase1 as having *feature* type and therefore will be pointing at its template with the information "*4 holes equi-spaced*". Sub-phrase2 also will be pointing at the its template with the *information "100 pcd dia 30 c'bore dia 50x20 deep*". The algorithm will also recognise that there are two *features* in sub-phrase2. This finding will trigger further processing to investigate more information about the two *features* in sub-phrase2.

The initial assumption of this situation is that the two *features* could be pitch circle and counterbore hole. The common-sense applied to identify these *features* is as follows; pitch circle will always be accompanied by a parameter value which must always be positioned before the *feature* in the phrase. While counterbore hole comes with three parameter values. The algorithm, at this stage will segment this part of the phrase which contains the two *features*, into sub-phrase2 and sub-phrase3. Sub-phrase2, which comes after the preposition term will be pointing at by pitch circle template. Sub-phrase3 will be associated with the same template of sub-phrase1 as it has been shown in Figure 5.8.

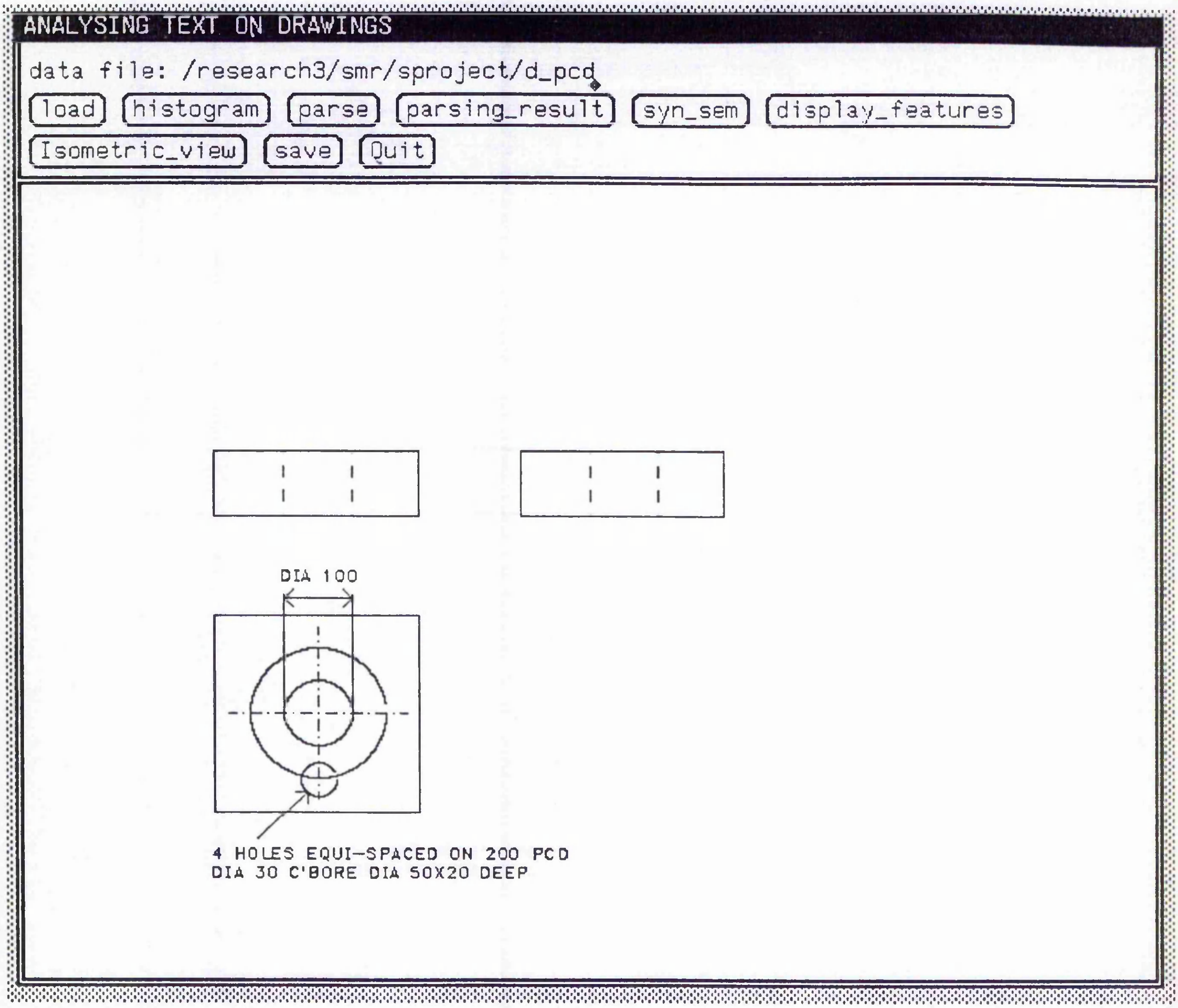
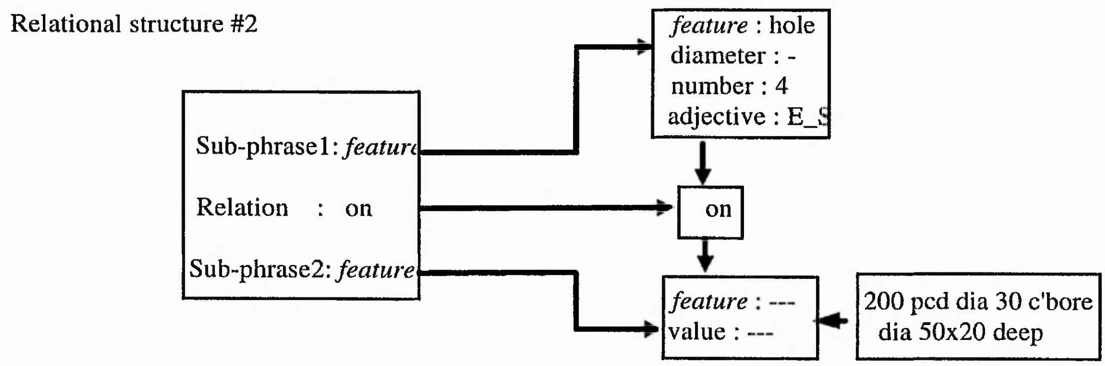
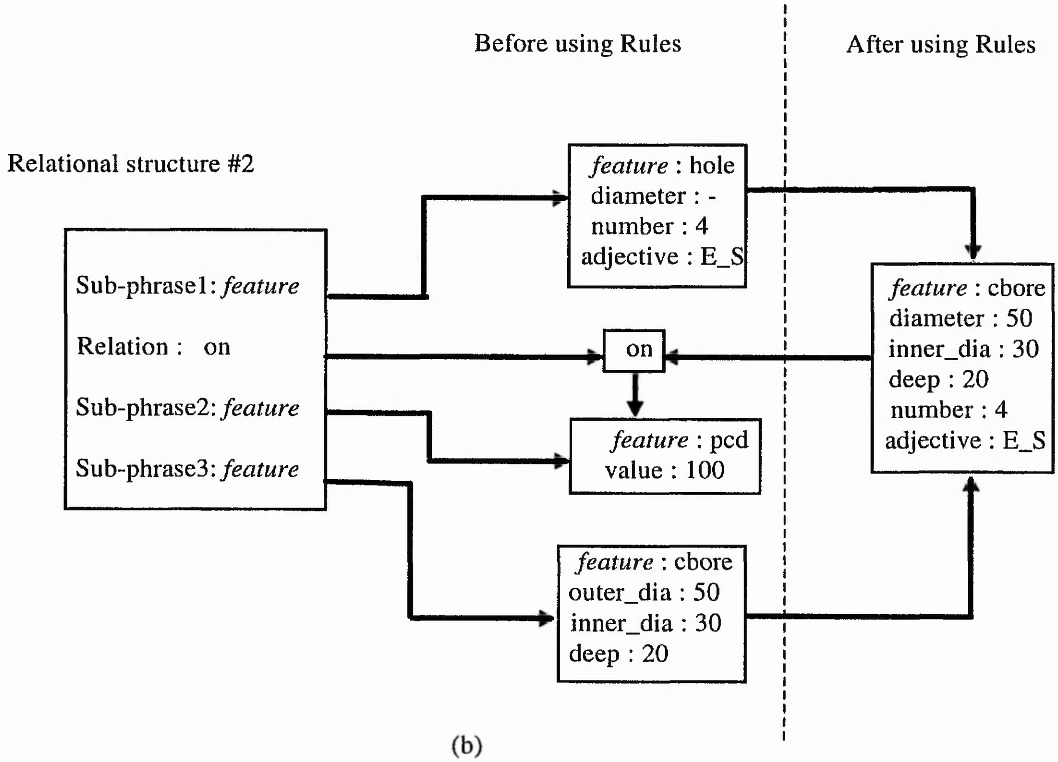


Figure 5.7: A drawing with textual information describing simplified features.



(a)



(b)

Figure 5.8: (a) Relational structure of two Sub-phrases and unrecognised text, (b) Relational structure with recognised text added as a third Sub-phrase.

The interpretation process is driven by logical rules which consist of two parts:

(i) condition, and (ii) inference. A normal rule is often of the form ;

If (condition) then (inference).

The condition clause might be composed of several simple conditions combined together by logical operators such as logical-and.

The Rules defined to recognise more complex phrases, such as "*4 holes equi-spaced on 100 pcd dia 10 c'bore dia 20x15 deep*", are described in Figure 5.9.

```
Begin
    IF prepositional phrase
        THEN identify Sub-phrase1;
            identify Sub-phrase2;
            IF Sub-phrase2 consists of two features
            THEN identify features and their values;
            Segment Sub-phrase2 into Sub-phrase2 and Sub-phrase3;
                IF feature1 and feature3 have a common Template
              Process Relational Structure (feature1 preposition feature2);
                THEN combined the two Templates into one;
                ENDIF
            ENDIF
    ENDIF
END
```

Figure 5.9: Shows the rules for identifying a prepositional phrase such as "4 Holes equi-spaced on 200 pcd dia 30 c'bore dia 50x20 deep".

In a relational template, the relationship made explicit by the recognition of the *features* hole and the pitch circle. Usually pitch circle has only a value indicating to the size of the circle. Any additional information within this sub-phrase (the nominal group) must be related to the first sub-phrase, "*4 Holes equi-spaced*".

The rules will identify the *feature* c'bore and its values from sub-phrase2. This will be related to sub-phrase1 which will generate a meaning of 4 repeated holes of type c'bore and of size indicated by the specified values in the phrase. Processing the knowledge associated with this relational structure indicating to the situation of *counterbore holes on a pitch circle*. Incomplete geometric descriptions of *features* will be pruned and replaced by their full geometric descriptions. On the other hand simplified descriptions of *features* will also be recovered. The geometric descriptions in the three projections will be added to the data file.

5.5 Summary

This chapter has described the extension of the interpretation process to include prepositional phrases. It has been recognised that these phrases are made up of a number of simple phrases such as those which have been described in chapter 4. The interpretation process therefore, treats this type of phrase as a combined one and segments into sub-phrases. Prepositional term has been used to identify this type of phrases and sub-phrases. These sub-phrases then interpreted individually within a relational structure. A relational structure can link and combine a number of templates. This strategy therefore, generates the semantic templates of each individual sub-phrase and a corresponding link with their processing knowledge. Using logic knowledge, these linked templates can be directed to produce a valid interpretation.

Chapter 6

Feature Recognition

6.1 Introduction

Understanding the shape of features is essential. This aspect can help define the correlation between textual descriptions and the geometry, and is important in engineering drawing for recognising *features* based on their properties. The traditional description of engineering drawing represents its geometry as a data structure involving vertices and edges. However, to recognise a feature its geometric description should be extracted from the whole data structure. The problem is then to find a useful description of the *feature* in terms of the shape attributes.

Several approaches have been proposed for the recognition of object from their geometric data. T. C. Woo [Woo82] used convex hull techniques to describe the object as alternating sums of volumes. L. K. Kyprianou [Kyprianou80] and S. M. Staley [Staley83] applied syntactic pattern recognition methods to classify depressions. L. De Floriani [De Floriana87] uses connectivity properties to classify *features* into *PD* features (protrusions or depressions) and *H features* (through holes). While most of the above methods use the boundary description of the object as input, Y. C. Lee and K. S. Fu [Lee87] proposed algorithms for the extraction and unification of some amenities from a CSG tree.

R. Bergevin and M. D. Levine [Bergevin93] introduced Primal Access Recognition of Visual Object (PARVO), a computer vision system that addresses the problem of fast and generic recognition of unexpected 3-D objects from single 2-D view. Trung Tat Pham and Guanrong Chen [Pham93] outlined a linguistic approach to recognise objects and fuzzy logic solution is given for this formulation. Fuzzy logic approach has some advantages over the other approaches, it has the ability to address ambiguity, and the reduction of computer storage requirements for keeping data.

Most of the above approaches are concerned with recognising 3-D objects and their 3-D *features* which have a little in common with the problem of recognising 2-D projections of 3-D *features* of 3-D objects. For a complete interpretation process of text on drawing, a suitable method which is capable of recognising individual "flat" rigid objects with moderate distortion is required. This implies that the recognition method can only recognise 2-D objects or 2-D projections of 3-D objects. Obviously, an engineering drawing describes a 3-D object in multiple views, so all these views need to be learned in order for the method to recognise the object properly.

The previous two chapters dealt with the interpretation of textual information on engineering drawings. Much text on drawings is associated one way or another with *features* as part of the whole object on the drawing. This chapter describes an algorithm for recognising a number of *features* that may be described by a phrase. The identification of simple *features* such as holes is aided by two engineering conventions.

(1) - The centre position of holes must either coincide with the intersection point of two centre lines or the intersection point of centre line with the pitch circle.

(2) - Phrases describing these *features* are usually accompanied by leader lines (arrow lines), which are pointing at and touching the *feature*.

Some other *features* require a general method to locate and recognise them on the drawings, as shown in Figure 6.1. This is because engineering conventions may not be sufficient to identify the position of these *features*. Although in most cases there are leader lines linking the text description to the geometric part, this can help in identifying the area on the projection close to the geometric data associated with the text. An expert method needs to be implemented that uses human interpretation to associate *features* indicated by the phrase with their shapes. This approach therefore, stores descriptions of *features* in a library along with their names. Geometric information can be processed to produce boundaries of unknown shapes. These boundaries can then be matched with the known descriptions of features stored in the library.

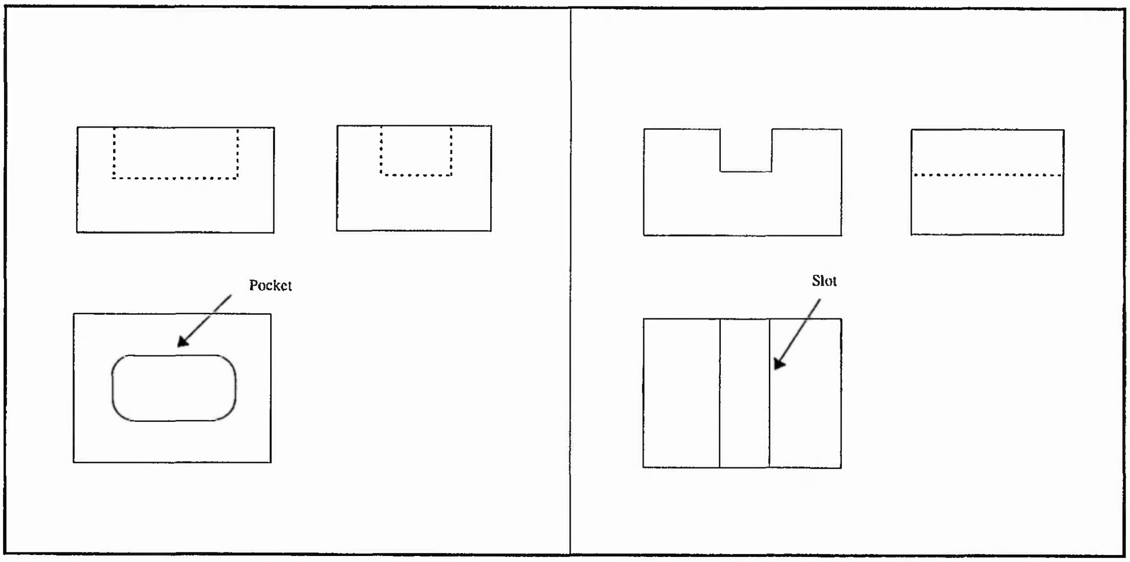


Figure 6.1: Shows two drawings with Rounded Pocket and Through Slot *features*.

In cases where a *feature* described by the phrase can not be associated with a location on the drawing, the interpreter associates the name of the *feature* from the phrase with its description in the library. This description is then matched with all the boundaries found in a projection. The boundary with the nearest approximation to the selected *feature* is chosen to have the shape of the *feature*. The information obtained from the phrase is used to check the consistency of this shape, and any discrepancy or distortions can then be rectified.

As described in Chapter 1, engineering drawings are descriptions of objects in three orthographic projections, and this indicates that objects and their *features* may have more than one shape on the drawing describing them. The recognition process need to learn all the shapes of *features* but may only need one shape to recognise the feature. For instance, as we have mentioned in the last chapter, holes are represented by a circle in one view and rectangles in the other views. This method clearly depends upon the knowledge provided by the descriptions of the *features* that are included in the library.

6.2 Feature Classification

Interpreting text on engineering drawings requires the association of the text to the part of the drawing that the text is describing. Some times leader lines will lead the interpreter to the geometric part which is described by the text. Much of the text on engineering drawings as we understand is associated with features. It is therefore, necessary to implement a general algorithm to recognise features on a drawing. The recognition algorithm comprise two processes: learning and recognition. In the learning process the algorithm is made aware of the *feature* it is expected to recognise by presenting it with the description of the *feature*. In the recognition process, the algorithm confronted with unknown *features* (contours), and it is expected to classify one of these *features* as belonging to the same class as the description it has been given in the learning process.

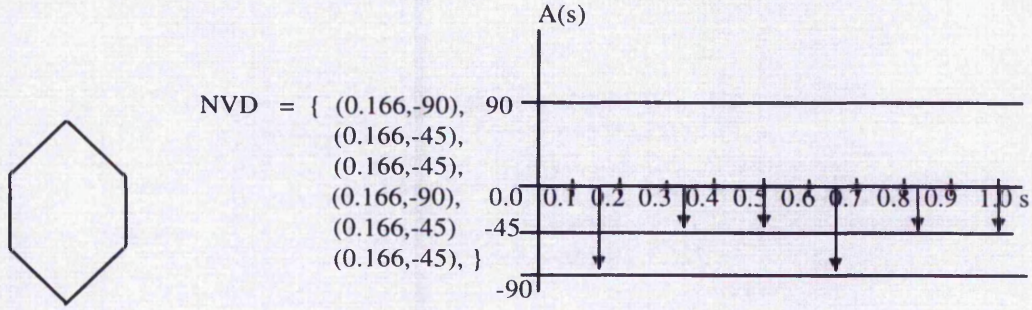
The recognition method [Parra-Loera92] has been adapted which uses Similarity Measures (SM) that measure the resemblance among *features* and decision functions that assign *features* to prototype classes influence the overall robustness of the classification process.

Many different measures can be applied to simplify a shape. The idea is that if two shapes have measures which are close in value then they will be similar in shape. Edges of the drawings that have a common ends (nodes) or within a small tolerance are connected. Contours can be found by starting with an edge, traverse to the next connected or nearest edge until it reaches the starting edge. This process repeated for all unconnected edges to derive all contours in a particular projection. These contours can be considered as the candidate *features* on the projection. The algorithm then process these unknown *features* to generate a shape descriptor. These descriptors are matched with the descriptor of the known *feature* stored in the library. A match can clearly classify this descriptor as being the indicated *feature* and its size and shape can be correctly generated. Using the values of this descriptor as well as the information in the knowledge base its shape in the other projections can be determined.

The Normalised Vertex Descriptors (NVD) representation can be described as a chain code resulting from the encoding of the normalised run-length d_i and the angle a_i between successive edges of the contour. Normalised run-length is obtained by first dividing the individual edge lengths of the contour by the total length (so that the total length now adds up to 1). The result is encoded into tuples formed by the normalised run length of the edge and the angle between successive edges;

$$NVD = \{ (d_1 , a_1), (d_2 , a_2), (d_3 , a_3), ... (d_n , a_n) \}.$$

Traversing anticlockwise along the contour, this information can be represented as a function of the normalised length (see Figure 6.2). Angles between successive edges of the contour are represented as impulses separated by the normalised length of the edges. A concave angle is a positive impulse and a convex angle is a negative impulse. From this it can be seen that the NVD representation is a periodic function of normalised length with period of one. This therefore, is used to derive local descriptors and similarity measures for the algorithm.



(a) diamond. (b) NVD chain code. (c) NVD plot.

Figure 6.2: Illustration of the NVD for a diamond shape.

The library of known *features* is organised into a Relational Database. The relation *feature* is defined as a set of M-tuples (with N attributes) where M is the number of *features* in the database. The relation can be express using the following notation:

FEATURE (NAME, CM, NSA, NCA, NVD),

where

NAME	*Feature* name,
CM	Complexity Measure of the *Feature*,
NSA	Number of Sharp Angles present in the *Feature*,
NCA	Number of Concave Angles present in the *Feature*,
NVD	Normalised Vertex Descriptor of the *Feature*.

The complexity measure CM of the *feature* is defined as the summation of the absolute values of the angles in the NVD representation, i.e.,

$$\mathbf{CM} = \sum_{i=1}^{k} | a_i |$$

where k is the number of links and the a_i's are the angles in the NVD representation.

The Number of Sharp Angles (NSA) is defined as the count of all angles in NVD whose magnitude is greater than some threshold **th1**, i.e.,

$$\mathbf{NSA} = \sum_{i=1}^{k} A_i \qquad \text{where} \qquad A_i = \begin{cases} 1 & \text{if } |a_i| > th1 \\ 0 & \text{otherwise} \end{cases}$$

The Number of Concave Angle (NCA) is defined as the count of all positive angles which are greater than some threshold **th1** in the NVD's, and it can be expressed as:

$$\mathbf{NCA} = \sum_{i=1}^{k} C_i \qquad \text{where} \qquad C_i = \begin{cases} 1 & \text{if } a_i > th1 \\ 0 & \text{otherwise} \end{cases}$$

The threshold value **th1** used is 20° and is chosen to minimise the effect of the distortion caused by scanning.

Computation of distance measures (D_i) directly on the Normalised Vertex Descriptors (NVD) representations of *features* may not be very effective because of the discontinuous nature of the NVD's. The representation is somewhat influenced by distortions present in the contour. Therefore, the function that minimises the effects of these distortions is described in (1).

$D = \sum_{i=1}^{k} d_i a_i$ where d_i are the normalised length of edges of feature,

and a_i are angles between successive edges.

for (i = 1,...., k) number of links. (1)

Once these functions are computed, a template matching algorithm operating on these functions can be used to determine the degree of similarity among *feature* descriptions. The classification algorithm is described in Figure 6.3.

Procedure CLASSIFY

```
BEGIN
    COMPUTE       boundaries of unknown Features
    COMPUTE       NVD representation of unknown Features.
    EXTRACT       "candidate" Feature from library.
    COMPARE       candidate's amenities with unknown features.
    COMPUTE       features (attributes).
    IF            set empty
          EXIT          no match.
    ELSE
              IF   set more than one candidate
                            COMPUTE similarity measure (distance).
                            SELECT      minimal distance.
              CLASSIFY          unknown Feature.
END
```

Figure 6.3: *Feature* Classification Algorithm.

The implementation has only five attributes defined, however, others may be defined for future work. New attributes should be selected based on their ability to reduce the set of candidates.

The constrained set, the relation CANDIDATES which is defined as

CANDIDATES (NAME, DESC)

A number of RULES have been implemented in order to minimise the set of candidates which map on the one *feature* from the library. The process can be outlined as follows:

1. Apply a RULE on candidate set operating on CM (Complexity Measure) such that:

$CM_i > | CM_{UK} - t_1 |$, i.e.,

RULE1 = *features*[$CM_i > | CM_{UK} - t_1 |$].

Where CM_i is Complexity Measure for Known *feature* and CM_{UK} is Complexity Measure for Unknown Candidate *features* under investigation, t_1 is a small tolerance.

2. Apply a RULE on RULE1 operating on NSA (Number of Sharp Angles) such that:

$| NSA_i - NSA_{UK} | < t_2$, i.e.,

RULE2 = RULE1[$| NSA_i - NSA_{UK} | < t_2$].

Where NSA_i is the Number of Sharp Angles for the Known *feature* and NSA_{UK} is the Number of Sharp Angles for the Unknown Candidate *features* under investigation, t_2 is a small tolerance.

3. Apply a RULE on RULE2 operating on NCA (Number of Concave Angles) such that:

$| NCA_i - NCA_{UK} | < t_3$, i.e.,

RULE3 = RULE2[$| NCA_i - NCA_{UK} | < t_3$].

Where NCA_i is the Number of Concave Angles for the Known *feature* and NCA_{UK} is the Number of Concave Angles for the Unknown Candidate *features* under investigation, t_3 is a small tolerance.

4. Select a Candidate *feature* from RULE3 on Distance Measures D on the NVD representation of *features* such that:

CANDIDATES = Min [RULE3[$| D_i - D_{UK} |$]]

Where D_i is the known *feature*, and D_{UK} for the unknown *feature* and D_i defined by equation (1).

The t_j's are tolerances to adjust the candidate set. They provide a way for compensating for distortions caused by scanning. The bigger the tolerances the more tolerant the algorithm is to distortions. In this recognition algorithm, the following values for t_j's are used, $t_1 = 20^o$, $t_2 = 3$ and $t_3 = 3$.

RULE1 eliminates those *features* that are less complex than the known *feature*. RULE2 eliminates those that do not have similar number of prominent vertices. RULE3 eliminates those that do not have similar concave vertices. The only information needed after this operation is the NVD representation to compute the similarity measure to classify the *feature*.

6.3 Classification Method

Initially the descriptions of all expected *features* are stored in the library, Table 6.1. Figure 6.4 displays a pictorial descriptions of these *features*. Knowing the name of *feature* which is indicated by the phrase, the algorithm then, first searches the library to retrieve its description and stores it in a buffer. Once the *feature* is retrieved from the library, a boundary tracing algorithm manipulates the geometric outlines of the projection associated with text, and generates a polygonal approximation of candidate *features*. The polygonal approximation of these candidate *features* are then converted to the NVD representations of the *features*.

The classification algorithm computes the Similarity Measure of candidate *features* and uses the stored attributes of known *feature*. Finally, a candidate *feature* with the smaller similarity factor is assigned to the known *feature*. This algorithm also provides a learning mode to incorporate new prototype classes to the library of known *features* which also requires user interface to enter *feature*'s name. Figures 6.5-6.7 show the result of the recognition algorithm for the unknown *feature* to be classed as *feature9* in the library.

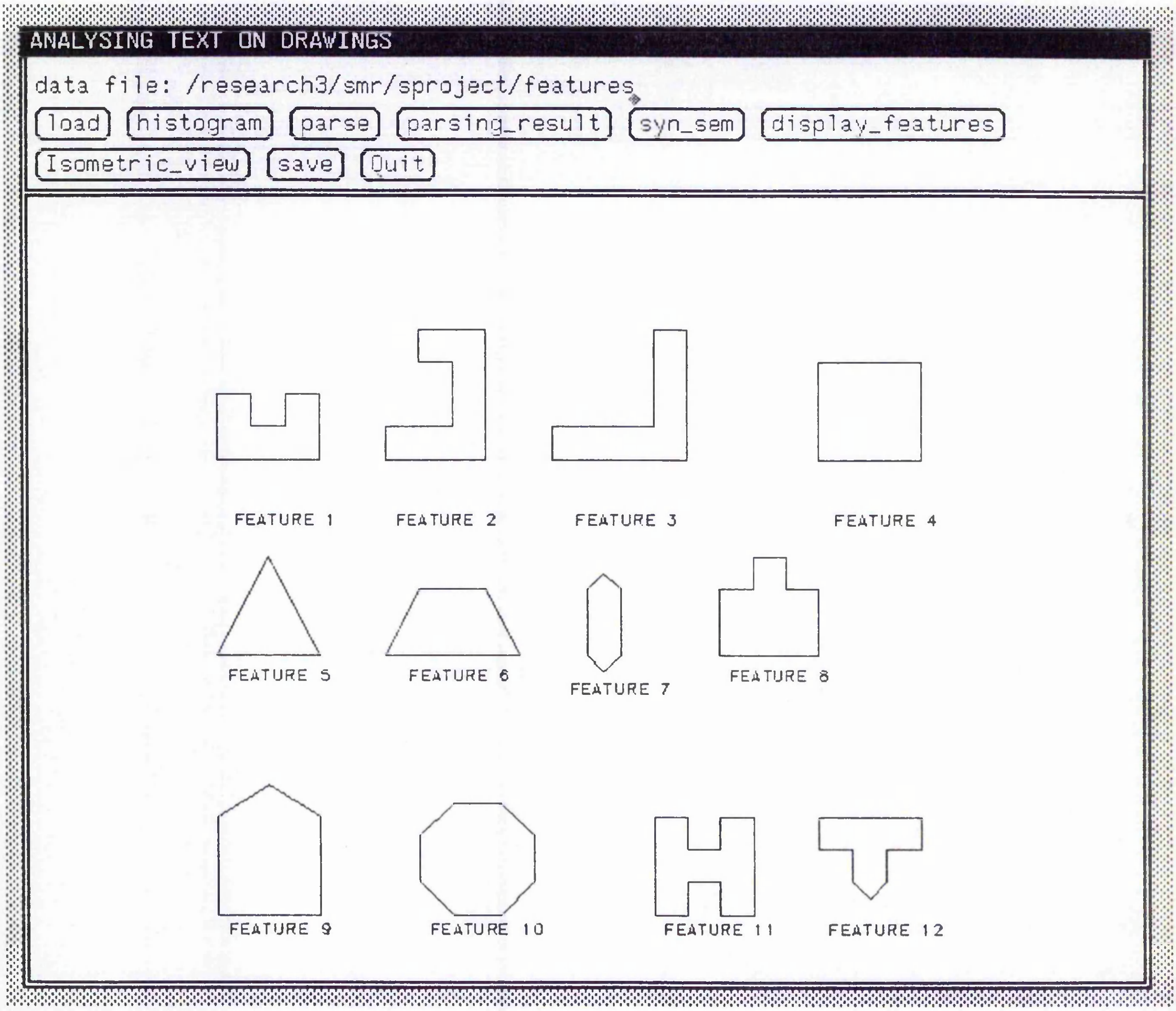


Figure 6.4: *Features* stored in library.

Name	CM	NSA	NCA	DESCRIPTION
Feature1	720	8	2	(0.083 -90), (0.166 -90), (0.250 -90), (0.166 -90), (0.083 -90), (0,083 90), (0.083 90), (0.083 -90)
Feature2	720	8	2	(0.125 -90), (0.0625 -90), (0.0625 90), (0.125 90), (0.125 -90), (0.0625 -90), (0.1875 -90), (0.250 -90)
Feature3	540	6	1	(0.083 -90), (0.166 90), (0.166 -90), (0.083 -90), (0.250 -90), (0.250 -90)
Feature4	360	4	0	(0.250 -90),(0.250 -90),(0.250 -90), (0.250 -90)
Feature5	360	3	0	(0.344 -116), (0.309 -117), (0.347 -127)
Feature6	360	4	0	(0.166 -45), (0.250 -135), (0.333 -135), (0.166 -45)
Feature7	360	6	0	(0.125 -30), (0.250 -30), (0.125 -120), (0.125 -30), (0.250 -30), (0.125 -120)
Feature8	720	8	2	(0.071 -90), (0.071 90), (0.071 -90), (0.214 -90), (0.214 -90), (0.214 -90), (0.071 90),(0.071 -90)
Feature9	360	5	0	(0.20 -30), (0.20 -90), (0.20 -90), (0.20 -30), (0.20 -120)
Feature10	360	8	0	(0.125 -45), (0.125 -45), (0.125 -45), (0.125 -45), (0.125 -45), (0.125 -45), (0.125 -45), (0.125 -45)
Feature11	1080	12	4	(0.083 -90), (0.166 -90), (0.083 -90), (0.042 90), (0.083 90), (0.042 -90), (0.083 -90), (0.166 -90), (0.083 -90), (0.042 90), (0.083 90), (0.042 -90)
Feature12	720	9	2	(0.250 -90), (0.042 -90), (0.083 90), (0.166 -30), (0.083 -120), (0.083 -30), (0.166 90), (0.083 -90), (0.042 -90)

Table 6.1: Relation *Features* in (LIBRARY).

Name	CM	NSA	NCA	DESCRIPTION	D_{UK}
Unknown	360	5	0	(0.207 -30), (0.1887 -90), (0.1887 -90), (0.1887 -25), (0.226 -125)	73.1435

Figure 6.5: Unknown *feature* under investigation.

Figure 6.5 contains the description of the unknown *feature* under investigation, rules 1-4 are used to narrow the matching of the unknown with a candidate in the library. Applying RULE1 (Complexity Measure) produces a list of all the *features* in the Library as candidate *features*. This new list has then been tested for RULE2 and RULE3, the new candidates *features* are shown in Figure 6.6.

In cases where more than one candidate *features* are generated, Distance Measure is used to select a candidate *feature* which has a minimum value for the Distance Measure which indicates close resemblance of the candidate with that of the unknown *feature*, see Figure 6.7.

Name	CM	NSA	NCA	DESCRIPTION
Feature3	540	6	1	(0.083 -90), (0.166 90), (0.166 -90), (0.083 -90), (0.250 -90), (0.250 -90)
Feature4	360	4	0	(0.250 -90),(0.250 -90),(0.250 -90), (0.250 -90)
Feature5	360	3	0	(0.344 -116), (0.309 -117), (0.347 -127)
Feature6	360	4	0	(0.166 -45), (0.250 -135), (0.333 -135), (0.166 -45)
Feature7	360	6	0	(0.125 -30), (0.250 -30), (0.125 -120), (0.125 -30), (0.250 -30), (0.125 -120)
Feature9	360	5	0	(0.20 -30), (0.20 -90), (0.20 -90), (0.20 -30), (0.20 -120)

Figure 6.6: Relation after applying RULE1, RULE2 and RULE3.

Name	CM	NSA	NCA	DESCRIPTION	D_i	$\|D_i - D_{UK}\|$
Feature3	540	6	1	(0.083 -90), (0.166 90), (0.166 -90), (0.083 -90), (0.250 -90), (0.250 -90)	89.82	12.06
Feature4	360	4	0	(0.250 -90), (0.250 -90), (0.250 -90), (0.250 -90)	90	18.0
Feature5	360	3	0	(0.344 -116), (0.309 -117), (0.347 -127)	120.126	48.126
Feature6	360	4	0	(0.166 -45), (0.250 -135), (0.333 -135), (0.166 -45)	93.645	21.645
Feature7	360	6	0	(0.125 -30), (0.250 -30), (0.125 -120), (0.125 -30), (0.250 -30), (0.125 -120)	52.5	19.5
Feature9	360	5	0	(0.20 -30), (0.20 -90), (0.20 -90), (0.20 -30),(0.20 -120)	72	1.14

Figure 6.7: Classification results for the unknown feature.

6.4 Summary

The method described here [Parra-Loera92] for the recognising of *features*, is the most convenient for recognising 2-D *features* of the 3-D objects. As the engineering drawing describes 3-D objects in a number of 2-D orthogonal projections, an approach is required to recognise the *feature* in the projection referenced by the phrase and derive or check the consistency of the *feature's* outline in other projections.

Related edges of the *feature* on drawing are connected to generate boundaries of shapes. A number of measures are generated describing these shapes. Descriptors of these boundaries are matched with a library of *features* to recognise the shape of the feature indicated by the phrase. This method also allows for some distortion which is one of the problems that this project is dealing with.

The only restrictions of this method are that it recognises only individual non-touching *features* and only *features* which are represented or approximated by polygons. These restrictions can be relaxed by extending the recognition algorithm to include interacting simple *features* [Meeran93].

The objective of this work, is to outline an approach that helps to produce an automatic interpretation of text on engineering drawings. This approach is implemented to recognise a general shape with the restrictions indicated above.

The author has identified and implemented an approach which is capable of recognising the general shape of *features* on engineering drawings. This approach is designed to use the feature's name indicated by the phrase on drawing and with knowledge provided about the *feature*, its geometry can be recognised from the drawing. The consistency of the information recognised can be checked against the information provided in the text and the knowledge provided by the knowledge base.

The procedure of recognising *features* on drawings is to mimic human recognition of shapes. Human will recognise a shape due to previously learning and remembering the shape, while this approach uses the knowledge provided about the shape and matching this shape with the geometric data on the drawing to recognise it.

Chapter 7

Implementation Results

This chapter outlines the areas where the algorithms in this thesis are most useful to improve the interpretation of engineering drawings. Some examples are presented to demonstrate the efficiency and the capability of the described approach. These results demonstrate the importance of textual information in order to produce a complete geometric representation of the drawing. Strengths and possible further development of the interpretation method are discussed later in this chapter.

7.1 Overview of the Interpretation Method

The strategy used in this thesis for the interpretation of engineering drawings is concentrated on the utilization of textual information in order to improve the reconstruction of 3-D models. This information purposely put on the drawing by the designer to complement the geometric information. Thus, ignoring this information may result in ambiguous interpretation especially when the information on a paper drawing has been translated into a computer format. Translation of geometric information from paper drawing into a computer format may also relinquish some of the geometric information.

The interpretation method implemented in this thesis is directed towards an automatic approach to the interpretation of engineering drawing including solving possible ambiguities. Initially, the scanned 2-D geometric information (nodes, lines and arcs) are entered. This 2-D geometric information has been produced from scanning and vectorising simple engineering drawings on paper.

At this stage, the 2-D geometric data file is in no particular logical order, and no explicit information is present regarding their connectivity. Therefore, this information first has been sorted into three groups, each belong to one of the three views before further processing can take place. And second each group has been tested for line classification, where lines are assigned a type identifier as being solid, hidden or centre lines. However, this did not cover more complex cases as it is still under research.

Dring the next stage (interpretation of high level information), a number of common annotations have been added on to the data file. The interpretation uses Syntactic and Semantic knowledge to associate these annotations to their corresponding geometric parts. This process therefore, checks the consistency of the information provided by the annotations and geometric information. Any discrepancy between these two parts indicates a possible distortion or simplification of the geometric information and the interpretation algorithm was made to recover them. This approach provides explicit geometric information for the 3-D reconstruction process.

Finally, to complete the method an algorithm for *feature* recognition has been investigated which currently gives a good results for simple shapes and can be used as a foundation for a more powerful one that can be extended in future work. Figure 7.1 shows a flowchart of the interpretation method implemented in this research.

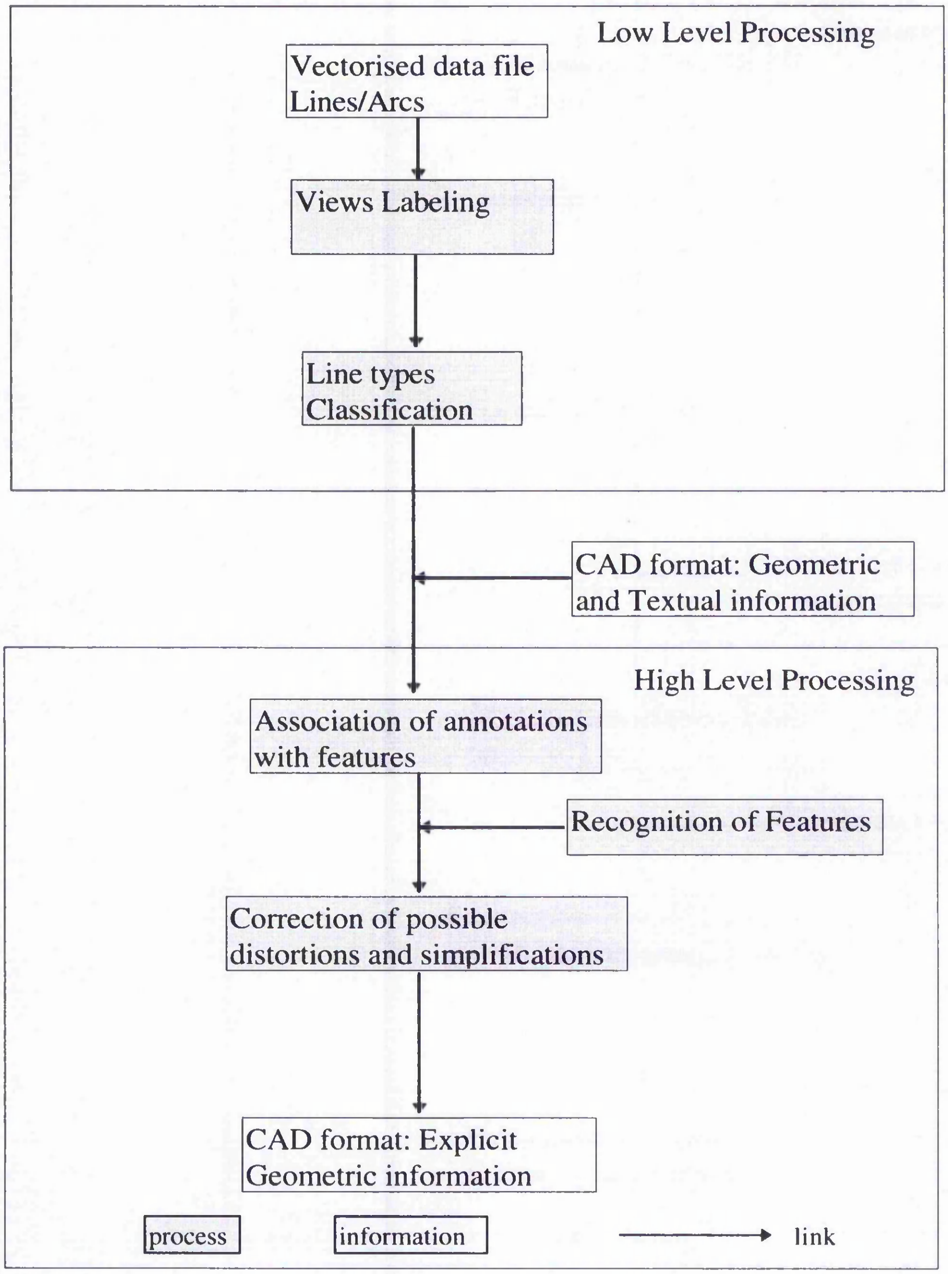


Figure 7.1: Flowchart of the interpretation method.

7.2 Simplification of Repeated Feature

A number of conventions are used on engineering drawings for representing repeated *features*. For example when splined shafts are drawn, usually only two or three splines are drawn and the remaining are represented by two concentric circles indicating the depth of the splines. Similarly when a number of identical holes are present only one hole may be drawn as a circle and the remaining may be indicated by intersecting centre lines and a phrase describes the information of these holes as shown in Figure 7.2.

In this example holes which are projected by circles in one view would therefore need to be inserted in the correct locations and hidden lines added to the other views where necessary before 3-D reconstruction process could take place.

Matching process which correlates the geometric information in the three views will not be able to assemble all *features* of the object. This is because the geometric information does not fully describe the object on the drawing. Much of previous research has concentrated on producing a consistent 3-D model from complete 2-D projections rather than a real 2-D engineering drawing.

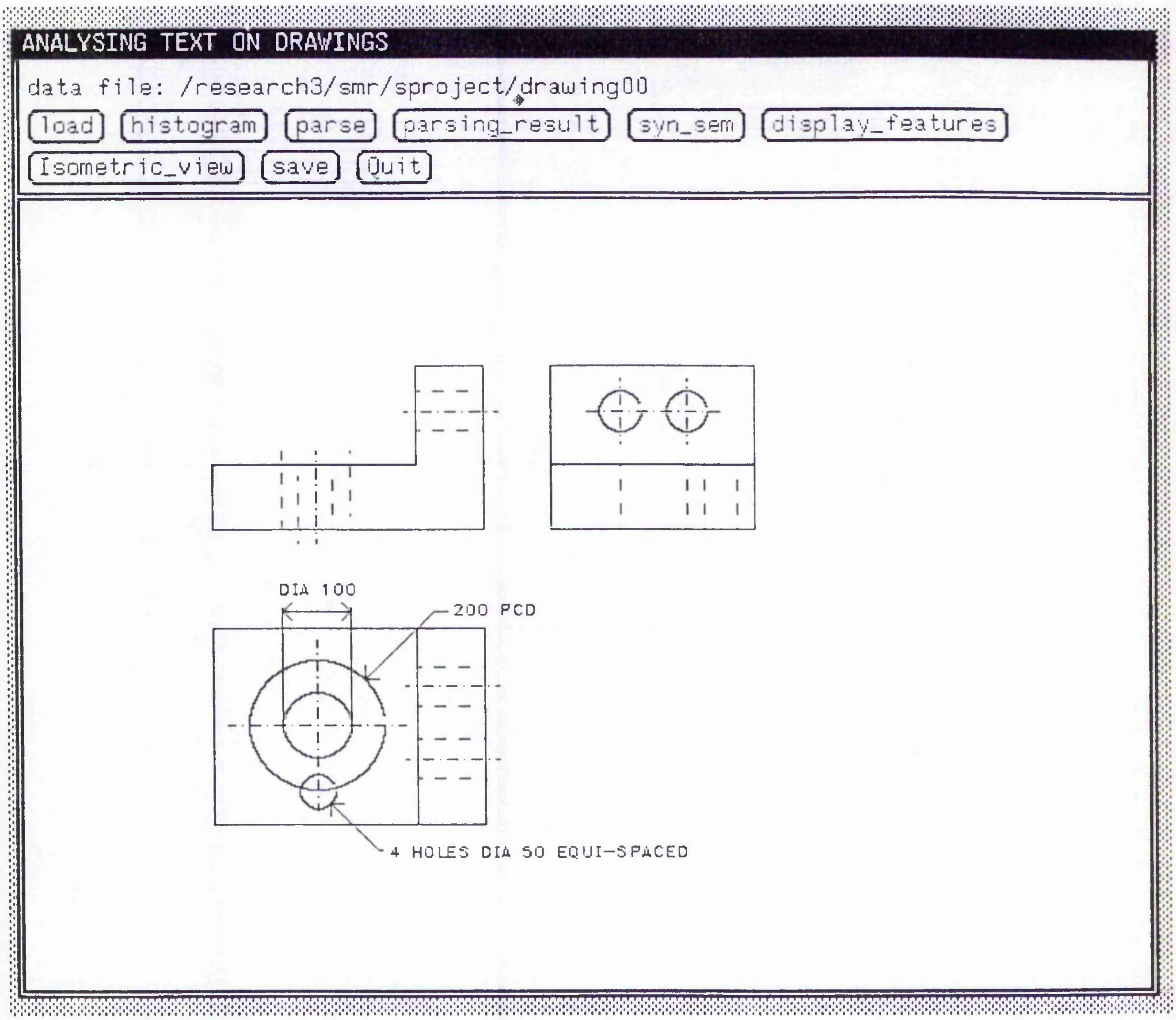


Figure 7.2: An example of an Engineering Drawing with missing features. The text in the Top view describes 4 holes but only one is drawn. This drawing cannot be interpreted correctly without using the textual (annotations) information.

Figure 7.2 shows an example of a drawing containing simplifications. Only one of the 4 holes is shown and additional information related to these holes is indicated by a phrase. Applying a reconstruction method on this kind of drawing which uses only the outline can produce an ambiguous result. Since the reconstruction method correlates geometric information from all views, only *features* which have outlines in all views can be recognised. In the above example, only one hole has a complete description and therefore, the reconstruction of 3-D model of this drawing may produce an object with only one hole present on it.

The interpretation of textual annotations adds an extra dimension to the interpretation of engineering drawings. This has been achieved by understanding the technical details of engineering drawings. Thus, simplified or distorted *features* can often be made explicit.

The method of textual understanding starts by dividing annotations on engineering drawings into smaller items (tokens), each which can be matched to an entry in a look up engineering dictionary. A successful match for a token with an entry in the dictionary, will assign that token to a particular category. The categorization of all tokens in the phrase translates the phrase into a Syntactic Net. This Net consists of a root node linked to a set of elements by specific relations. The phrase '4 HOLES DIA 50 EQUI-SPACED', from Figure 7.2, is first divided into five tokens with types *Number*, *Feature*, *Parameter*, *Value* and *Adjective*. The Syntactic Net corresponding to this phrase has its root node occupied by the *Features* token and the relations 'feature, 'number', 'parameter' and 'adjective' will link other tokens to the root node.

Also there is a 'value' relation which links the phrasal type *Value* with the *Parameter*, as shown in Figure 7.3.

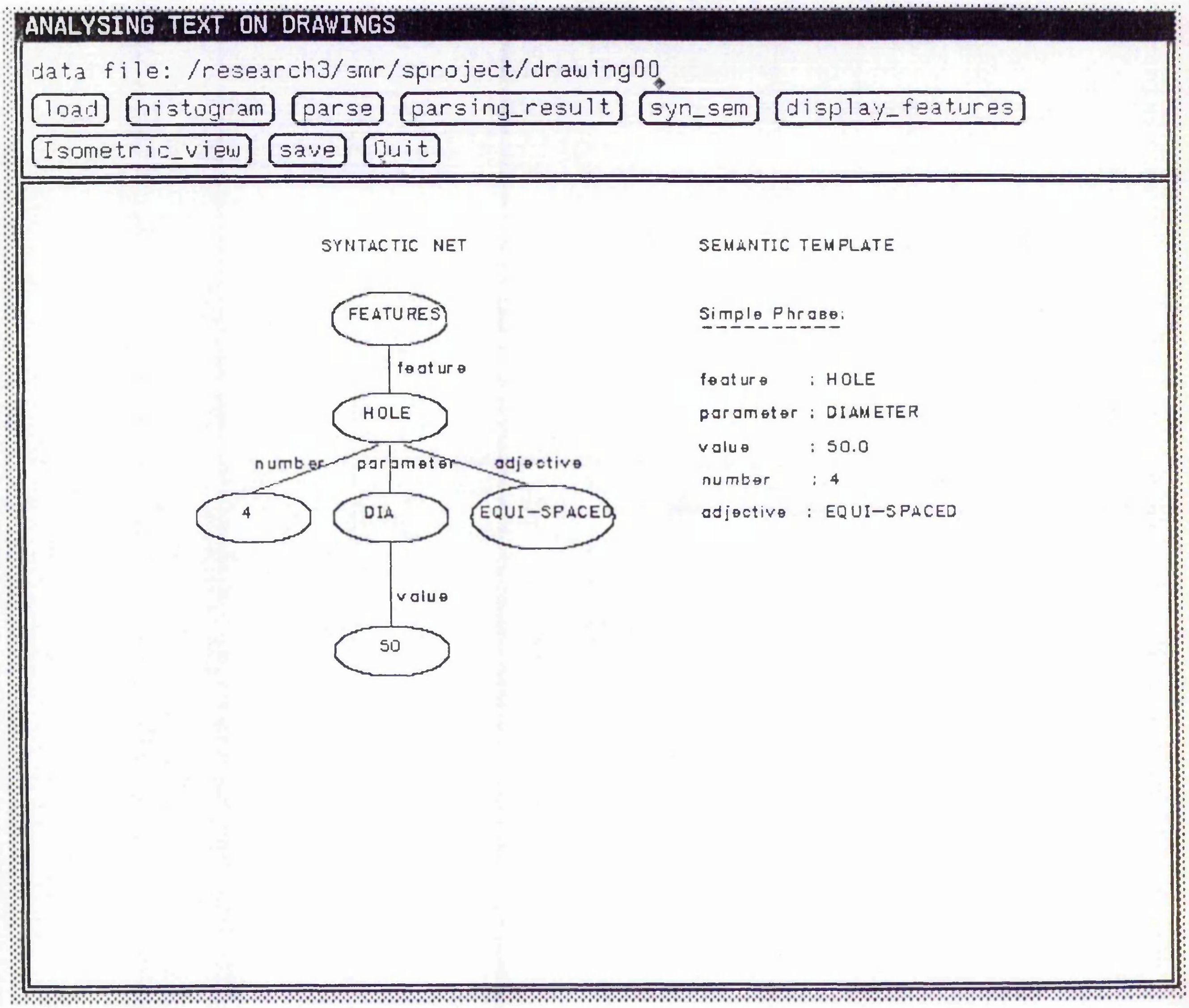


Figure 7.3: A Syntactic Net and Semantic Template for the phrase '4 HOLES DIA 50 EQUI-SPACED'.

The relationships between the tokens are made explicit. This is especially important when processing more complex phrases which may involve several *features* and many parameters and their values which must all be associated correctly with the *features*.

The numerical tokens are assigned to the number or value categories by a process which uses the position in the phrase (exploiting the fact that a number normally precedes the *feature*). Numerical category is terminal in which it terminates on the identification of being either a value or a number. Parameter category defined by a list of parameters, and in the above example, *Diameter* is specifically mentioned in the phrase. Whilst, adjective category defined by a list of processes, each describes a procedure in which *features* are related.

Values for each slot in the Template are either found from the Syntactic Net, as in the case of diameter, or searched for in the drawing, as for depth in the example of Figure 7.2. Conventions such as pitch circle and centre lines play important role in locating the positions of holes on the drawing.

Once all circles (projection of hole) have been located on the drawing, the EQUI-SPACE procedure then checks the positions of these circles which should be at an equal distances. Hidden lines in other views are added which then complete the geometric description of all holes. Hidden lines for the initial hole are identified and the holes are assumed to be penetrating through the surface. The expectation therefore, is that the ends of the hidden lines must be intersecting two lines which represent the top and bottom surfaces of the object. The procedure then checks whether these hidden lines represent the true depth of the hole or not. This has been achieved by finding all lines intersected or very close to be intersected by these hidden lines.

Hidden lines meeting lines at both ends are assumed to have the true depth of the hole. Otherwise, they are assumed to be incorrect and their ends are either clipped or extended to touch the nearest intersection lines as shown in Figure 7.4. From this Figure, the only lines that hidden lines (a, b, c and d) can be intersecting are lines i and j. Hidden line 'a' indicates that one end has met line j, but the other end has crossed line i. This error is corrected by clipping the top extended end of hidden line 'a' and make it touch line i. Other erroneous cases are indicated by hidden lines 'b' and 'd', where an end of a hidden line was too short to meet the crossing lines.

These are corrected by extending hidden lines to meet the nearest intersection line. Hidden line 'b' is extended to meet line i and hidden line 'd' extended to meet line j, also other end of hidden line 'd' is clipped to meet line i. Hidden line 'c' in this example indicates the true depth of the penetrating holes which can be used to create hidden lines for other holes.

All simplifications on the initial drawing are then made explicit, also an isometric view of these features is shown in Figure 7.5.

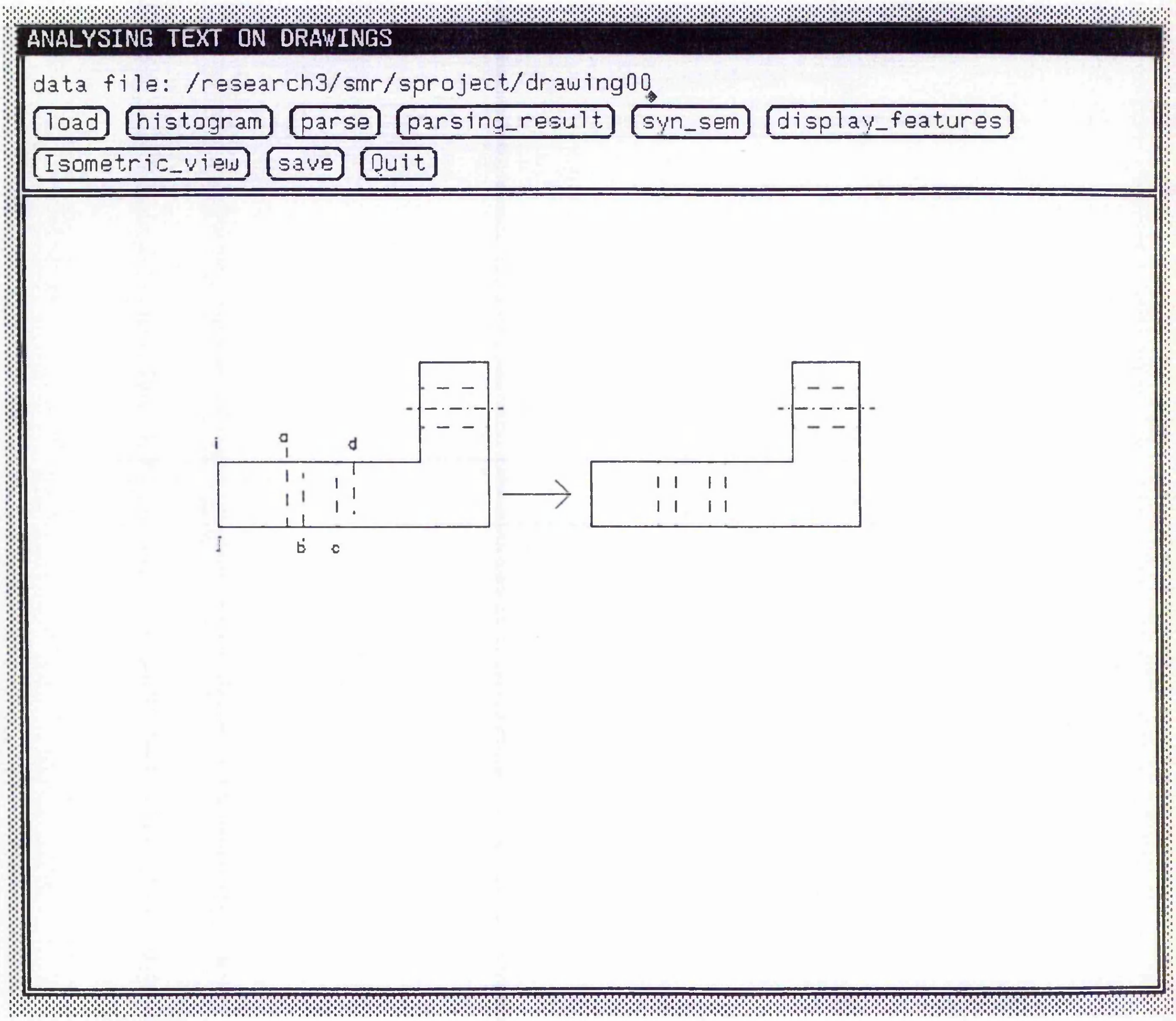


Figure 7.4: Detection and correction of hidden lines which represent the true depth of penetrating holes.

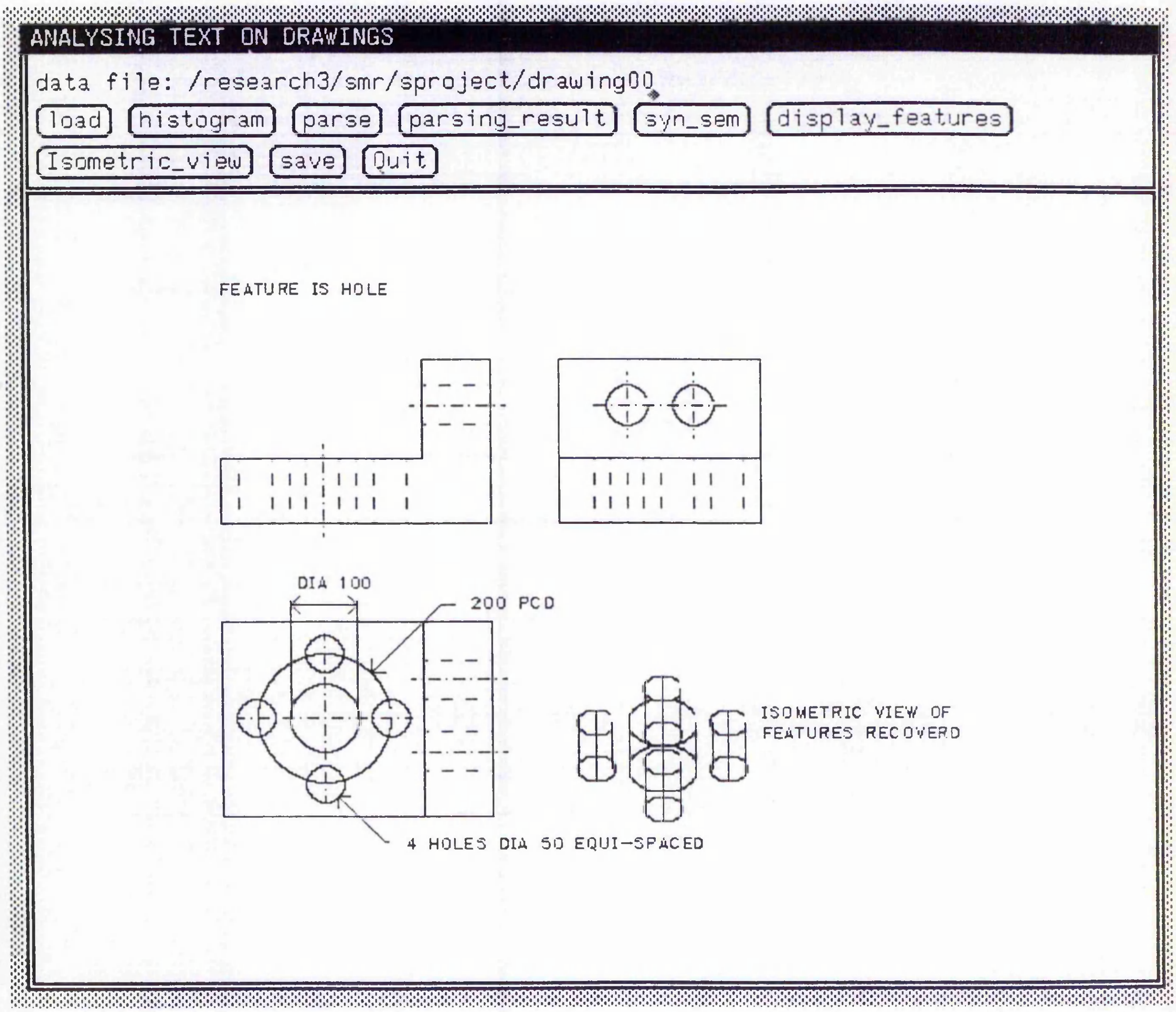


Figure 7.5: Shows the geometric information explicitly describing the object on the drawing. This made possible by interpreting textual annotations.

7.3 Distorted Features

Translating the information of engineering drawing from paper to a suitable computer format is liable to cause some distortions. This may be further exacerbated during preprocessing of the data. Faded lines on paper drawing may have been interpreted by the scanning and vectorising processes to represent a number of short lines. Parts of small circular arcs may also be interpreted as short lines, and small lines may even be completely missing. Figure 7.6 shows a different version of the drawing from figure 7.2 where the single circle representing one of the 4 holes is now distorted. A direct implementation of the bottom-up approach on this representation will generate an incomplete 3-D model. Thus, distorted *features* may not be recognised as such and therefore may be excluded from the reconstruction. In this situation textual annotations can help to rectify the distorted *feature*. Distortions occurring on *features* that are not associated with annotation have not been included in this implementation results. For instance, the distortions of the outlines of an object in one view require the identification of the object together with stored knowledge about the object to be able to rectify these distortions. These cases may be considered for a future work.

Drawings with simplifications and distortions, require more information than just relying on the geometry to produce a consistent 3-D model. Thus, in order to have a better chance of producing a consistent 3-D model, it is necessary to incorporate the interpretation of textual annotations on the drawing.

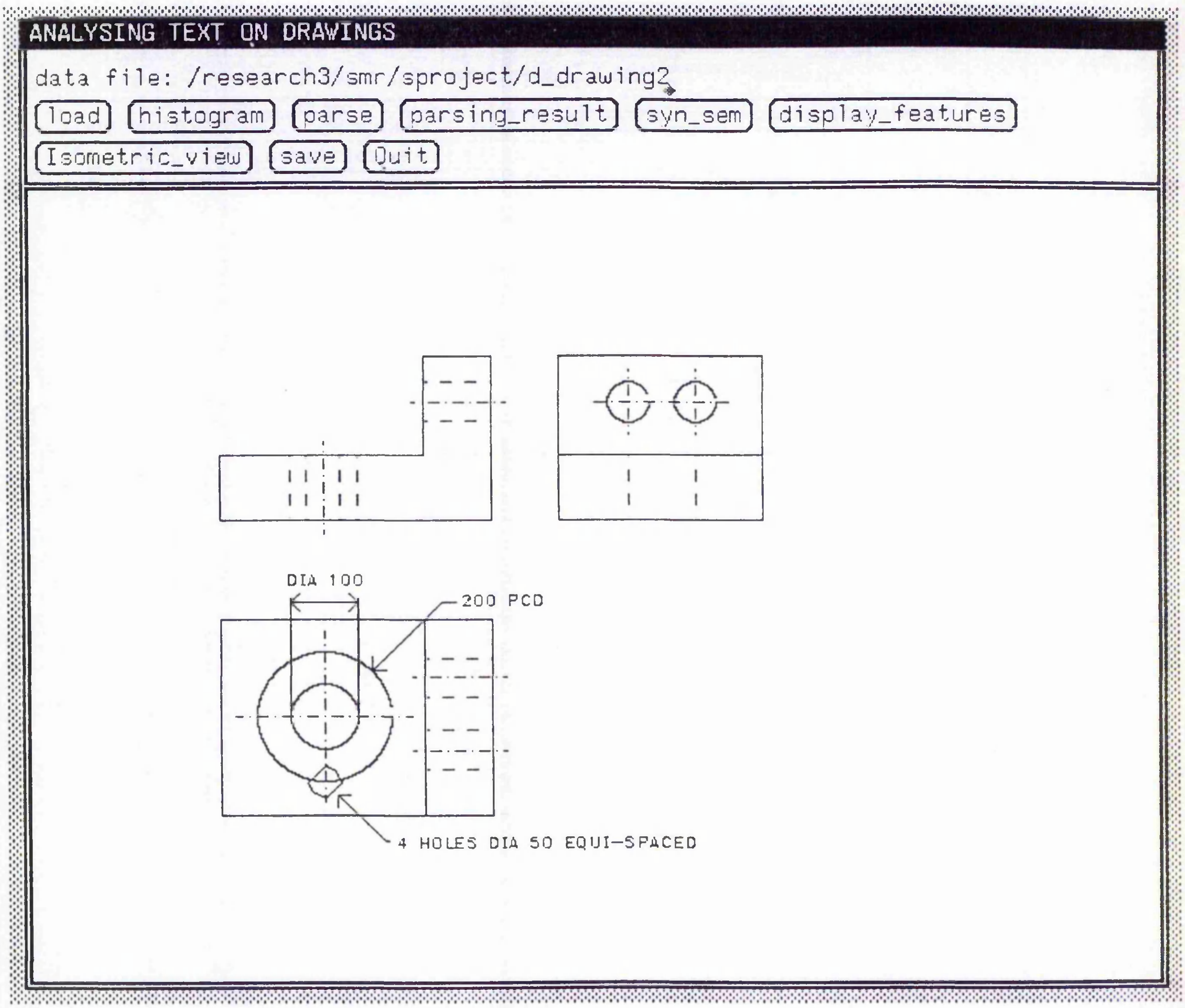


Figure 7.6: An example of an Engineering Drawing with distorted feature in the Top view.

The analysis of the phrase '*4 HOLES DIA 50 EQUI-SPACED*', leads the interpretation process to expect to find at least one of the four holes indicated by the phrase. Also from the Semantic Template of this phrase, the size of these holes and the relation between them are known.

Despite the distorted *feature* (hole), there may still be sufficient evidence to indicate the position of the hole on the drawing. The searching procedure uses convention lines such as centre lines and pitch circles. The intersection points of centre lines or intersection points of centre line with pitch circle are checked for the existence of these holes. Leader lines have also been used to narrow the area at which the search can be conducted to locate the first hole.

A number of criteria have been used to locate the hole. Once the intersection points of a centre line and the pitch circle have been found, these points are tested for a possible existence of holes on them. The size of the hole from the phrase is used to impose an annulus on each of these intersection points and one annulus covers the presumed hole as shown in Figure 7.7.

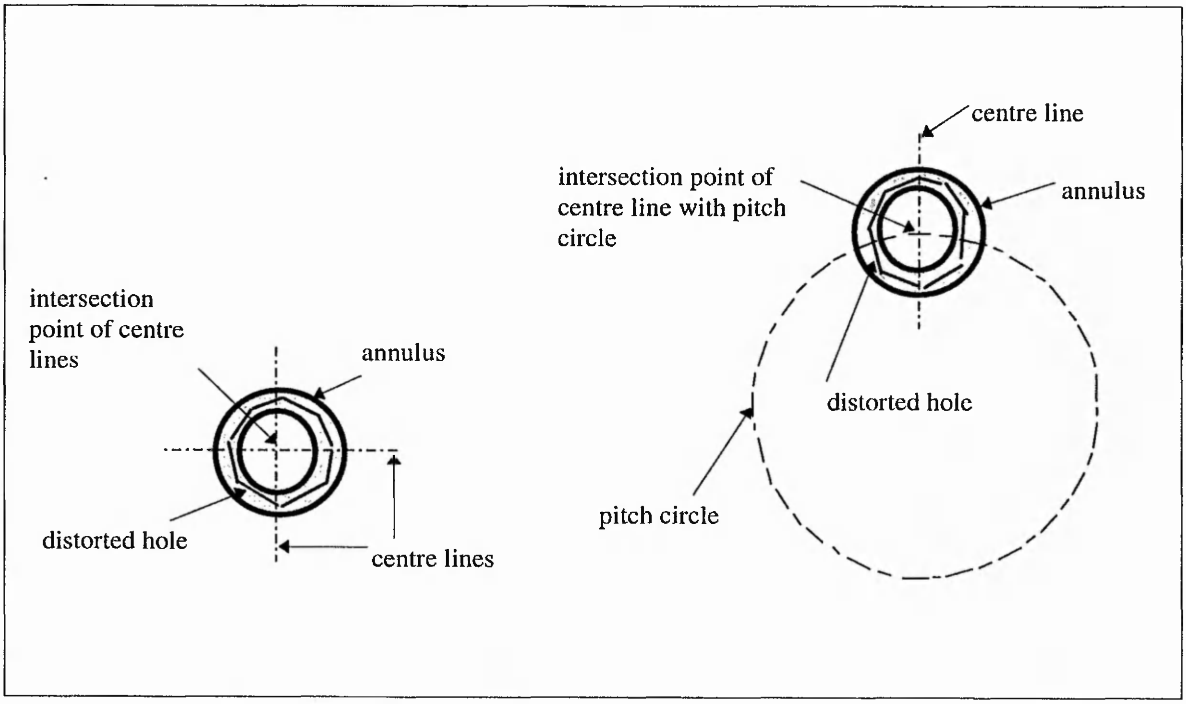


Figure 7.7: Illustrates the detection of distorted circles using intersection points of centre lines and centre line with pitch circle. The annulus has been used to detect line or arc segments which meant to represent the circle.

The geometric information of the distorted hole (circle) maybe represented by combination of lines and arcs. These are then pruned and a complete circle is put in its place. Other holes are found using a similar approach by using the fact that some of these holes may be distorted or simplified.

7.4 Ambiguous interpretations

Interpretation algorithms have been designed to mimic human understanding of engineering drawings. This understanding is passed to the algorithm, for example the expectations of locating a particular *feature* on drawing. The algorithm expects the position of the hole *feature* on drawing is related to conventions (centre line and pitch circle). Crossing of centre lines or centre line with pitch circle may indicate that possible holes exist on them.

Textual information describing a *feature* on a drawing provides additional information and together with the geometry, a complete representation of the *feature* can be derived. However, in some cases, where distortions have occurred, the combination of textual and geometric information may still not provide sufficient description of the *feature*. In such cases an incomplete interpretation of the drawing is generated. For example, the Top view of the drawing in Figure 7.8 has the phrase '4 HOLES DIA 50 INLINE', while the geometry describes one hole. The algorithm then searches the drawing for the evidence of the positions of other holes. It uses the intersection of centre lines and finds 2 more holes, so in this case only three of the four holes indicated by the phrase have been found. These three intersection points will have holes placed on them, but there is no clear evidence of the position of the fourth hole as shown in Figure 7.9. Thus, in some cases the algorithm recovers what it can.

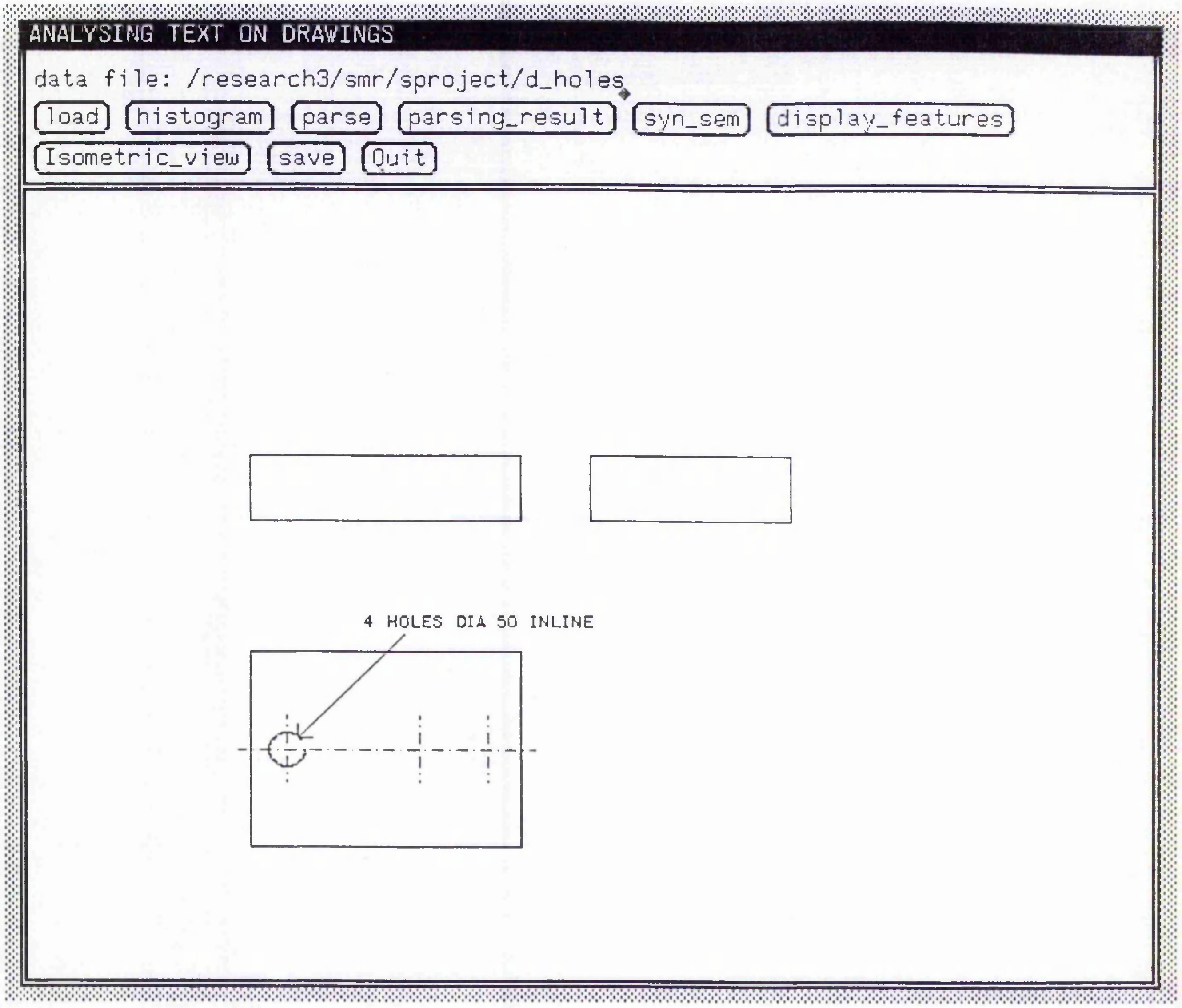


Figure 7.8: A drawing with Top view has a phrase describing a number of repeated features. The evidence shows a fewer number of features than indicated by the phrase.

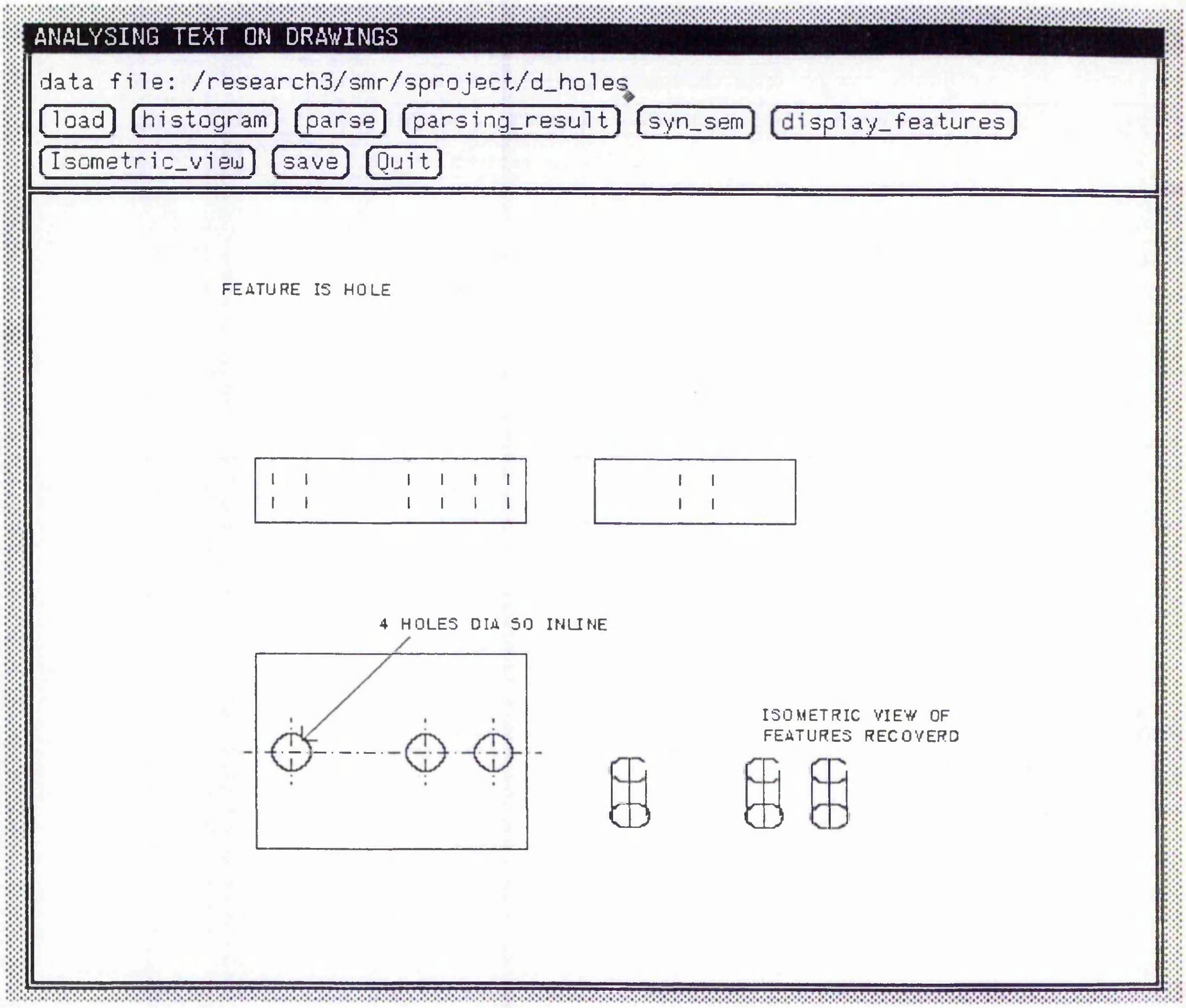


Figure 7.9: Interpretation result of the drawing in Figure 7.8.

Prepositional phrases were recognised to be a combination of simple ones. The developed algorithm was extended to interpret such phrases. This type of phrases require the pre-definition of the relationship between the *features* involved in a prepositional phrase. The example in Figure 7.10, shows a drawing with prepositional phrase indicating the relationship between a number of counterbore holes and a pitch circle. The concept of interpreting this type of phrases is to recognise the simple phrases and their individual *feature*. The algorithm then uses the information obtained to generate meaning of the phrase (explicit description of *features*), Figure 7.11.

A feature recognition algorithm have been proposed and experimented and is described with an example in chapter 6. This algorithm gives an intelligent approach to the text interpretation which is very closely related to a human's interpretation of engineering drawings.

Descriptions of unknown *features* are matched with descriptions of simple *features* stored in a library, Figure 7.12. Applying rules comparing attributes of the unknown *feature* with attributes of *features* in library may produce more than one candidates as shown in Figure 7.13. In this situation, further rule is applied to find a candidate which most closely fits the description of the unknown *feature*, Figure 7.14.

The recognition algorithm works well with simple *features* with small distortions. Also *features* can be recognised despite of their orientations or scaling factors.

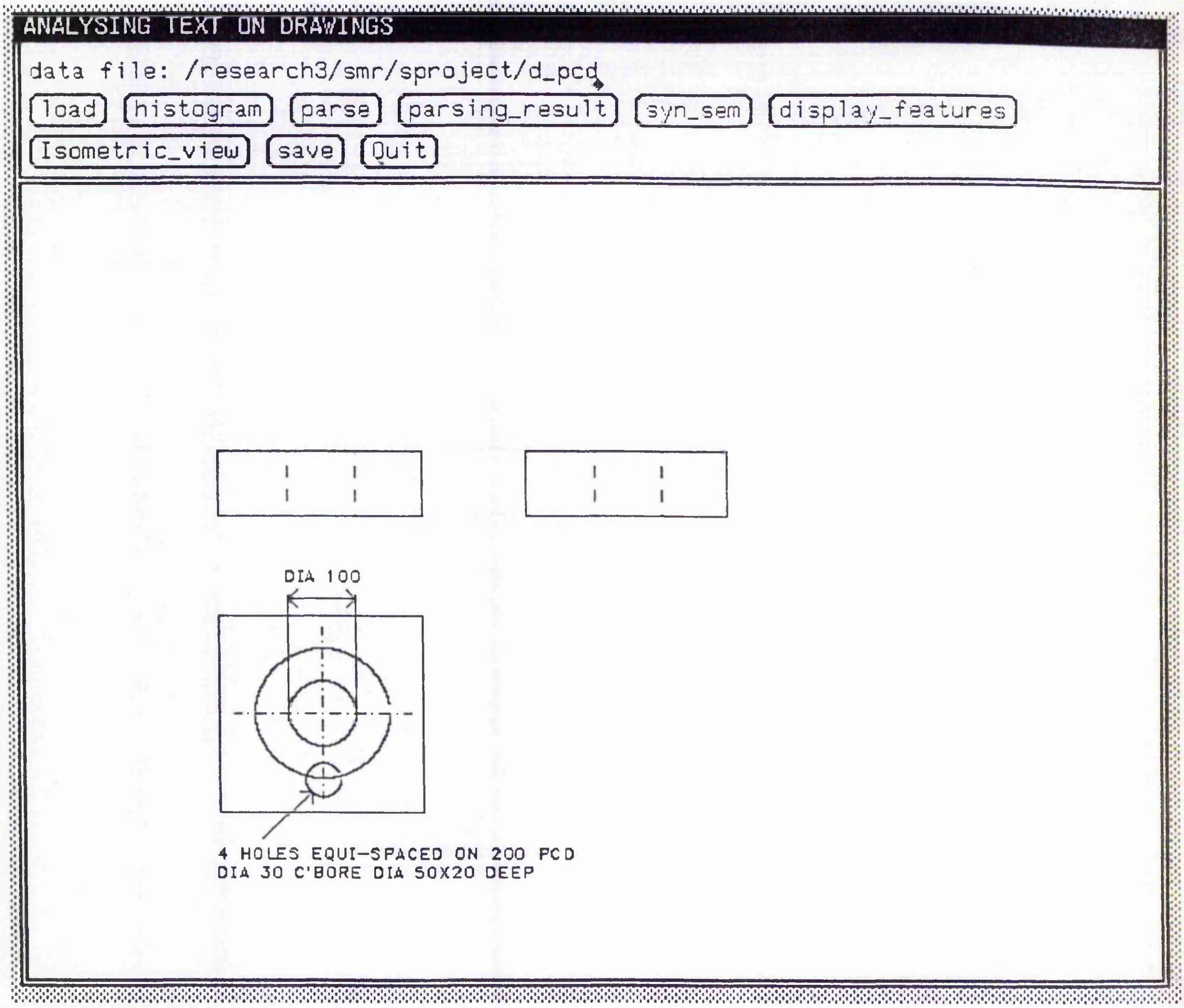


Figure 7.10: Example of prepositional phrase with simplifications.

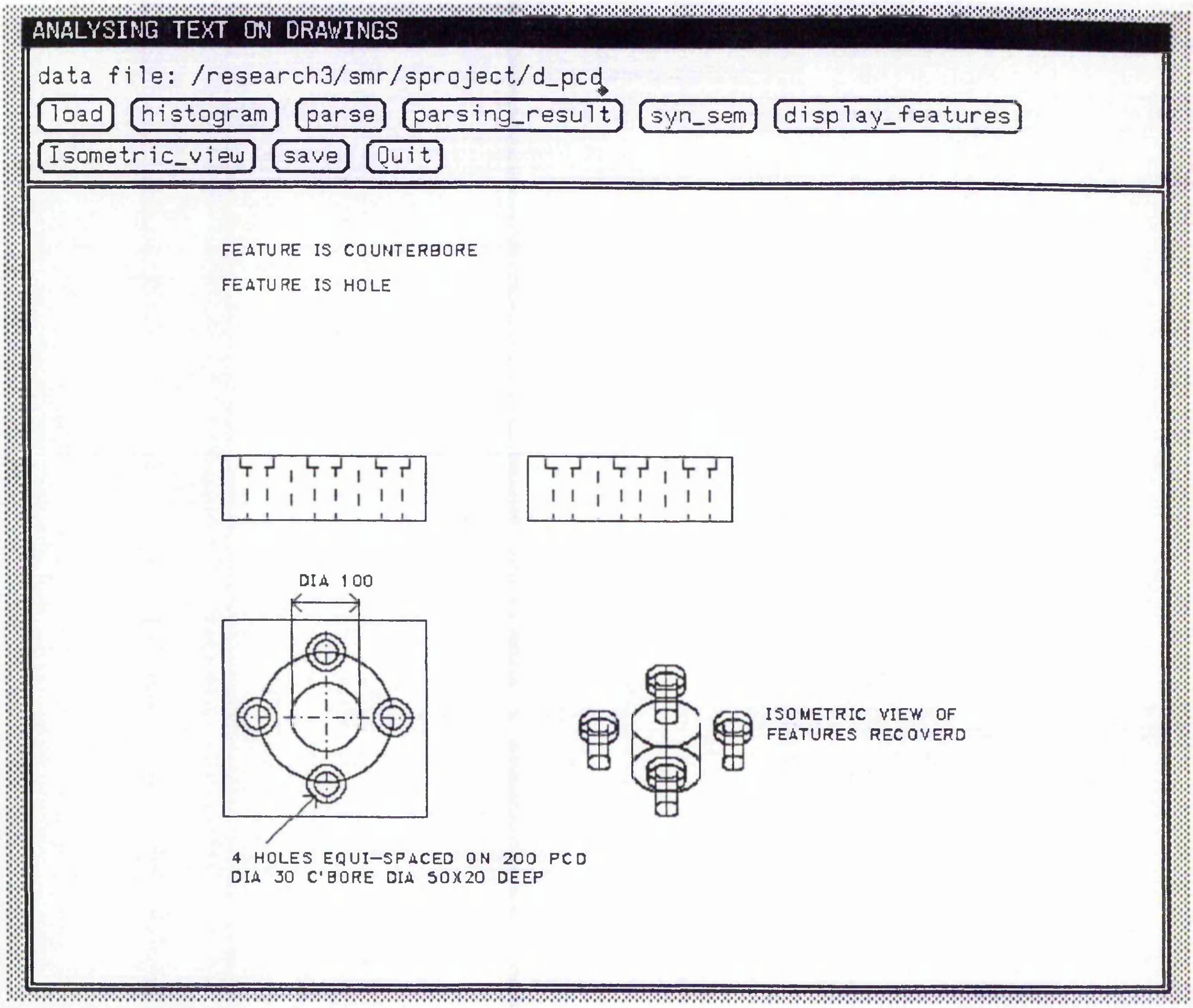


Figure 7.11: Features of prepositional phrase are explicitly described on the drawing and an isometric view showing the features recovered.

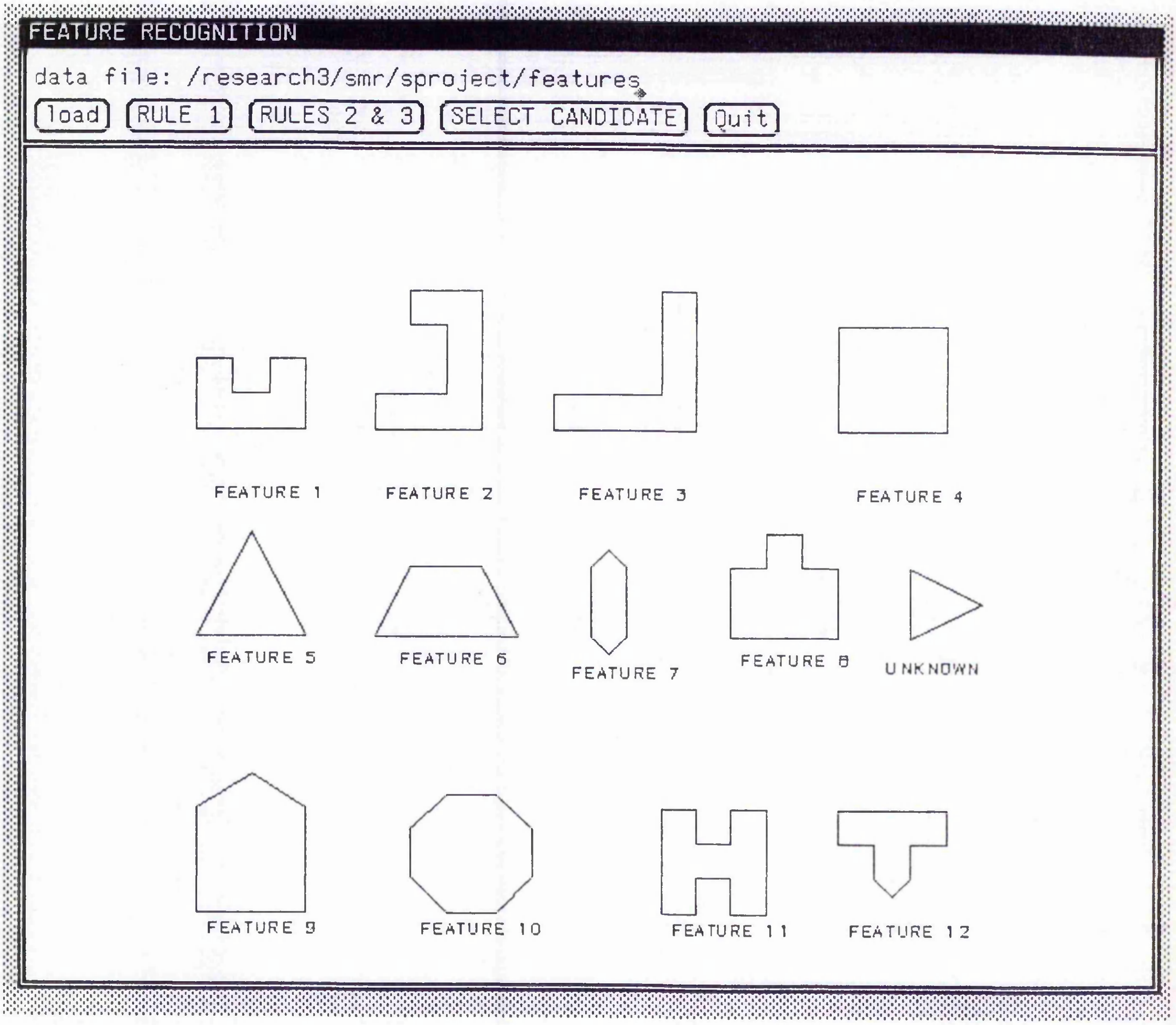


Figure 7.12: Shows an unknown *feature* which is to be matched with *features* stored in library.

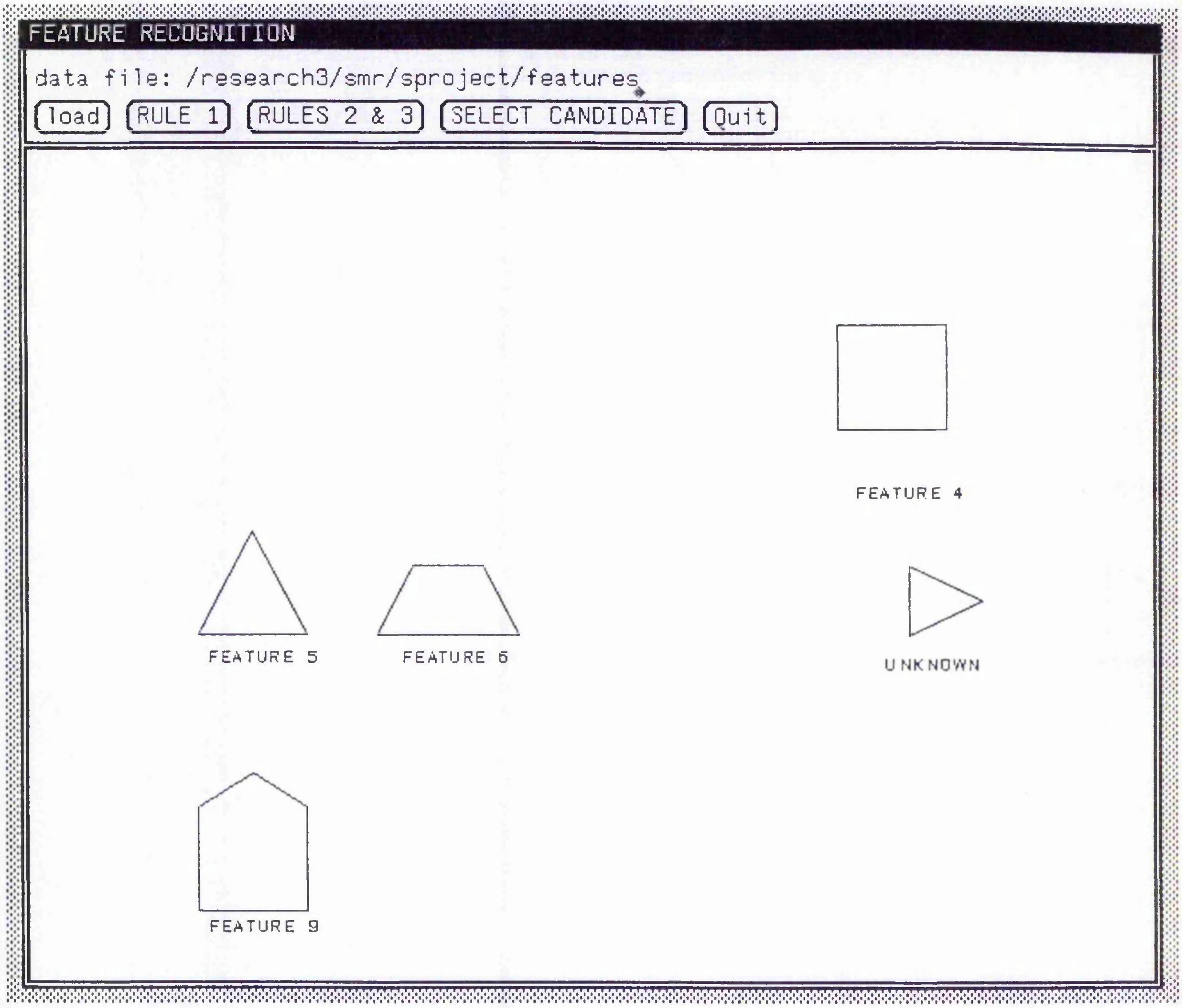


Figure 7.13: Shows a number of candidate *features* matched the unknown *feature* after applying RULE1, RULE2 & RULE3.

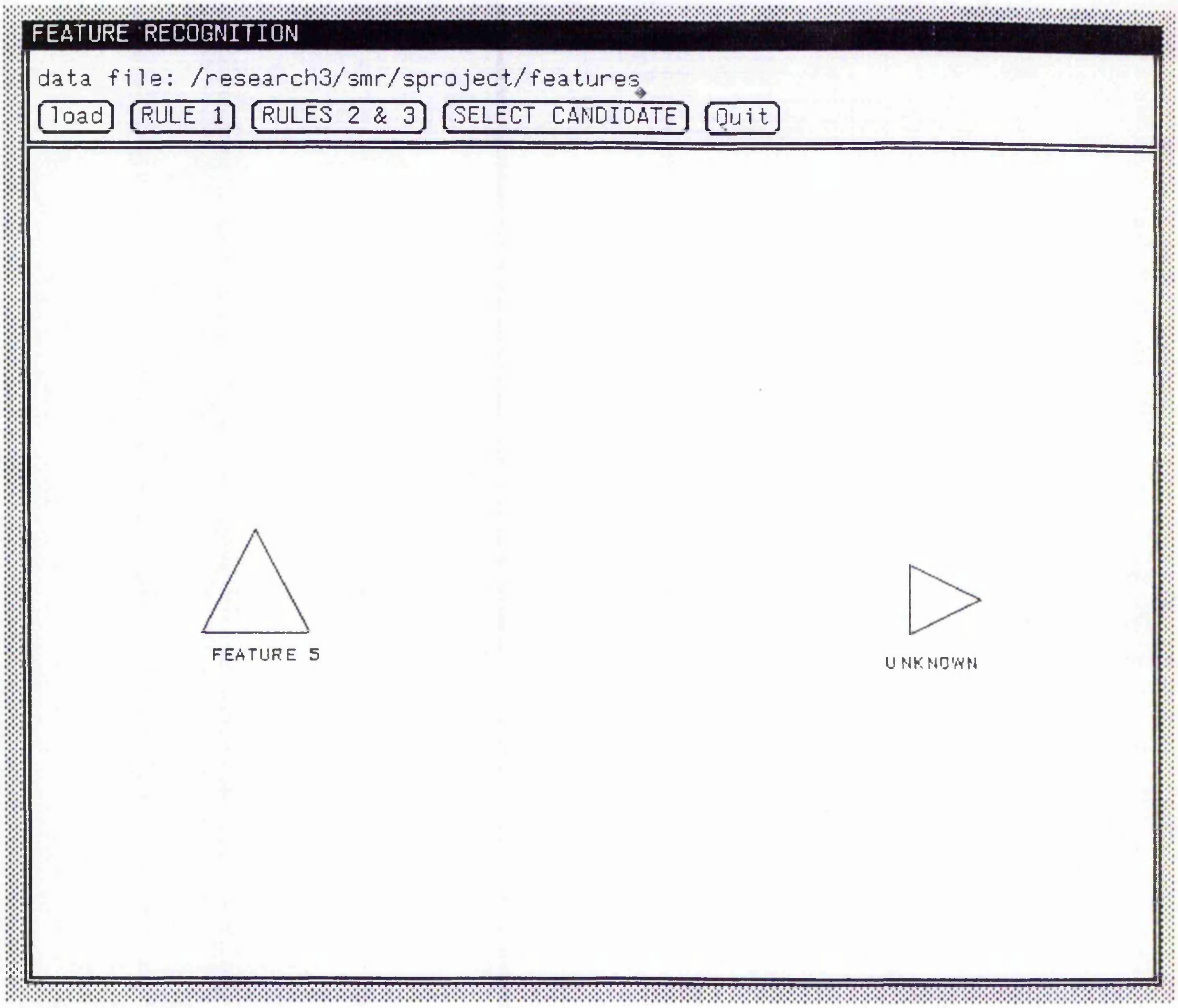


Figure 7.14: Shows one candidate *feature* that has matched the unknown *feature*, a result determined by selecting the minimum value for the difference of their similarity measures.

7.5 Summary

Several examples were given to demonstrate a number of ambiguous cases that can be handled by the algorithms in this thesis. The emphasis has been made on using the information provided in the text. This information together with the geometric information attempt to provide a complete description of the object on the drawing.

As seen in the many cases considered in this chapter, textual information not only provides a description for simplified features, but it can also resolve for errors and distortions arising from other processes.

The interpretation of textual information requires the association of this information with geometric parts on the drawing. This requires an algorithm which is capable of recognising features on the drawing. The algorithm implemented in this work successfully recognises a number of simple features, some were shown in Figure 7.12.

Chapter 8

Discussion and possible Future Work

8.1 Discussion

Automatic interpretation of engineering drawings; are we there yet? This question could be answered by looking at what have been achieved so far by the research in this field for the past two decades. It was not untill the late 80s' that researchers started to attempt automatic interpretation of real engineering drawings. This has been sparked off by the growing advances and development of topics associated with engineering drawing such as scanning, text/graphics separation and in general the advances made in document analysis. Thus, the direction of their research has been diverted from interpreting noise-free, complete (synthetic) drawing lines to simple scanned images of a real drawing. This has introduced a number of problems which kept researchers occupied to seek solutions. The problem of cleaning up the bitmap image, understanding geometric and textual information started to show feasible solutions to the automatic interpretation.

The language, specified by drafting standard, was originally intended solely for use by humans. However, advanced research, analogous to natural language processing, has to aim at a standard method of communication. Structural relations among *features* are the relative location and orientation of one *feature* with respect to others on the drawing. Natural Language Processing can provide interpretation of the textual annotations describing these *features*.

The top-down approach is often used in computer science and applied fields such as parsing of computer languages and image understanding. In order to implement such an approach, input text is first analysed using Natural Language Techniques. A critical point is how larger concepts are broken down into their component parts. Usually, there is so much branching that it is very difficult or impossible to do an exhaustive search of all the appropriate breakdown paths without any supplementary information.

Fortunately, this problem is manageable with our approach, because the number of primitive *features* to be considered, in both 2-D and 3-D, is usually quite small. For example, holes appear in engineering drawings as circles in one view and rectangles in the other two views, all or part of which consist of hidden lines for the sides of the holes. Thus simplifying the task of searching views for matching parts. In other word, the top-down concept is very effective in cases where the breakdown process is simple and straightforward. In this respect, the author has found that this procedure, which exploits the inherent characteristics of mechanical objects used in conjunction with the bottom-up procedure, is more practical than the bottom-up approach alone.

The algorithms that the author presented in this thesis use Syntactic Net defined with respect to the engineering domain. The concept of parsing the phrases has been implemented to encode the phrase and generate a phrasal pattern or syntactic type. This pattern provides the parser with the necessary information to progress to the next stage to generate meaning (Semantic Template).

8.2 Outline of the algorithms

The algorithms described in this thesis are capable of interpreting a number of phrases referring to holes on drawings. This approach is novel and it can recognise phrases describing different type of holes. The interpretation process has been extended to parse prepositional phrases. These phrases contain tokens such as *on, in, to, from* and so on. Thus, the rules are extended to recognise these phrases which represent a geometric relationship between two or more *features*.

To recognise the shape of *features* other than holes on the drawing, the author has implemented a method to give a full description of these *features*. To interpret the input phrases, the parser needs to know which part of the projections correspond to the *feature* described by the phrase, as well as how to pass the stored knowledge about the *feature* to the searching procedure. Therefore, it was necessary to develop procedures to search the drawing for these *features* and match them with the descriptions stored in the knowledge base.

Features and their characteristics are hierarchically organised in the syntactic nets, this arrangement establishes the relationship between tokens within the phrase. The algorithm has been developed for parsing the textual information design to recognise tokens of the phrase. These then matched up with a look up engineering dictionary that associates each token with a category. This approach is most appropriate, since the interpreter is dealing with incomplete sentences (phrases).

In the engineering domain, textual annotations support the geometric information in describing the information on the drawing. Thus, it is necessary for annotations to be simple and meaningful rather than grammatically correct and complicated. The relationships between tokens within a phrase, and the collection of related tokens are driven by rules. These pre-defined rules guide the algorithm from the initial recognition of a token to its eventual geometric contribution in describing the *feature* or the object.

The rules and the facts of some engineering events stored in the knowledge base are capable of interpreting a number of phrases. Additional rules and knowledge are needed for interpreting further phrases. Thus, the limitation of the algorithm, is that only phrases concerning different types of holes are considered in the interpretation. For future work, knowledge base and rules should be extended to accommodate additional *features* or relationships between them.

The principles for the interpretation of annotations on engineering drawings has been set in this work. This can simply be extended to include *features* other than holes, where additional information about these *features* can be described in the knowledge base and defined by the rules. The method of recognising 2-D shapes is described in chapter 6, which also can be extended to recognise more complex shapes.

8.3 Possible Future Work

There are still a number of other problems with the interpretation of engineering drawing that remain unsolved. It was not untill late 80's that researchers started to design more realistic systems which involve the interpretation of real engineering drawings. Initially, they imposed a number of restrictions on their interpretation for example by using a complete drawing data that have no errors. Gradually, new systems started to emerge where a number of the imposed restrictions were relaxed.

Practically, the interpretation process should start with the bitmap image of the drawing including the errors and noise generated by scanning the drawing. This approach can mitigate a number of constraints and restrictions made by previous researchers.

Although it has been recognised previously that textual information is a vital part of the interpretation process [Fletcher88], this has been limited to the textual dimensions [Chai92], [Dori95a], [Dori95b]. The argument made by their work is that the use of dimensions can help generate a precise shape of the object on drawing. However, other textual information (annotations) describing geometric parts have been overlooked.

There are many engineering drawing which have textual annotations on them, and without this information ambiguous 3-D models may be reconstructed. The purpose of annotations on drawing is to complement the geometric information and in many cases such as in repeated *feature*, they even replace part of the geometric information. Drawings such as these can not rely only on the geometric information for the reconstruction of 3-D models. A better system should incorporate algorithms which interpret annotations on drawings and recover simplified or distorted *features*.

The inclusion of textual annotations as part of the interpretation process adds an extra dimension to the interpretation. *Features* or even the object as a whole can be recognised from the annotations, this also allows the interpretation of a wider range of drawings. An expert algorithm can then recognise these primitive *features* and check upon their consistency.

8.3.1 Geometric Information

The geometric information of engineering drawings can only provides partial information when drawing simplifications and textual information are involved. Generally, human intuition need to be applied to understand these kinds of drawings.

It is useful to know if the geometry resembles a complete and unambiguous description of the object on the drawing or additional alternative information is needed to compensate for the complete description. Some drawings may contain additional views, the interpreter needs to identify these additional views which may allow it to use the information on them to produce a complete 2-D representation.

8.3.1.1 Auxiliary Views

This information is sometimes provided in assembly drawing or an auxiliary, partial or sectional views. Because of the variety of conventions and the different characteristics of specialised drawings, the interface between engineering drawings and 3-D solid representation is a large and complicated problem. One of these problem is the sectional view representation also been suggested as future work by [Shaw95]. These views are used to show the internal structure of an object, to clarify the arrangement of assembled parts of the drawing and to emphasise the important portion of an object. Sectional views are frequently used in engineering drawings.

There are occasions when an elevation and a plan view do not give the actual true shape of a surface. The elevation of a pipe junction is shown in Figure 8.1.

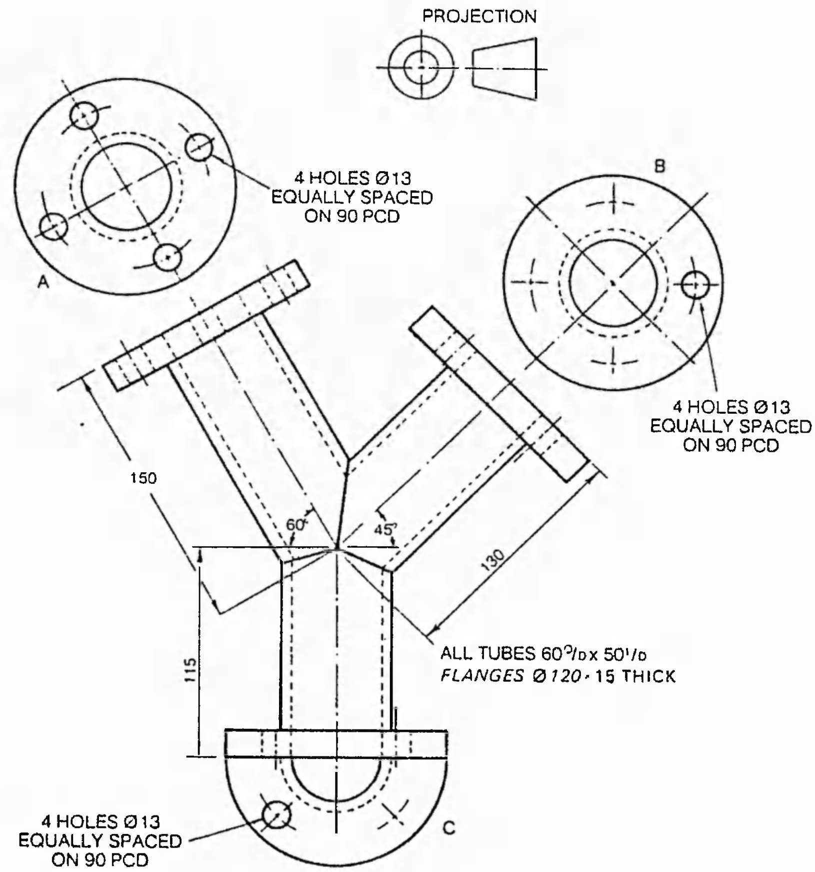


Figure 8.1: Example of Auxiliary view (taken from [Hewitt75]).

To add a plan view would be tedious, for it would involve the construction of several ellipses, and the true shape of the flanges would not be seen. A side elevation seen from either the left or right of the given elevation would be equally time-wasting, for again several ellipses would have to be drawn without the necessary true shape appearing - auxiliary views are required. At A is shown the auxiliary view of the left-hand side flange, while the others are at B and C. These are each views seen when looking at the flanges along a line perpendicular to the face, each in third-angle projection.

The holes in flanges A and B are in the alternative drilling arrangement, but A shows all four holes, which is not essential. At B the centres of four holes are given but only one hole is shown. This is better practice, but real time-saving is shown at C where only half of the true shape is shown.

8.3.1.2 Sectional Views

Quite often an outside view of an object does not adequately describe it, as no internal *features* are shown. In order to show the internal *features* without excessive use of hidden detail lines, the object is imagined to be cut along a cutting plane, Figure 8.2. Sectional views are usually produced to:

1 - Clarify details of the object.

2 - Illustrate internal *features* clearly.

3 - Reduce the number of hidden detail lines.

4 - Facilitate the dimensioning of internal features.

5 - Show the shape of the cross-section.

6 - Show clearly the relative positions of parts forming as assembly.

Cutting planes are represented on drawings by long thin lines thickened at each change of direction and at both ends and labeled by capital letters. A procedure should be developed to understand these views and extract the relevant information required to complete the description of the object on the principal views.

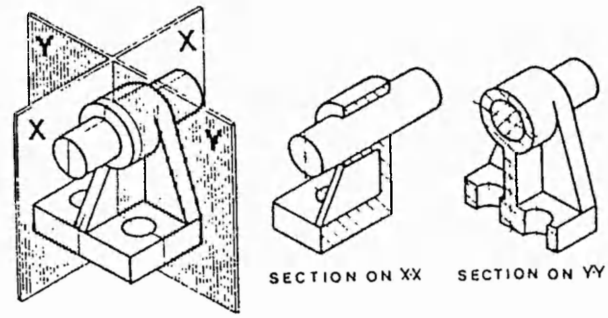


Figure 8.2: Example of Sectional view (taken from [Ostrowsky79]).

8.3.2 Textual Information

Design and manufacturing systems based on solid modeling are gradually replacing the traditional systems. The automatic interpretation of engineering drawings and integration of CAD/CAM systems are the ultimate goal of many designers. Interpretation of engineering drawings may allow less room for error, if all the information provided on them are fully interpreted and used. This task as many others became possible and cheaper to run by computer system. Thus, any computer system has to mimic human interpretation of a specific application to be able to do the task usually done by a human. Here, the specific application is the interpretation of *engineering drawing*, that can be recognised and understood by a computer system if this system can adopt some of human's intuition.

8.3.2.1 Annotations

The capability of interpreting text on engineering drawings gives the algorithm the benefit of using the technical information to understand the drawings. The more information the algorithm can understand, the better chance it has of producing a complete geometric representation of the drawing.

Annotations on engineering drawings can be categorised into a number of different types. Annotations associated with outlines of the drawing, annotations associated with views on the drawing and general annotations.

The work in this thesis mainly concentrated on interpreting annotations associated with outlines, where a collection of these outlines may define a *feature*. But there are many other annotations which need to be interpreted. For example annotations associated with the type of auxiliary view on the drawing. Title block of the drawing may provide information about the company manufacturing the component, this may help in identifying the object or the size of it. Tolerances and measurement may also be obtained from the title block, which can specify the unit of measurement (*mm, cm, inch, etc*).

Information of the material used to manufacture the object may also be indicated in the title block, which may provide hint into the design and manufacture of the object. Any other information in the title block which may be used to understand and help describing the object could be utilised.

A procedure should be designed to identify and categorise the different type of annotations. These different categories then may be associated with a number of parsing processes that generate meaning from them. Information can then be stored in a database and used when needed.

8.3.2.2 Database for Features

Objects and their components (*features*) can be stored in a database. This database, therefore, can provide the interpretation process with the description of any particular *features*. A method already described in chapter 6, for this purpose, but could be extended to include *features* with more complicated descriptions.

Components and even objects as a whole can have their geometric descriptions stored in this database, and upon finding the component or the object form the drawing, a matching description can then be retrieved from the database. Drawing lines of a distorted *feature* can be pruned and replaced by the corresponding description retrieved from the database.

Chapter 9

Conclusions

Interpreting engineering drawing by computer is made more difficult by the fact that most engineering drawings are imperfect and parts of the drawing may have been simplified or distorted during scanning. One of the main objectives of the work described in this thesis was to develop algorithms for interpreting textual information on engineering drawings to recover simplifications or distortions.

When the scanned image of an engineering drawing has been vectorised the interpretation process can begin. At this stage, the drawing data is in no particular logical order, and no explicit information is present regarding their connectivity. It was therefore, necessary to sort the information into three groups, each pertaining to one of the principal projections. An algorithm has been developed which successfully identifies these projections. The drawing is then corrected for rotation and stretching. Then drawing data needs to be identified. A simple algorithm has also been developed which successfully identifies line types on engineering drawing. Line segments which are not connected to others are removed from the list and stored in a residual list. Recognised drawing information were suitably represented in a data file for further processing.

Algorithms have been developed to interpret the textual information which comprised three subproccesses: recognition, parsing and generating representation of meaning. Phrases are broken down into smaller parts (tokens). In the recognition subproccess, these tokens are matched up against a lookup dictionary. The parsing subproccess uses a pattern matching approach. Then by following parsing rules, relationships between tokens are determined and partial semantic template is generated. A parsing rule is represented as a pattern-structure pair, where a pattern is a sequence of tokens and a structure describes the relationship between these tokens (syntactic net).

The third subproccess uses the partial semantic template generated from the parsing subproccess to search the drawing for matching features, and determine the actual location and size of each specific feature.

The interpretation algorithm has been extended to include prepositional phrases. The approach used for this type of phrase is to recognise the prepositional term and then the combined subphrases. Subphrases are processed as before and their generated templates are combined in a relational template.

To make the interpretation process behave in the same way as human would to understand engineering drawings, an algorithm has been implemented to recognise simple features on the drawing. Badly distorted features on drawings can be recognised from the text and their geometric descriptions recovered from a database of features.

The algorithms described in this thesis have made good progress toward the objectives set in section 1.8. The interpretation algorithm has been used to successfully parse a number of phrases related to holes. The general concept of text interpretation has been described here and, therefore, in principle it is not difficult to include other phrases or *features*.

Previous methods involved input by an expert user whereas, the method described in this thesis, has made use of the information existed on the drawing.

The algorithms implemented in this thesis have made a number of advances from previous methods:

1 - They can recognise geometric information and make use of textual annotations already on engineering drawings.

2 - The inclusion of textual annotations as part of the interpretation process allows the solution of a number of cases which are ambiguous when only the outline information is used.

3 - A wider range of engineering drawings can be interpreted.

4 - Expert algorithms are used, because textual annotations are technical information.

5 - The interpretation method relies on the knowledge provided by the knowledge base and does not require assistance from the user.

6 - Finally, the interpretation of textual annotations allows the possibility of converting existing engineering drawings directly into CAD format.

The work described in this thesis involves algorithms for the recognition and understanding of engineering drawings. This work has extended the automatic interpretation of engineering drawing from dealing with noise-free synthetic information to dealing with noisy scanned images. Also the interpretation of textual information on engineering drawing can solve a number of ambiguous cases.

Previous researchers [Alsefeld83a], [Preiss81], [Sakurai82] and [Wesley80] have mainly concentrated on using the geometric information of engineering drawings in order to reconstruct a perfect and a wider range of objects. On the other hand vital information in the text on engineering drawings has been totally overlooked. The algorithms developed in this thesis have embarked on the interpretation of textual annotations in order to correct distortions which may occur due to preproccessing of geometric information. This approach also recovers simplified features which makes the geometric information explicitly describe the object on the drawing. This research has opened the door to a new dimension of interpreting engineering drawings coupled with the extension into an intelligent interpretation.

References

Abdalla94 H. S. Abdalla and J. A. G. Knight, "An Integrated Knowledge-Based and CAD System for Facilitating the Implementation of Concurrent Engineering", *IEE* 1994.

Agam95 Gady Agam, Huizhu Luo and Its'hak Dinstein, "Morphological Approach for Dashed Lines Detection", *Proceedings of the International Workshop on Graphics Recognition, IAPR*, pp. 23-32, August, 1995.

Aldefeld83a B. Aldefeld, "On automatic recognition of 3D structures from 2D representations", *Computer Aided Design*, Vol. 15, No. 2, 1983.

Aldefeld83b B. Aldefeld, "Automatic 3D reconstruction from 2D geometric part descriptions", *IEEE Proceeding on Computer Vision and Pattern Recognition* 1983.

Aldefeld84 B. Aldefeld and H. Richter, "Semiautomatic three dimensional interpretation of line drawings", *Computer and Graphics*, Vol. 8, No. 4, 1984.

Astley78 P. Astley, "Engineering Drawing and Design II", ISBN 0 333 21703 9, 1978.

Balagurusamy90 E. Balagurusamy and J. Howe, "Expert Systems for Management and Engineering". 1990, ISBN 0-13-296690-5.

Banks89 R. N. Banks & D. G. Elliman, "Expectation Based Feedback in a Neural Network which recognises hand-drawn characters and symbols", *First IEE International Conference on Artificial Neural Networks* 1989.

Benjamin88 D. Benjamin, P. Forgues, E. Gulko, J. B. Massicotte, C.Meubus, "The Use of High Knowledge for Enhanced Entry of Engineering Drawings", *IEEE Compt. Soc. Press* Vol. 2, 1988.

Bergengruen89 O. Bergengruen, "About 3D reconstruction from technical drawings", *International Workshop on Industrial Applications of Machine Intelligence and Vision*, Tokyo, April 1989.

Bergevin93 Robert Bergevin and Matrin D. Levine, "Generic Object Recognition: Building and Matching Coarse Descriptions from Line Drawings", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 1, January 1993.

Besl88 Paul J. Besl, "Geometric Modelling and Computer Vision", *Proceedings of the IEEE*, Vol. 76, No. 8, August 1988.

BSI British Standard Institution, Engineering Drawing Practice, B.S.308.

Chai92 I. Chai and D. Dori, "Extraction of Text Boxes from Engineering Drawings", *Machine Vision Applications in Character Recognition and Industrial Inspection, Proceedings Series, 1661, SPIE*, 1992.

Charniak85 Charniak, E. and McDermott, D. (1985) "Introduction to Artificial Intelligence", Addison-Wesley.

Cheng90 Yong-Qing Cheng, Jing-Yu Yang, "A Knowledge-based Graphic Description Tool for Understanding Engineering Drawings", *IEEE Comput. Soc. Press*. Proceeding of the 2nd International IEEE Conference on Tools for Artificial Intelligence, p. 792-6, 1990.

Chhabra94 A. K. Chhabra, "Neural Network Based Text Recognition for Engineering Drawing Conversion", *IEEE International Conference on Neural Networks*, Vol. 1-7, 1994.

Chiou92 Rong-Nan Chiou, King-Chu Hung, Jinn-Kwei Guo, Chin-Hsing Chen, Tzu-I Fan and Jau Yien Lee, " Polyhedron Recognition Using Three-View Analysis", *Pattern Recognition*, Vol. 25, No. 1, pp. 1-16, 1992.

Chou88 Hon-Yue Chou and Michael J. Wozny, "A flexible Interface between Two-Dimensional Representation and three-Dimensional

Representation", *IEEE International Conference on Computer Integrated Manufacturing*, 1988.

Chung93 Minhwa Chung and Dan Moldovan, "Parallel Memory-Based Parsing on SNAP", *IEEE* 1993.

Chung94a Minhwa Chung and Dan Moldovan, "Memory-Based Parsing with Parallel Marker-Passing", *IEEE* 1994.

Chung94b Minhwa Chung and Dan Moldovan, "Applying Parallel Processing to Natural-Language Processing", *IEEE Expert*, Vol. 9, pp. 36-44, Feb. 1994.

Collier82 P. Collier & R. Wilson, "Engineering Drawing", Second Level, ISBN 0 09 146611 3, 1982.

Cooley88 A. Kargas, P. Cooley and T. H. E. Richards, "Interpretation of engineering drawings as solid models", *Computer-Aided Engineering Journal*, April 1988.

Crocker91 Gary A. Crocker and William F. Reinke, "An Editable Nonmanifold Boundary Representation", *IEEE Computer Graphics and Applications*. Vol. 11, pp. 39-51, March 1991.

Dori90 Dov Dori, "Pre-processing of Engineering Drawings for 3D Reconstruction", *IEEE Comput. Soc. Press* 1990.

Dori95a Dov Dori and Yelena Velkovitch, "Segmentation and Recognition of dimensioning Text From Engineering Drawings", *Proceedings of the International Workshop on Graphics Recognition, IAPR*, pp. 141-150, August, 1995.

Dori95b D. Dori, "Expressing Structural Relations Among Dimension-Set Components Using the Object-Process Methodology", *Proceedings of the International Workshop on Graphics Recognition, IAPR*, pp. 131-140, August, 1995.

Dori95c Dov Dori and Karl Tombre, "From engineering drawings to 3D CAD models: are we ready now?, *Computer-Aided Design*, Vol. 27, No. 4, April 1995.

Elliman90 D. G. Elliman, "A Review of Segmentation and Contextual Analysis Techniques for Text Recognition", *Pattern Recognition*, Vol. 23, No. 3/4, PP. 337-346, 1990.

Engelbercht88 Jan R. Engelbrecht and Friedrich M. Wahl, "Polyhedral Object Recognition Using Hough-Space Features", *Pattern Recognition*, Vol. 21, pp. 155-167, 1988.

Er83 M. C. Er, " Interpreting Engineering Drawings", *Australian Computer Science Communications*, Vol. 5, 1983.

Feldman93 Jerome A. Feldman, "Structure Connectionist Models and language learning", *Artificial Intelligence Review* 7, 301- 312, 1993.

Feng95 Chang-Xue Feng and Andrew Kusiak, "Constraint-based design of parts", *Computer-Aided Design*, Vol. 27, No 5, May 1995.

Filipski92 A. J. Filipski and R. Flandrena, "Automated conversion of engineering drawing to CAD form", *Proceedings of the IEEE*, Vol. 80, No. 7, pp. 1195-1209, 1992.

Flasinski95 Mariusz Flasinski, "Use of graph grammers for the description of mechanical parts", *Computer-Aided Design*, Vol. 27, No. 6, June 1995.

Fletcher88 L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from text/graphics images", *IEEE Trans. Pattern Anal. Machine Inter.*, Vol. 10, pp. 910-918, 1988.

Floriana87 L. De Floriana, "A graph based approach to object feature recognition", in *proc. 3rd Annu. Symp. Computational Geometry*, Waterloo, Ont., Canada, 1987.

Ginsparg83 Jerrold M. Ginsparg, "A Robust Portable Natural Language Data Base Interface", *Proc Conf Applied Natural Language Processing*, Feb. 1983.

Gopin79 A. Gopin and O. Gossard, "Symbolic dimensioning", *Proc. First Ann. Conf. Comput. Graphics CAD/CAM Syst.* Massachussetts Institute of Technology, Cambridge, MA USA (April 1979).

Graham88 Ian Graham and Peter Llewelyn Jones, "Expert Systems Knowledge, Uncertainty and Decision", 1988, ISBN 0 412 28510 X.

Haralick82 R. M. Haralick and D. Queeney, "Understanding engineering drawings", *Computer Graphics and Image Processing*, Vol. 20, pp. 244-258, 1982.

Hart75 K. R. Hart, "Engineering Drawing with problems and solutions", ISBN 0 340 17692 X, 1975.

Hewitt75 T. H. Hewitt, "Advanced Geometrical and Engineering Drawing", ISBN 0 340 20352 8, 1975.

Hillyard78 R. C. Hillyard and I. C. Braid, "Analysis of dimensions and tolerances in computer-aided mechanical design", *comput. Aided Des.* Vol. 10, No. 3 (may 1978) pp. 161-166.

Idesawa73 M. Idesawa, "A System to generate a solid figure from three views", *Bulletin of the JSM*, Vol. 16, No. 92, Feb. 1973.

Ji95 Qiang Ji and Michael M. Marefat, "Bayesian Approach for Extracting and Identifying Features", *Computer-Aided Design*, Vol. 27, No. 6, June 1995.

Joseph92 S. H. Joseph and T. P. Pridmore, "Knowledge-Directed Interpretation of Mechanical Engineering Drawings", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 9, September 1992.

Karima85 Medhat Karima, Kuldip S. Sadhal and Tim O. McNeil, "From Paper Drawings to Computer-Aided Design", *IEEE Comput. Graphics Applications*, Feb. 1985, pp. 27-39.

Kasturi90 R. Kasturi and S. T. Bow, et al., "A system for interpretation of line drawing", *IEEE Trans. PAM*, Vol. 12, No. 10, pp. 978-991, 1990.

Kuo96 M. H. Kuo, "RECONSTRUCTION OF QUADRIC SURFACE SOLIDS FROM THREE ORTHOGRAPHIC VIEWS", Ph.D. thesis, Nottingham University, UK, 1996.

Kyprianou80 L. K. Kyprianou, "Shape classification in computer-aided design", Ph.D. dissertation, Cambridge Univ., Cambridge, July 1980.

Lai94 Chan Pyng Lai and Rangachar Kasturi, "Detection of Dimension Sets in Engineering Drawings", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 8, August 1994, pp. 848-55.

Laine91 Andrew Laine, William Ball and Arun Kumar, "A Multiscale Approach for Recognizing Complex Annotations in Engineering Documents", *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (91CH2983-5).

Lee87 Y. C. Lee and K. S. Fu, "Machine understanding of CSG: Extraction and unification of manufacturing features", *IEEE Compt. Graphics Appl.*,pp. 20-32, 1987.

Lequette88 R. Lequette, "Automatic Construction of Curvilinear solids from Wireframe Views", *Computer Aided Design*, Vol. 20, No. 4, 1988.

Lysak90 D. B. Lysak and R. Kasturi, "Interpretation of line drawings with multiple views", In *Proc. of Int. Conf. on Pattern Recognition*, Atlantic City, 1990.

Lysak91 D. B. Lysak and R. Kasturi, "Interpretation of engineering drawings of polyhedral and nonpolyhedral objects", In Proc. of Int. *Conf. on Document Analysis and Recognition*, pp. 79-87, Saint-Malo, France, 1991.

Lysak95 D. B. Lysak, P. M. Devaux and R. Kasturi, "View Labeling for Automated Interpretation of Engineering Drawings", *Pattern Recognition,* Vol. 28, No. 3, pp. 393-407, 1995.

Manesh90 Ali Manesh, Joseph Wrobel and Jianhua Gao, "Automatic Vectorization of Scanned Engineering Drawings", *IEEE Comput. Soc. Press* 1990.

Marefat90 M. Marefat and R. L. Kashyap, "Geometric Reasoning for Recognition of three-Dimensional Object Features", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 10, October 1990.

Marefat93 M. Marefat, Sandeep Malhotra and R. L. Kashyap, "Object-Oriented Intelligent Computer-Integrated Design, Process Planning, and Inspection", *IEEE Computer*, March 1993.

Marks90 Joe Marks, "A Syntax and Semantics for Network Diagrams", *IEEE Comput. Soc. Proceedings of the IEEE workshop on Visual Languages*, 1990.

Mayer89 Glenn Mayer, Chiye Yamamoto, Martha Evens and Joel A. Michael, "Constructing a Knowledge Base from a Natural Language Text", *IEEE Compt. Soc. Press, Proceedings Second Annual Symposium on Computer-Based Medical Systems*, 1989.

Meeran93 S. Meeran and M. J. Pratt, "Automated feature Recognition from 2D drawings", *Computer-Aided Design*, Vol. 25, No. 1, January 1993.

Meier86 Andreas Meier, "Applying Relational Database Techniques to Solid Modlling", *Computer Aided Design*, Vol. 18, No. 6, 1986.

Nagasamy90 V. Nagasamy and N. A. Langrana, "Engineering Drawing Processing and Vectorisation System", *Computer Vision, Graphics and Image Processing*, Vol. 49, No. 3, Mar. 1990, pp. 379-397.

Nagasamy91 V. Nagasamy and N. A. Langrana, "Reconstruction of three-dimensional objects using a knowledge-based environment", *Engineering with Computers*, Vol. 7, 1991.

Nishihara94 Seiichi Nishihara, "Understanding Engineering Drawings - An Intelligent Approach to Solid Modeling", *IEEE Comput. Doc. Press* 1994.

Ostrowsky79 O. Ostrowsky, "Engineering drawing for Technicians", Vol. 1, 1979, ISBN 0 7131 3408 9.

Parra-Loera92 Ramon Parra-Loera, Wiley E. Thompson and Gerald M. Flachs, "A Database Management System for Object Recognition Using

Normalized Interval Vertex Descriptors", *Proc. SPIE*, Vol. 1823, pp. 72- 84, 1992.

Parra87 R. Parra, W. E. Thompson and A. L. Gilbert, "Contour representation: Normalized interval / Vertex Descriptiors", *Proceddings of the 18th Annual pittsburgh Conference on Modeling and Simulation*, Vol. 18, Part 4, PP. 1407-1414, April 1987.

Pavlidis82 T. Pavlidis, "Vector and Arc Encoding of Graphics And Text", Proc. Sixth Int'l Conf. Pattern Recognition, Munich, Oct. 1982, PP. 610-613.

Pham93 Trung Tat Pham & Guanrong Chen, "A fuzzy logic approach to object recognition", *SPIE*, Vol. 2061, 1993.

Pickup79 F. Pickup and M. A. Parker, "Engineering Drawing with Worked Examples", Third Edition, 1979, ISBN 0 09 126451 0.

Poliakoff95 J. F. Poliakoff, P. D. Thomas, S. M. Razzaq and N. G. Shaw, "3-D Reconstruction for Correction of Errors and Imperfections in Scanned Engineering Drawings", *Combined Proceedings of (EDUGRAPHICS '95) and (COMPUGRAPHICS '95),* pp. 98-107, December 1995. Alvor, Algarve, Portugal.

Preiss81 K. Preiss, "Algorithms for Automatic Conversion of a 3-view drawing of a plane-faced part to the 3-D representation", *Computer in Industry*, Vol. 2, 1981.

Preiss84 K. Preiss, "Constructing the solid representation from engineering projections", *Computer and Graphics*, Vol. 8, No. 4, 1984.

Requicha77 A. A. G. Requicha, "Dimensioning and tolerancing", PADL TM-19 Production Automation Project University of Rochester, NV, USA (may 1977).

Sakurai83 H. Sakurai and D. C. Gossard, "Solid model input through orthographic views", *Computer Graphics*, Vol. 17, No. 3, 1983.

Shah88 Jami J. Shah and Mary T. Rogers, "Functional requirements and conceptual design of the Feature-Based Modeling System", *Computer-Aided Engineering Journal*, February 1988.

Shaw95 N. G. Shaw, "3D Reconstruction and Correction of objects described by Engineering Drawings", PhD The Nottingham Trent University 1995.

Shi91 Zesheng Shi & Jianhua Ding,"A New Input Method For CAD ------ Automatical Read-in Technique Of Paper Drawings", *1991*

International Conference on Circuits and Systems, June 1991, Shenzhen, China.

Simmons80 Simmons & Maguire, "A Manual of Engineering Drawing Practice", 1980, ISBN 0 340 17996 1.

Srinivasan90 Partha Srinivasan, Ping Liang and Susan Hackwood, "Computational Geometric Methods in Volumetric Intersection for 3D Reconstruction", *Pattern Recognition,* Vol. 23, No. 8, pp. 843-857, 1990.

Staley83 S. M. Staley, "Using syntactic pattern recognition to extract feature information from a solid geometric data base", *Comput. Mech. Eng.*, Vol. 2, pp. 61-66, Sept. 1983.

Sutherland77 I. E. Sutherland, “Sketchpad: a man-machine graphical communication system”, MIT Lincoln Laboratory Technical Report No 296 (January 1977).

Thomas95 P. D. Thomas, J. F. Poliakoff, S. M. Razzaq and R. J. Whitrow, "A Combined High and Low Level Approach to Interpreting Scanned Engineering Drawings", *Proceedings of the International Workshop on Graphics Recognition, IAPR*, pp. 179-188, August, 1995.

Vaxiviere92 Pascal Vaxiviere and Karl Tombre, "Celesstin: CAD Conversion of Mechanical Drawings", *Computer*, Vol. 25, No. 7, July 1992, pp. 46-54.

Wahl82 F. M. Wahl, M. K. Y. Wong, and R. G. Casey, "Block Segmentation and Text extraction in mixed text/image documents", *Comput. Vision, Graphics, Image Processing*, Vol. 20, pp. 375-390, 1982.

Waite89 M. Waite, "Data Structures for the Reconstruction of Engineering Drawings", PhD Nottingham Polytechnic 1989.

Weiler85 K. Weiler, "Edge-Based Data Structures For Solid Modeling in Curved-Surface Environments", *IEEE CG&A*, Jan. 1985, pp. 21-40.

Wesley80 M. A. Wesley and G. Markowsky, "Fleshing out Wireframes", *IBM Journal of Research and Development*, Vol. 24, No. 5, 1980.

Wesley81 M. A. Wesley and G. Markowsky, "Fleshing out Projections", *IBM Journal of Research and Development*, Vol. 25, No. 6, 1981.

Willis89 D. Willis, I. A. Donaldson, A. D. Ramage, J. L. Murray and M. Williams, "A Knowledge-Based System for Process Planning Based on a Solid Modeller", *Computer-Aided Engineering Journal*, February 1989.

Woo82 T. C. Woo, "Feature extraction by volume decomposition ", in *proc. conf. CAD/CAM Technol. Mech. Eng.*, M. I. T., Cambridge, Mar. 1982, pp. 76-94.

Yoshiura84 Hiroshi Yoshiura, “Top-Down Construction of 3-D Mechanical Object shape from Engineering Drawings”, *IEEE Computer*, Dec. 1984.

Appendices

Appendix A - Examples

Knowledge needed for Interpretation

As the author developed the algorithm to a number of different situations of features on engineering drawing, the main question is how much knowledge should the algorithm requires before it can produce a valid interpretation. If we wish for an algorithm to interprets a specific domain, we obviously need less knowledge than if we intended for a general domain. On the other hand if we aim for an expert algorithm, then the more knowledge and skill the algorithm possesses the greater the chances of success in understanding the domain. Interpretation algorithms should make use of thorough understanding of the following basic areas of draughting.

(1) Geometric Construction

(2) Orthographic Projection

(3) Annotations

(4) Dimensioning

(5) Sectional Views

(6) Auxiliary Views

All of the areas mentioned above contribute in describing the object on the drawing. Thus, the interpretation process required to be familiar with these areas in order to fully understand engineering drawings.

The knowledge incorporated within the algorithm should includes all the possible representations and dimensioning of a *feature*. This knowledge about the domain makes the interpretation process behaves the same way as human would in interpreting engineering drawings. Many situations related to *features* are made explicit in the form of knowledge added to the interpretation process. This knowledge depicts the possible shape, the type of dimension and location of *features* described by the phrase on engineering drawings.

The following Figures in this section show a number of different representations which include notes and dimensions, some of the situations about holes that the algorithm is required to be familiar with.

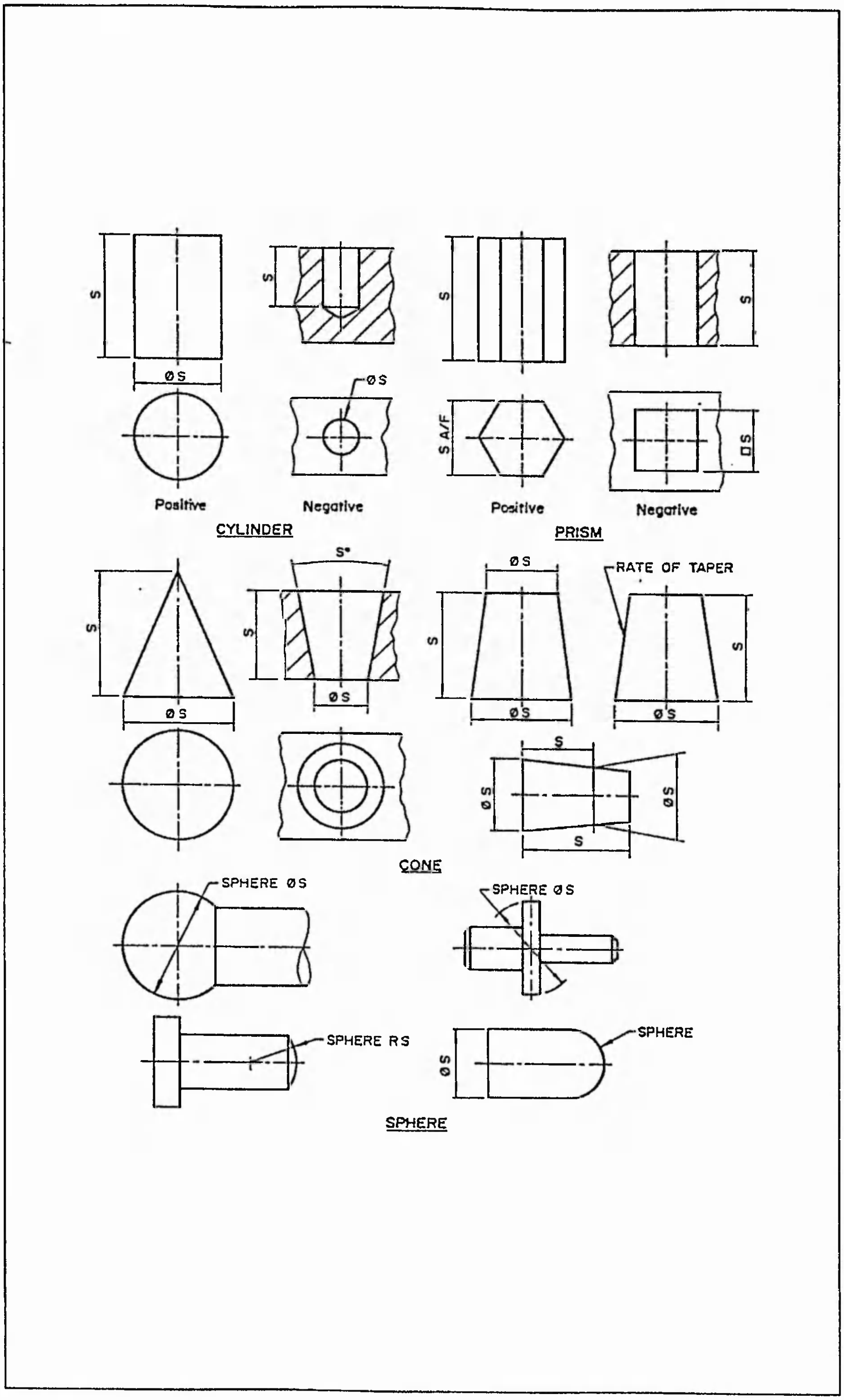


Figure A.1: Example of size dimensions for common geometrical shapes (taken from [Pickup79]).

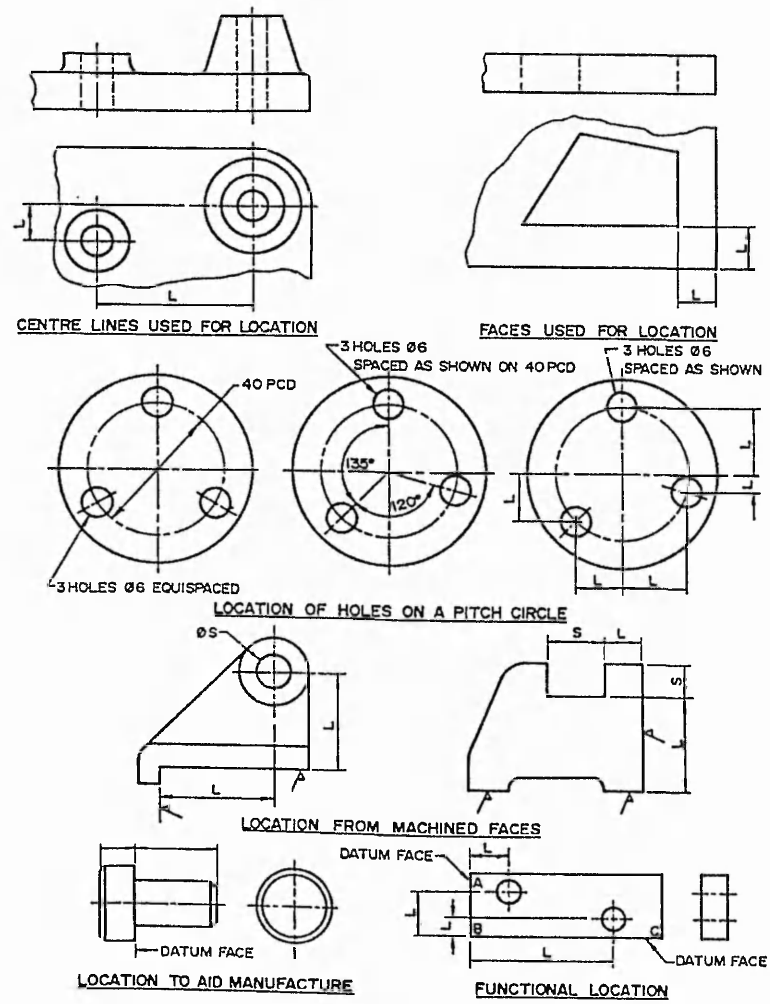


Figure A.2: Example of location dimensions (taken from [Pickup79]).

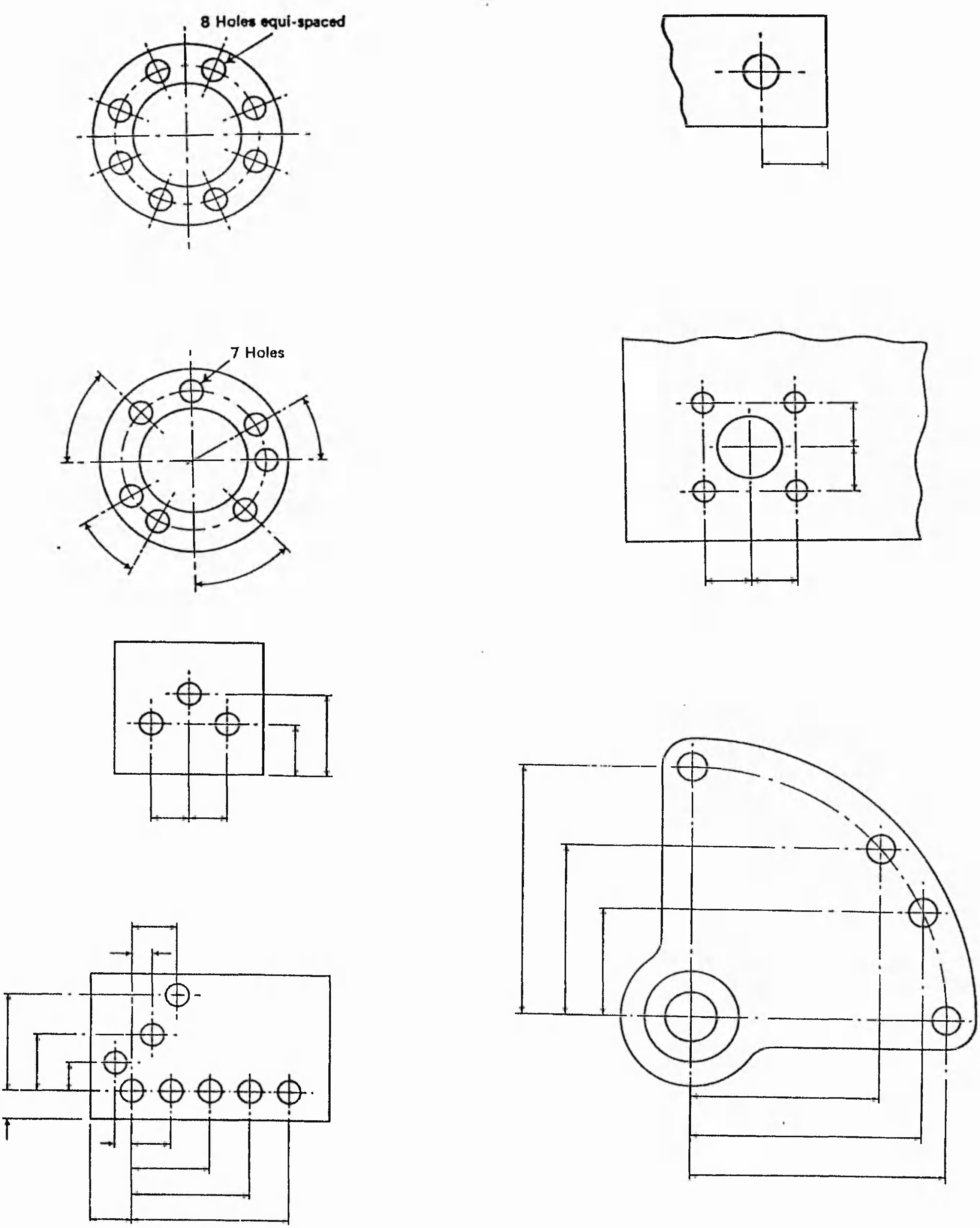


Figure A.3: Example of using dimensions and conventions to locate holes on Engineering Drawings (taken from [BSI]).

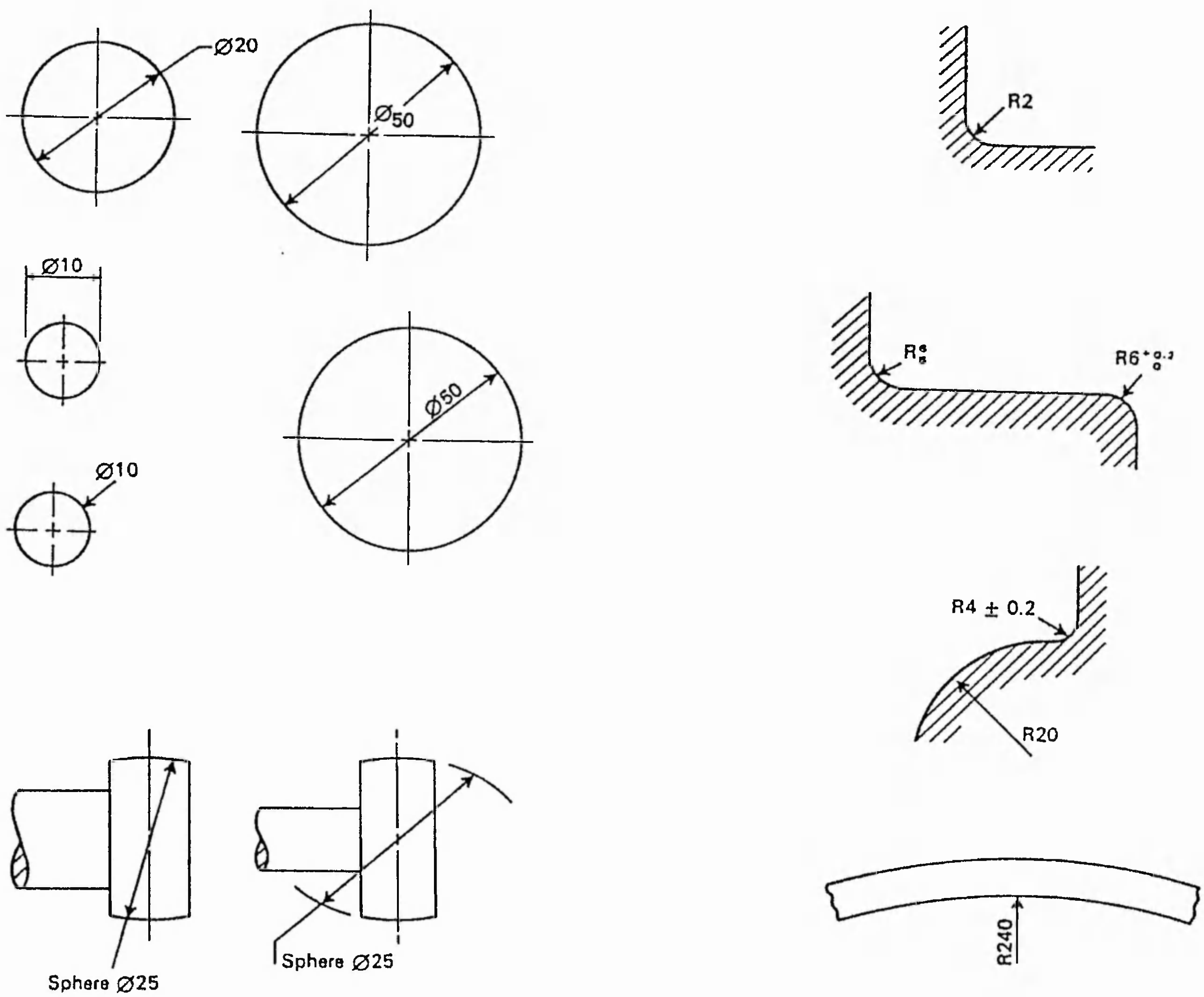


Figure A.4: Example of dimensioning circular features (taken from [BSI]).

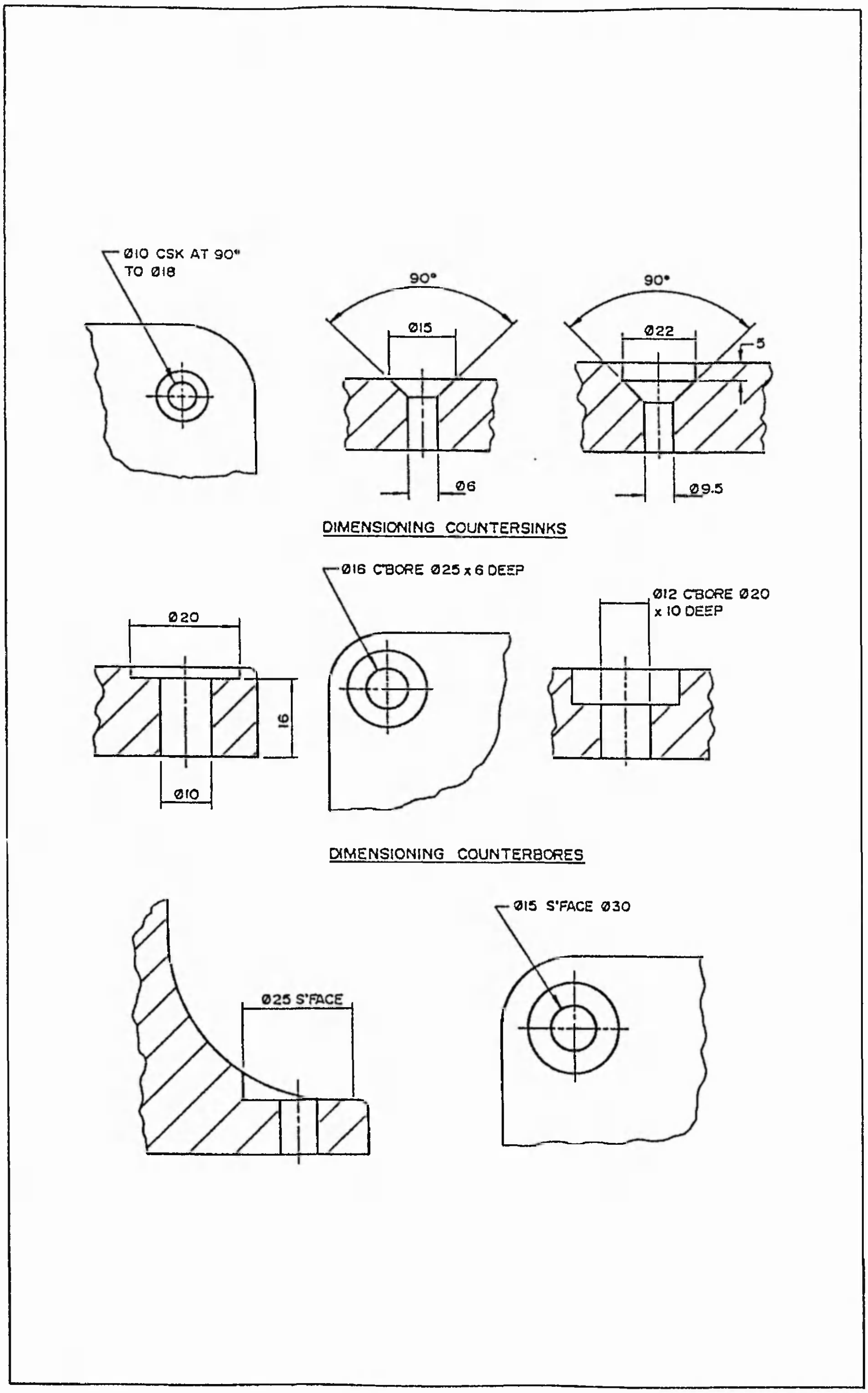


Figure A5: Example of dimensioning Spotface (taken from [Pickup79]).

Appendix B - Published Papers

1 - "A Combined High and Low Level Approach to Interpreting Scanned Engineering Drawings", *Proceedings of the International Workshop on Graphics Recognition, IAPR, pp. 179-188, August, 1995.* A modified version is published in "*Lecture Notes in Computer Science*", pp. 233-245, ISBN 3-540-61226-2.

2 - "3-D Reconstruction for Correction of Errors and Imperfections in Scanned Engineering Drawings", *Combined Proceedings of (EDUGRAPHICS '95) and (COMPUGRAPHICS '95), pp. 98-107, December 1995. Alvor, Algarve, Portugal.*

A Combined High and Low Level Approach to Interpreting Scanned Engineering Drawings

Peter D. Thomas, Janet F. Poliakoff, Sabah M. Razzaq and Robert J. Whitrow

Department of Computing, The Nottingham Trent University,
Burton Street, Nottingham NG1 4BU, England.
E-mail: pdt@doc.ntu.ac.uk
Fax: +44 (0)115 948 6518

Abstract

The effective computer interpretation of engineering drawing remains a desirable aim yet it continues to provide academic challenge. Much early work was concerned with the interpretation of low level vectorised data. For simple drawings, direct association and interpretation of the low level data often provides a very effective technique but drawing data, whether linework or higher level textual information, can be subject to inaccuracies and uncertainties of interpretation. Thus drawing errors and problems introduced by scanning are likely to introduce ambiguities which cannot be resolved directly from the low level data. The approach described in this paper combines features of a low level approach based on node and vertex association with a higher level interpretation of the textual content of the drawing. The textual description of dimensions, etc. has previously been used by the authors and by others, for the correction of drawing structures, in some cases using 3-D reconstruction as a means of validating the data association. The present work attempts to model an aspect of human drawing interpretation, whereby an 'envelope of expectation' is developed, through the interpretation of dimensioning and annotation information. This approach allows a link to be established between the highest level information on the drawing (such as the title block) and the low level vectors of the three elevations. It is thus no longer necessary to interpret obscure detail within the vector data directly. Separation of text on the drawing using OCR techniques allows the field of interpretation for the linework to be significantly narrowed.

1 Introduction

An engineering drawing is a description of an object, which consists not only of geometric information but also of annotations and details of the manufacturing process. The geometric information is commonly contained in three orthographic projections, as shown in Figure 1. Each projection will normally include dimensioning and may contain additional textual information. The production of engineering drawings has changed greatly in recent years because of the use of Computer Aided Design (CAD) packages. The drawings produced by such packages are then be stored in computer data files for future updating and alteration. Furthermore, these computerised drawings may be used by the computer to generate 3-D reconstruction of the object, a task which was relatively difficult with the older paper drawings [1] - [4]. With increasing automation and integration of CAD with Computer Aided Manufacture (CAM) the use of such data files is growing in importance. Unfortunately many engineering drawing designs are available only on paper and cannot be used by a CAD system until they have been converted into a suitable format. Several groups [2] - [6] including our own are working on the automatic conversion of these paper drawings into computer readable form. The conversion process begins with the scanning and vectorisation of the drawings, then automatic interpretation takes place. Checking for consistency between the projections can be done and many errors can be corrected using redundancy and the dimensioning information in the projections. 3-D reconstruction allows the low level geometrical information in all three projections to be correlated and much work has been done in this area [5], [6].

Although low level processing is able to model some aspects of human drawing interpretation, another aspect of human interpretation has been largely overlooked. The engineer uses the textual information on the drawing to develop an 'envelope of expectation' and then goes on to identify the expected features. This accelerates the process of understanding the drawing. Even more importantly, the use of more complex information (dimensioning and annotation etc.) is vital to the successful interpretation, particularly in those cases where features are either omitted or distorted. For example, in the drawing in Figure 1, the phrase "6 HOLES DIA 5 EQUI-SPACED" leads an engineer to expect to find six holes each with a diameter of 5 and at an equal distance from each other, whereas only one is actually drawn. Thus, the engineer uses the textual information to locate the positions of the other five holes. The best which could be done by a "computerised interpreter" without this information would be to produce an object with only a single hole, as shown in Figure 2. In other cases, errors caused by scanning of the original drawing may lead to failure of the computer to interpret a drawing correctly. In Figure 1, the phrase "DIA. 10" allows the engineer to decide that the distorted shape is actually meant to be a hole and to replace it with a feature of the correct size and shape.

This paper builds on previous work by our group [5] on low level 3-D reconstruction and error correction using node and vertex association. We describe our progress towards implementation of the high level approach which involves extending the interpretation of textual information from simple dimensioning such as lengths and angles to include whole phrases such as "6 HOLES DIA 5 EQUI-SPACED". Yoshiura et al. [7] proposed a related method for drawing interpretation by entering complete grammatical sentences into the computer process. These sentences were derived by an engineer from the phrases on the original drawing and of the leader lines (arrowed lines) associated with the text. By contrast the aim of the work described here is to make use of the actual phrases found on the engineering drawing. These phrases occupy a linguistic domain greatly reduced in size but including abbreviations such as "DIA." which occurs with greater frequency in drawings than in normal usage. Thus, the range of possible phrases is significantly reduced because of the restriction to phrases in the engineering domain, with consequent implications for the speed and efficiency of the interpretation process. In what follows we describe the main steps in our combined high and low level approach and demonstrate its use in identifying and locating "missing" or distorted holes.

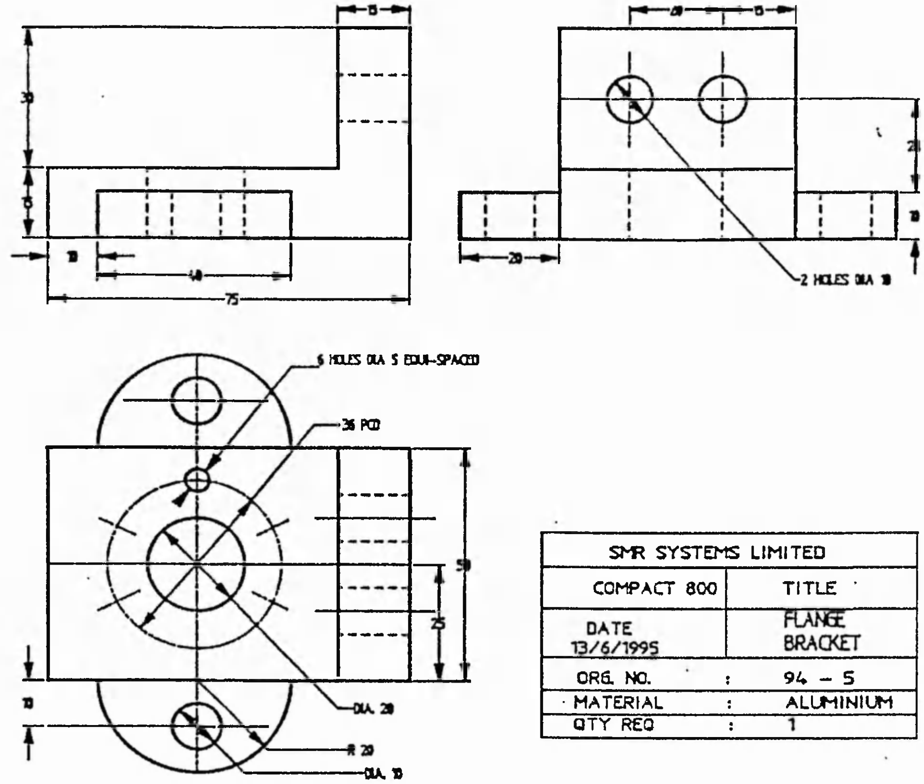


Fig. 1. An example of an engineering drawing with distorted and missing features. The text describes six holes but only one is actually drawn in the plan and none of them are shown in the other projections. In addition the hole in the front flange is distorted in the plan. This drawing cannot be interpreted correctly without using the information in the textual annotation. (The line types we discuss in this paper are listed in the Appendix.)

2 Low Level Interpretation

As explained above, the generation of a solid object by directly associating and interpreting the low level data from a simple engineering drawing is often an effective technique [8], [9]. Simple elements such as nodes and lines are combined to create vertices and edges, and then to construct surfaces. These elements are combined to create solid blocks, out of which objects are then assembled. However, errors in the drawing and distortions introduced by scanning are likely to introduce ambiguities not all of which can be resolved by direct interpretation of the low level data. In addition designers often simplify drawings and supply information in the form of

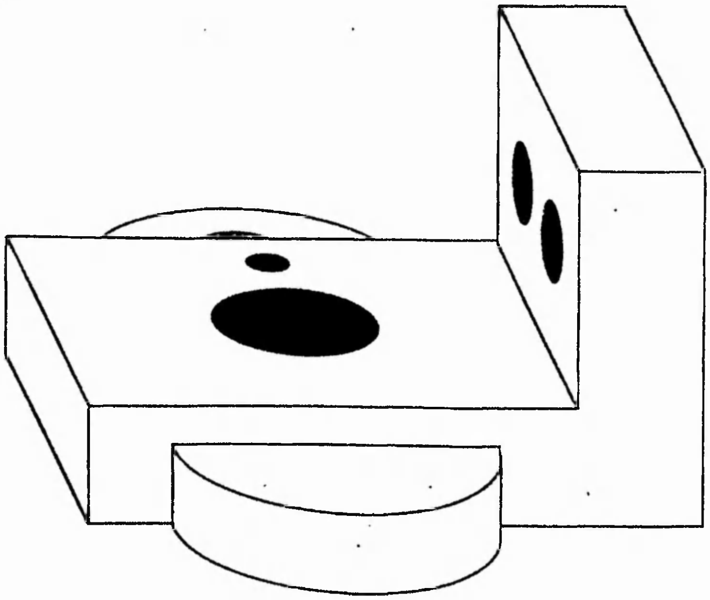


Fig. 2. The interpretation of the drawing in Figure 1 obtained using low level interpretation alone. Only one of the six holes in the top surface has been recognised and the hole nearest the front is missing.

textual annotation instead, where more appropriate, as shown in Figure 1. Therefore, an approach is needed that provides the information omitted by the simplifications and distortions. In order to generate a complete interpretation of an engineering drawing the information contained in the annotations on the drawing must be used.

3 The Combined High and Low Level Approach

The strategy for the automatic interpretation of interpreting engineering drawings using the high level details as well as low level data can be summarised as follows:

1 - The drawing is scanned and everything on the drawing (including text) is converted into a bitmap image.

2 - The captured bitmap image is vectorised (into short lines etc.) and, where appropriate, potential text blocks are identified for recognition using OCR.

3 - The low level vectorised linework is converted, wherever possible, to long line segments and circular arcs.

4 - The textual parts of the bitmap are analysed using OCR techniques to give alphanumeric phrases associated with particular positions on the drawing.

5 - The textual phrases are analysed using parsing techniques to generate associated syntactic nets.

6 - Meaning is generated using knowledge about engineering drawings which has been incorporated into the software.

7 - The meaning is used to provide extra data or to correct data in the vectorised data file (eg. the projections of the missing holes in Figure 1).

8 - The modified vectorised data file is then interpreted to reconstruct a 3-D model [5], [8], [9] using a combination of the information derived from both high and low levels.

Steps 1, 2 and 3 have already been described by other authors [1] - [9], while step 4 is accomplished by using OCR techniques. This paper concentrates on steps 5 - 7 where the high level information is analysed.

4 Engineering Drawing Text Interpretation

In general, interpreting text requires a vast quantity of knowledge. Both vocabulary and grammar rules are necessary for interpreting general textual information. However, in the case of the text on engineering drawings, the application domain can be much more limited. The vocabulary is restricted to technical terms and description of geometrical figures. Grammatical rules can also be restricted, because only assertive phrases need to be considered. The general process of analysing textual (phrasal) information on engineering drawings is expressed in the structural diagram in Figure 3. We have developed our algorithms initially for the interpretation of drawings containing three orthographic projections but clearly the principle could be extended to other types of drawings.

5 Our Approach

We have developed a strategy to derive meaning from phrases within the limited domain of engineering drawings. The phrases are assumed to provide information about sub-objects or dimensions of the object described in the drawing. (We allow sub-objects to include construction lines such as centre lines or pitch circles, although they are not "real".) For each phrase a Syntactic Net is generated, which carries information about the grammatical type of each component of the phrase and the relations between them. The text in each phrase is then interpreted and the information is used to generate a semantic template for the sub-object described. Before searching begins, the construction sub-objects are separated from the "real"

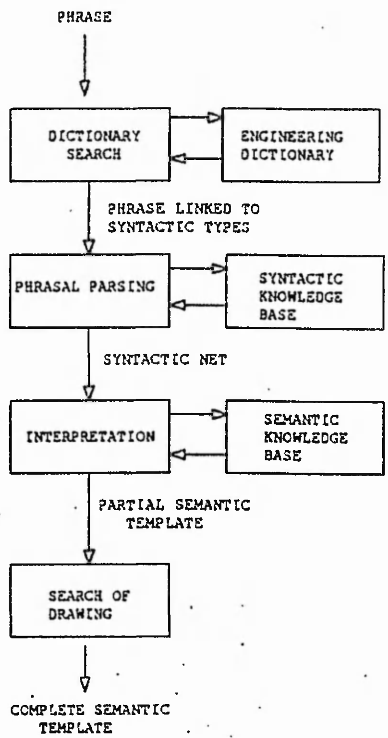


Fig. 3. An overview of the process of parsing and interpretation of a phrase on an engineering drawing.

ones and processed first. This allows for cases where the identification of the "real" sub-object depends on construction sub-objects, such as in our example in Figure 1, where the six holes have centres lying on the pitch circle described by the phrase "36 PCD". The projections on the drawing are searched for matching features, which help to determine the actual location and other details of the specific sub-objects. The searching procedure uses the information in the template and the three projections in the drawing to find all the characteristic values required to recognise and display each object on the computerised drawing.

5.1. Dictionary Search and Phrasal Parsing

Our parsing strategy is to match a coded phrase to a syntactic net stored in the knowledge base. The syntactic net consists of a root node linked to a set of elements by specific relations, with, where possible, geometrical figures as the top ranking elements and, failing that, parameters or values. Each element can be a verbal syntactic type, such as an Object (or a Parameter or Adjective associated with an

Object), or it can be a numerical type, such as a Value or Number. These elements are defined in terms of restricted syntactic categories rather than general grammatical categories. The parsing process, therefore, leads more quickly to interpretations meaningful in the domain.

Initially, we identified the syntactic types which an engineering phrase is likely to contain and we use a single letter to denote each one. Most commonly, apart from simple dimensions, phrases refer to sub-objects as part of the whole object. All of the engineering drawings which we are considering represent three dimensional objects. Therefore we define "3-D Object" as an 'O' type and assign it as the highest ranking type in the syntactic network. The characteristics of the 3-D Object are then defined by the elements constituting our particular net. Other syntactic types which occur in relevant phrases include:- "Parameter", termed 'P', representing a sub-object's parameter such as diameter; "Adjective", termed 'A', which gives additional information about the sub-object; "Preposition" termed 'R', which describes the relation between two objects. In addition, the phrase may contain numbers which could refer either to a repeated sub-object (e.g. 6 HOLES) or to a value associated with the parameter of that object (e.g. DIA 5). These two terms are numerical syntactic types, either a "Number", 'N',where a number of objects are described, or a "Value", 'V', for a value associated with a parameter, depending on their positions in the phrase, and so no further processing is needed to determine their meaning. Thus, anticipating the analysis of our specific example, "6 HOLES DIA 5 EQUI-SPACED", the Syntactic Net contains Number (6), Object (HOLES), Parameter (DIA), Value (5) and Adjective (EQUI-SPACED). The corresponding Syntactic Net therefore contains the letters A, N, O, P and V, with 'A' linked to 'O' and 'V' linked to 'P', as shown in Net #3 in Figure 4. Thus, the relationships between the tokens or sub-phrases (usually words or numbers) are made explicit. This is especially important when processing more complex phrases which may involve several objects and many parameters and their values which must all be associated correctly with the objects. The sub-phrases play an important role in leading the parser through the phrase to an ultimate interpretation.

We have based our approach partially on the method of Chung et al. [10]. In order to interpret the textual information, we need to know which projections contain the sub-object described by the text, as well as how to pass the stored knowledge about the sub-object to the searching procedure.

The interpretation starts by pre-processing the phrase, i.e. splitting the phrase into separate tokens (sub-phrases, usually words or numbers) and matching each token in the phrase to an entry in the specially compiled engineering dictionary. The tokens in the dictionary are all assigned to a syntactic category. Thus, in our example phrase, "6 HOLES DIA 5 EQUI-SPACED", "HOLES" is identified as a sub-object which is classified in the Object category ('O') and "DIA" is recognised as a Parameter ('P') related to the Object. "EQUI-SPACED" describes additional information about the holes, and therefore is assigned to the Adjective category ('A').

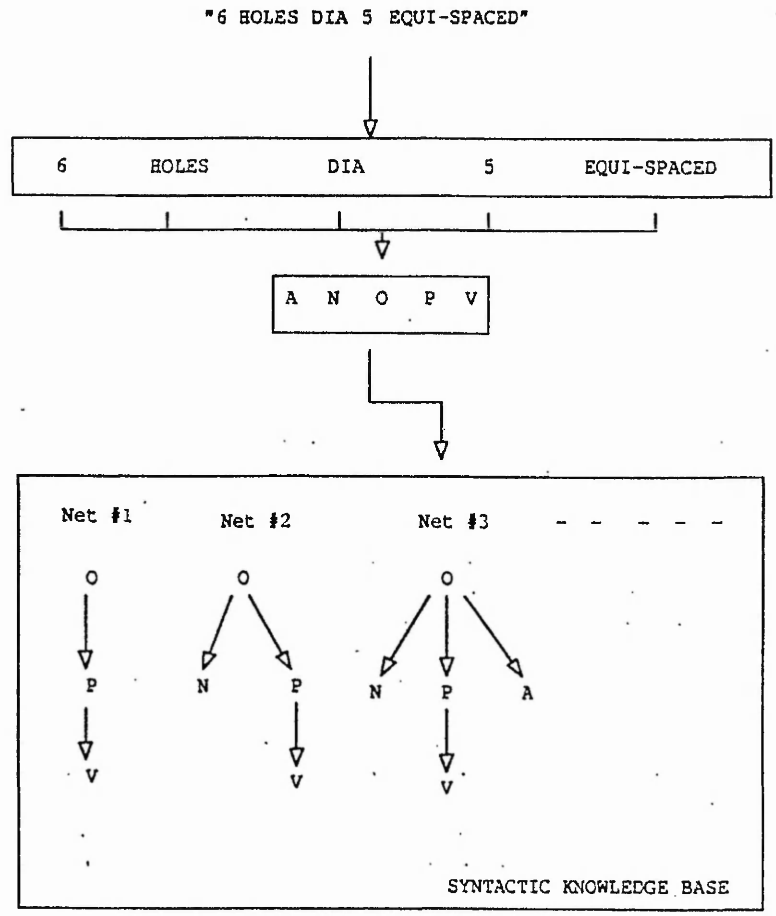


Fig. 4. An illustration of the linking of the phrase "6 HOLES DIA 5 EQUI-SPACED" to a syntactic net.

The numerical tokens are assigned to the Number ('N') or Value ('V') categories by a process which uses the position in the phrase (exploiting the fact that a value normally follows the parameter with which it is associated). Thus, in our example, the number of Holes (6) will be assigned to the 'N' category and the value of the parameter "DIA" (5) will be assigned to the 'V' category, as expected. The phrase has now been linked to the syntactic types 'N', 'O', 'P', 'V' and 'A'. The Syntactic Knowledge Base is then searched to find a matching syntactic net. The process of linking the phrase "6 HOLES DIA 5 EQUI-SPACED" to a syntactic net is illustrated in Figure 4, where it is linked to Net #3. In a similar way, the simpler example "2 HOLES DIA 10" yields syntactic types 'N', 'O', 'P', 'V' and is therefore linked to Net

#2. The phrase "36 PCD" yields initially only two tokens ("36" and "PCD") but "PCD" is further split into "PC" and "D". Thus, we have syntactic types 'V', 'O', 'P' and the phrase is linked to Net #1.

5.2. Interpretation and Searching

Once the phrase has been associated with a syntactic net, a Semantic Template is generated. The Semantic Template is associated with a particular sub-object (such as a simple hole) and specifies all the information required for that type of object (such as diameter, depth, position, orientation, etc.). In our example, the Hole Template will now indicate that there is a simple hole with diameter 5 and that this hole is repeated 6 times at equal spacing. Obviously, this information not sufficient by itself to identify the six holes on the drawing. Therefore, the procedure needs to know where these holes are located and it finds the location by searching the projections on the drawing. This searching is simplified by the fact that, in our system, phrases are stored together with a reference to the particular projection where the phrase was originally displayed, so allowing the search to be restricted initially to the projection where the phrase has appeared. The search is further simplified by exploiting information from the Leader Lines on the drawing. A Leader Line usually links a phrase to the geometrical object which is being described, as can be seen in Figure 1, so that the search can be limited initially to a small area as indicated by the leader line associated with the phrase. In the case of a hole the leader line points to the edge of the circular outline in a direction along a radius at that point. Thus, the searching function is not only focused on the appropriate projection and the most probable location on the drawing where it is likely to be displayed but it is also already armed by the knowledge base with information about the shape of the object. For example, the centre of a hole is usually indicated by the intersection point of two construction lines, eg. two centre lines or a pitch circle and a centre line. Such intersection points can be identified and tested to ascertain whether they represent the centre points of the holes and their coordinates can then be added to the template. A number of vectors approximating to a circle of the given radius confirms the presence of a hole. Figure 5 illustrates the process of associating a semantic template with the syntactic net for our example phrase. The search for the six holes in our example begins in the top view in the region at the end of the leader line and the first hole is identified when the crossing of the pitch circle with the centre line is found. The presence of the hole is confirmed by the vectors approximating to a circle of radius 5 centred on the crossing. The presence of the pitch circle together with the adjective "EQUI-SPACED" enables the other five centres to be found, even if some of the centre lines are missing.

5.3. Searching other projections

Once the sub-object has been located in one projection, the possible representations of the sub-object which may appear in the other projections are derived from the stored knowledge base. Thus, having located the projection of the sub-object, such

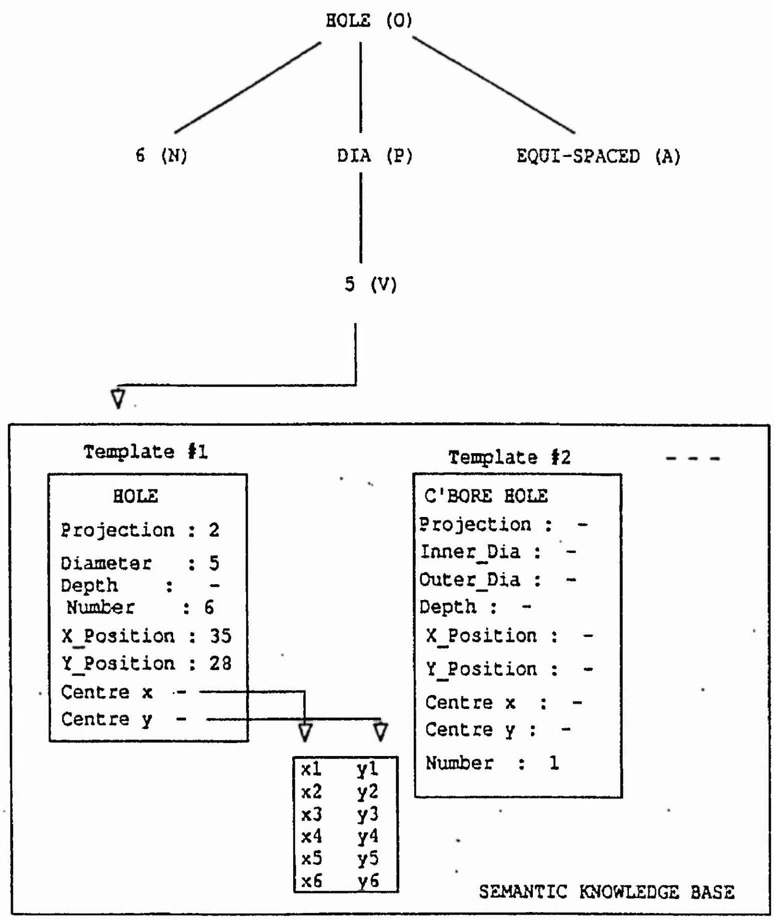


Fig. 5. An illustration of the association of a semantic template with the syntactic net linked to the phrase "6 HOLES DIA 5 EQUI-SPACED".

as the two circles representing the holes labelled "2 HOLES DIA 10" in Figure 1, the computer "expects" to find the remaining two representations of the hole depicted in the other two projections (which in this case are rectangles made up of both hidden lines and solid lines). The two projections are then searched for the appropriate representation of the sub-object; this search yields additional information about the sub-object, such as the position of the centre and, in this case, the depth of the hole (since holes are assumed to be penetrating unless specified otherwise in the text). Thus, additional information about the sub-object can be derived from the three projections to complete the semantic template. For the two holes the projections are found and the depth is found to be 15. Even when no representation is found in other

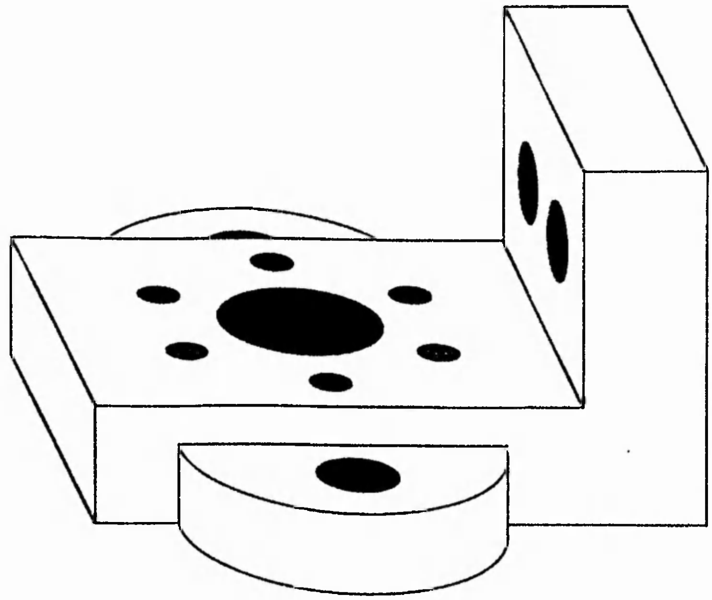


Fig. 6. The interpretation of the drawing in Figure 1 obtained using both high and low level interpretation. Comparing it with Figure 2, we see that all of the six holes in the top surface have now been recognised and the hole nearest the front has also been identified correctly.

projections, as, for example, in the case of the six holes, sufficient information can still be derived, such as the depth of the six holes, using the fact that the holes are penetrating (because the depth is not specified in the text).

6 Conclusions

This paper has demonstrated the validity of the combined high and low level approach. An algorithm is described for parsing textual information (on holes and their positions) in order to supplement the information from the low level interpretation. Without this knowledge from the textual information many holes would not be identified. In the case of the drawing from Figure 1, the interpretation of the phrase "6 HOLES DIA 5 EQUI-SPACED" enables all the six holes to be identified in spite of the fact that five of them are not shown in any of the outlines and one is shown in just one projection. The distorted hole is also identified correctly using interpretation of the phrase "DIA. 10". Figure 6 shows the resulting object complete with the set of six holes on the top surface and the hole in the front flange. The interpretation has been greatly simplified by making use of the fact that the domain is limited to engineering application. This method is an improvement on a previously reported method [7], which required an expert user to add sentences describing features on the drawing before interpretation could take place. Holes are one of the most common features described by the text on engineering drawings. Work is now in progress aimed towards extending the range of "knowledge" of our system and developing further algorithms for the interpretation of engineering drawings.

7 Acknowledgements

The authors gratefully acknowledge the support for this work given under the SERC/ACME scheme and the HEFCE as well as helpful comments from their colleagues.

8 References

[1] Joseph, S. H. and Pridmore, T. P., "Knowledge-Directed Interpretation of Mechanical Engineering Drawings", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, pp. 928-940 September, 1992.

[2] Nagasamy, V. and Langrana, N. A., "Engineering Drawing Processing and Vectorisation System", Computer Vision Graphics and Image Processing, 49, pp. 379-397, 1990.

[3] Pavlidis, T., "A Vectoriser and Feature Extractor for Document Recognition", Computer Graphics and Image Processing, Vol. 35, pp. 111-127, 1986.

[4] Karima, M., Sadha, K. S. and McNeil, T. O., "From Paper Drawings to Computer-Aided Design", IEEE Computer Graphics and Applications, pp. 27-39, Feb., 1985.

[5] Poliakoff, J. F., Thomas, P. D., Razzaq, S. M., Shaw, N. G., "3-D Reconstruction for Correction of Errors and Imperfections in Scanned Engineering Drawings", COMPUGRAPHICS'95 (To be presented).

[6] Bergengruen, O., "About 3-D Reconstruction from Technical Drawings", International Workshop on Industrial Applications of Machine Intelligence and Vision, Tokyo, pp.46-49, April, 1989.

[7] Yoshiura H., Fujimura, K. and Kunii, T. L., "Top-Down Construction of 3-D Mechanical Object shapes from Engineering Drawings", IEEE Computer, pp. 32-40, Dec., 1984.

[8] Wesley, M. A., and Markowsky, G., "Fleshing Out Projections", IBM Journal of Research and Development, Vol. 25, No. 6, pp. 934-954, 1981.

[9] Sakurai, H., and Gossard, D. C., "Solid Model Input Through Orthographic Views", ACM Computer Graphics, Vol. 17, No. 3, July 1983.

[10] M. Chung, M., and Moldovan, D., IEEE Expert, pp.36-44, 1994.

Appendix

We show in Figure A1 below the different line types discussed in this paper.

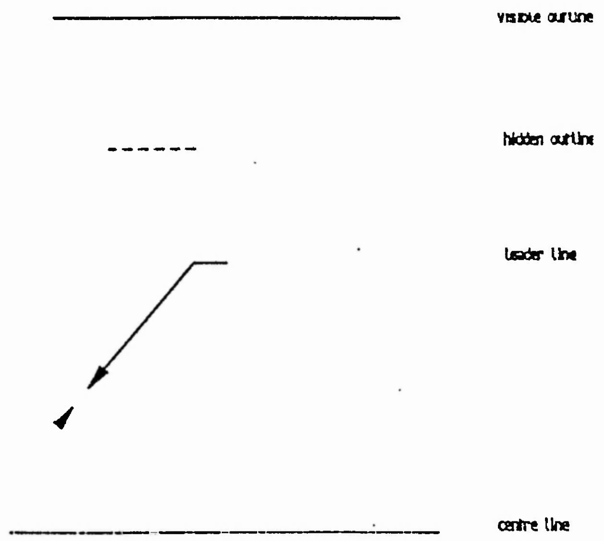


Fig. A1. Examples taken from Figure 1 to illustrate each of the different line types discussed in this paper.

The line types are:

Visible Ouline - part of the outline of the object which is visible in the given projection.

Hidden Ouline - part of the outline of the object which is not actually visible in the given projection, because it is behind another part of the object.

Leader Line - line used to associate a piece of text with the part of the object to which it refers, for example a hole.

Centre Line - a line used to indicate a centre of a feature on the drawing in a particular projection, for example of a hole.

3-D RECONSTRUCTION FOR CORRECTION OF ERRORS AND IMPERFECTIONS IN SCANNED ENGINEERING DRAWINGS

Janet F. POLIAKOFF, Peter D. THOMAS, Sabah M. RAZZAQ, N. Graham SHAW
The Nottingham Trent University, Department of Computing
Burton Street, Nottingham, NG1 4BU, England.
Tel: +44 (0)115 948 6538 Fax: +44 (0)115 948 6518
E-mail: jfp@doc.ntu.ac.uk

ABSTRACT

This paper describes algorithms for automatic detection and correction of errors and imperfections in data obtained by scanning technical drawings. The drawings are of simple machined components which contain three orthographic views and include dimensioning and other textual information. Some corrections can be made to the 2-D data of the drawing but further errors can only be detected and corrected by associating the data from different views using 3-D reconstruction. A wireframe model is constructed using an approach similar to that of previous workers and many cases of missing lines are corrected during this process. Textual information is then used to make corrections to the dimensions of the wireframe model and, finally, a corrected version of the drawing is produced from the wireframe model.

Key Words: engineering drawing, error correction, 3-D reconstruction, CAD/CAM data, image restoration, image understanding.

INTRODUCTION

An engineering drawing is a description of a technical object incorporating geometric and technical details as well as information concerning the manufacturing process. In the mechanical engineering domain, where the exact shape of an object under design is generally an essential aspect of the total product specification, large parts of the drawing pertain to the definition of 3-D geometry. The most usual method is a representation in the form of three orthographic views, which also usually incorporate dimensioning and other textual information, as shown in Fig. 1.

Engineering drawings are now usually created interactively using computer aided design (CAD) methods, with a graphics terminal as input device, and stored as computer data files. With increasing automation and integration of CAD processes, the use of such data files is growing in importance. However, many drawings still exist only on paper, so it is necessary to convert them to a suitable format before they can be used by a CAD system. The drawings are scanned and then vectorised before automatic interpretation by a computer system. At this stage some of the errors and imperfections in the data can be corrected before reconstruction is attempted. However, other errors cannot be corrected using one view

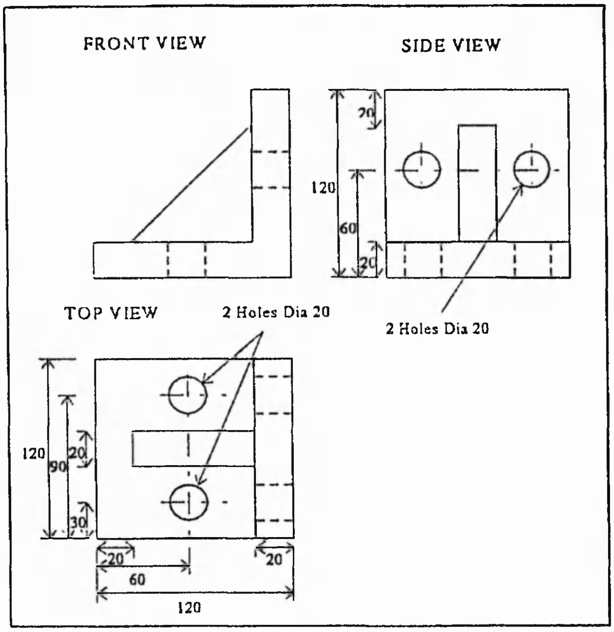


Figure 1. An example of information from an engineering drawing showing three orthographic views of an object with dimensions and textual information. The information given is sufficient to specify the shape exactly. Dashed lines indicate hidden parts of the object and chain-dashed lines represent centre lines.

alone and require information from all three views before they can be corrected.

Much research has been done into 3-D reconstruction and some success has been achieved [Idesawa73], [Idesawa-Shibata75], [Preiss80], [Markowsky-Wesley80], [Markowsky-Wesley81], [Bergengruen89], [Aldefeld83], [Aldefeld-Richter84]. We use an approach similar to that of [Sakurai-Gossard83] and [Lequette88] to construct a 3-D wireframe model of the object. Although some errors and imperfections will have been corrected before 3-D reconstruction, inaccuracies are catered for during 3-D reconstruction by allowing a certain tolerance when checking for matching between views. Corrections are then made to the wireframe model using both 3-D constraints and textual information from the drawing.

This work also extends the method of Bergengruen, who uses dimensioning information to correct each view before reconstruction. Instead we take a more global view and correct the dimensioning on the wireframe model directly. We have worked with drawings using first-angle projection but the principle is the same with third-angle projection.

IMPERFECTIONS AND ERRORS IN TECHNICAL DRAWINGS

Unfortunately, real technical drawings are often imperfect in a variety of ways. There may be a small gaps in a solid outline and sometimes a line may have been accidentally omitted altogether. In addition to this, some parts of a drawing may have been deliberately left out by the draughtsman in order to reduce the complexity of the drawing. This is standard practice in cases where no ambiguity is introduced. There will very often be a discrepancy between the actual angles and dimensions on the drawing and the values given by the text.

The process of scanning and vectorising an already imperfect drawing introduces yet more inaccuracies and errors. If the drawing is not correctly aligned when it is scanned, then a small rotation will be introduced. An error in the aspect ratio will result in stretching in one direction. Parts of lines or even whole lines may fail to be detected by the scanner, particularly if they are very thin or faded, thus increasing the number of gaps and omitted lines. On the other hand dirt or creases in the paper may be detected by the scanner as additional lines. After scanning, the discrepancy between the written angles and

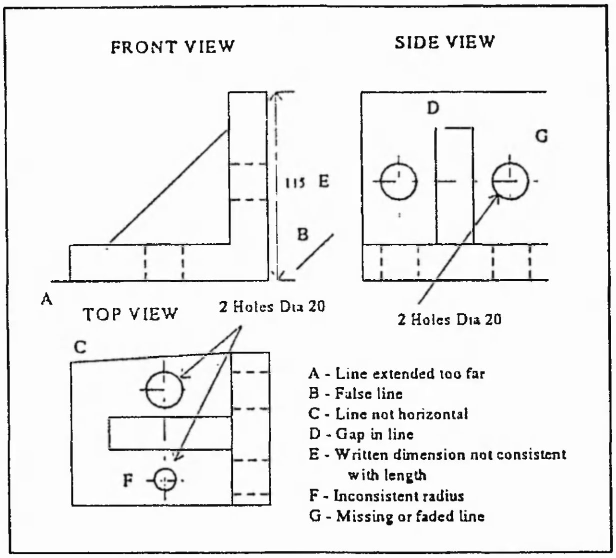


Figure 2. Examples of imperfections which can occur in a scanned drawing, some caused by defects in the original drawing and some caused or exacerbated by the scanning process. (Some dimensions have been omitted for clarity.)

dimensions and those on the drawing will generally be exacerbated. Fig. 2 shows some examples of defects that can occur.

SCANNING AND INITIAL PROCESSING

Engineering drawings are two-dimensional visual representations of three-dimensional objects and normally consist of three orthographic projections, as shown in Fig. 1. When the drawings are scanned, they are represented initially by a bitmap image. This image is then vectorised and lines and arcs are fitted where possible. Next the lines or arcs are identified as one of the following types: solid, representing part of the object; dashed, representing hidden parts of the object; and chain-dashed, representing construction lines such as centre lines. Text associated with linear or radial dimensions is also identified at this stage.

INITIAL CORRECTION OF IMPERFECTIONS ON THE DRAWING

Some imperfections can be corrected before reconstruction, for example rotation and stretching and cases A-D in Fig. 2. Rotation and stretching affect the whole drawing and need to be corrected by a global rotation and shrinking of the whole drawing.

In a projection of the outline of a solid object every line must connect with one or more lines at each of its ends. Therefore small gaps in the outline such as example D in Fig. 2, can also be corrected. If a line is found which is not connected at either end such as example B, however, it could either be part of an outline containing gaps or it could have been created by the scanning process. Thus, there is the potential for errors to be caused by such lines being connected to others or gaps being closed incorrectly.

The solution to this problem is to take a more global view of the correction process and check the consistency of the results after making any corrections. In order to achieve this a wireframe model is constructed from the three views. A failure to project onto the three views then indicates that the corrections made were erroneously.

3-D RECONSTRUCTION

We employ broadly the method described by [Sakurai-Gossard83] and [Lequette88] and explain it as follows. We define nodes as points in a 2-D drawing where lines or arcs (segments) meet and define vertices as points in 3-D space where edges of the object meet. The first stage of the 3-D reconstruction algorithm consists of

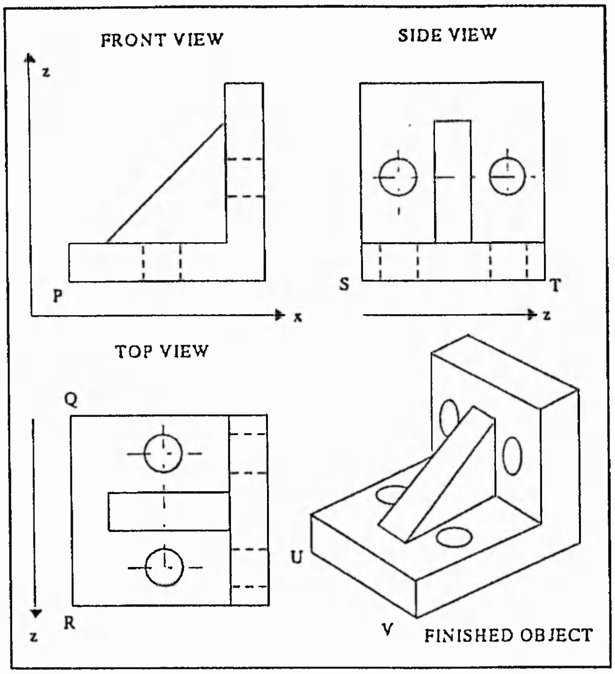


Figure 3. Matching nodes in different views to associate with vertices in the 3-D object. Node P is selected in the Front view and the Top view is searched for matching nodes which have an *x* co-ordinate within a certain tolerance of that of P. Two matches Q and R are found in this case, giving rise to two possible 3-D vertices. The vertex produced by combining the co-ordinates of P and Q is searched for in the Side view, again allowing a certain tolerance in the *y* and *z* co-ordinates. Node S is found to match, giving a new vertex U. Similarly, the vertex V produced by P and R matches T in the Side view. Each vertex is thus associated with its three projections, one node in each view.

finding all possible vertices in the object, each being associated with a node in each view. In the next stage we search for all the possible edges of the object. An edge is identified between two vertices when in each view the corresponding nodes are either joined by a line or are coincident. For a plane faced object, finding all the vertices is a relatively

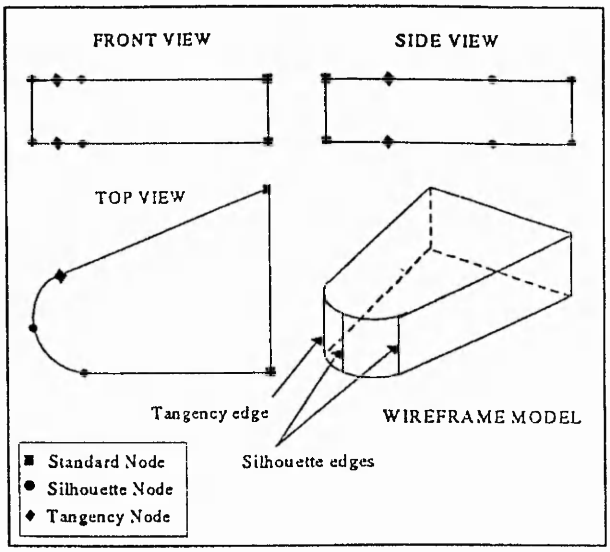


Figure 4. An example of a drawing with silhouette and tangency nodes. The Top view contains 2 silhouette nodes and 1 tangency node. Additional silhouette and tangency nodes are introduced in the other 2 views, thus creating two silhouette vertices and a tangency vertex. Silhouette and tangency edges are then introduced joining corresponding pairs of silhouette or tangency vertices in the wireframe model. (The "hidden" lines in the wireframe are shown dashed.)

straightforward process because all vertices will have projections in all three views. The matching process for such an object, illustrated in Fig. 3, will find all possible vertices in the object described by the drawing. However it is important to realise that it does not guarantee that all the vertices or edges found actually exist, as we shall see later.

If a drawing contains curves then the matching process is considerably more complicated because nodes in one view do not necessarily have matching nodes in other views, as shown in Fig. 4. We therefore define three types of nodes:

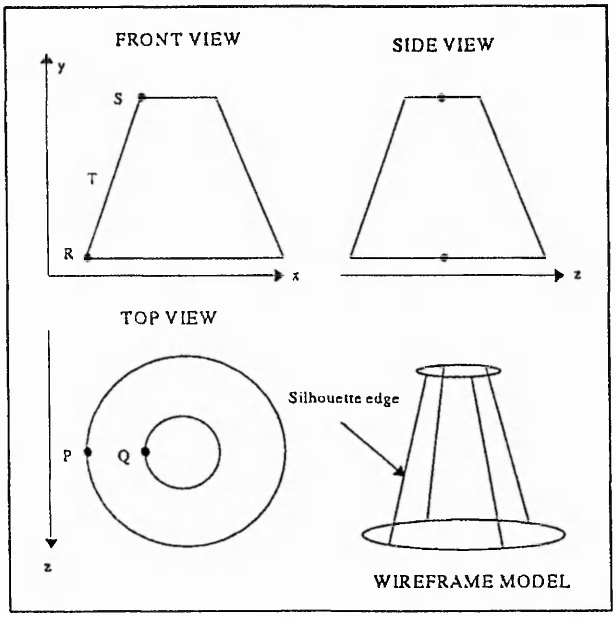


Figure 5. Another example with a silhouette edge. Nodes P and Q are both projections of silhouette vertices with the same *z* co-ordinate. A search is therefore made in the Front view for a segment connecting the projections of these two vertices. i.e. R and S. Segment T, the "silhouette", is found and a new silhouette edge is introduced connecting the vertices.

(1) standard node - a point where two segments meet (as in the case of plane-faced objects).
(2) silhouette node - a point which lies on an arc where the tangent to the arc is parallel to an axis of the drawing.
(3) tangency node - a point where two segments meet tangentially but which is not a silhouette node.

Corresponding to the three types of nodes, we then identify three types of vertices and three types of edges. In order to identify all silhouette and tangent vertices it is necessary to introduce additional nodes of these types, as we see below.

As in the case of plane-faced objects, standard vertices are identified where they can be associated with a standard node in each view. A silhouette node must have a matching standard node in one of the other views and correspond to a point on a line in the third view. This point is introduced as an additional silhouette node in that view. Then a silhouette vertex can be created which has a projection in every view. Each silhouette vertex is associated with the axis to which the arc is parallel. A tangency node does not usually have a matching node in either of the other views; instead it should correspond to a point on a line in each of the other views. As in the case of silhouette nodes, additional tangency nodes are introduced and a tangency vertex can then be created with a projection in every view. As in the case of vertices, all the edges in the object will be found but some edges may be found which do not exist in the object.

Corresponding to the case of plane faced objects, we define a standard edge as one which has a projection in all three views. In addition we introduce two new types of edge; a silhouette edge (which appears as a line in only one view) and a tangency edge (which does not appear as a line in any view), as follows. A silhouette edge is identified when it has a projection only in one view and it connects two silhouette vertices which appear in the same view and have the same value for at least one of their co-ordinates, as illustrated in Figs. 4 and 5. A tangency edge is identified when it has no projections and it connects two tangency vertices which appear in the same view and have the same value for at least one of their co-ordinates, as illustrated in Fig. 4.

Thus a silhouette edge will be introduced where a curved surface has a "silhouette" when viewed in a direction parallel to one of the orthogonal axes of the drawing. A tangency edge will be introduced where a curved surface has a "silhouette" when viewed in a direction

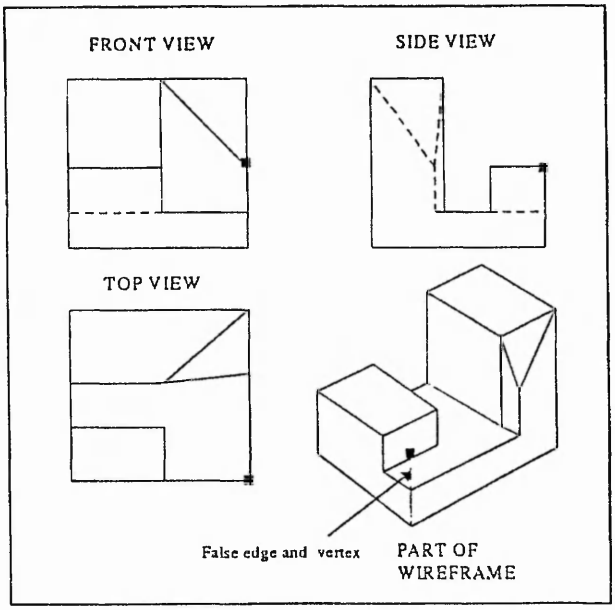


Figure 6. An example of a case where a false edge and vertex are generated; each view contains projection of the edge and the vertex. The false features will be deleted. (To avoid confusion, "hidden" lines are not shown in the wireframe model.)

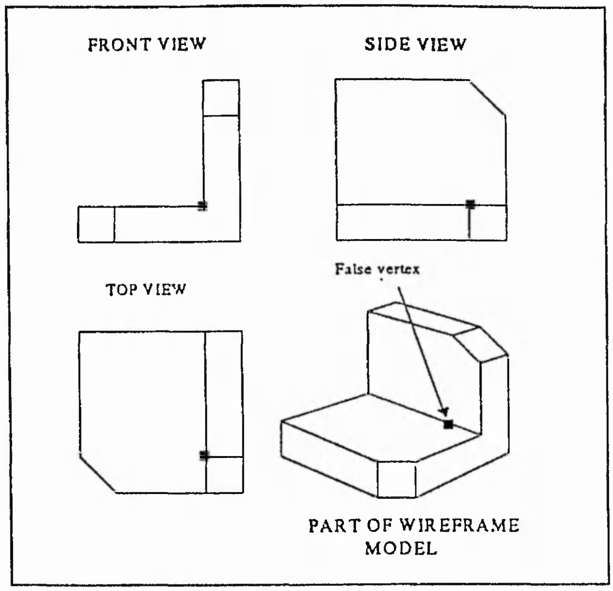


Figure 7. An example of a case which gives rise to a false vertex between collinear edges; the vertex projects onto a node in each view. The false vertex and edges will be replaced by a single edge. (To avoid confusion, "hidden" lines are not shown in the wireframe model.)

which is not parallel to an axis of the drawing. Neither are proper edges, in the sense that they do not represent discontinuities in the direction of the tangent of the surface of the object, but they are usually added to the wireframe model in order to give clearer idea of the shape of the object. For example, in Fig. 5, if silhouette edges were not included, the wireframe would consist of just two circles.

The matching methods for nodes and lines described above will, in some cases, give rise to vertices and edges which could not exist in a real object. For example, Fig. 6 shows a vertex which has been generated although it is only connected to a single edge which does not belong to any face of the object. It can be seen that the projections of the vertex and edge do appear in all views. Fig. 7 shows an unnecessary vertex which has been generated connecting two collinear edges, and Fig. 8 shows a case which gives rise to a possible false face.

REMOVING FALSE FEATURES AND FINDING MISSING EDGES

The false features described above can be eliminated by making use of some constraints which must apply to valid 3-D objects. We us the constraint that every vertex must have at least three edges connected to it. The following situations can occur:

(i) A vertex with no edges connected to it.
(ii) A vertex with one associated edge connected to it, as in Fig. 6.
(iii) A vertex with just two associated edges which are collinear, as in Fig. 7.

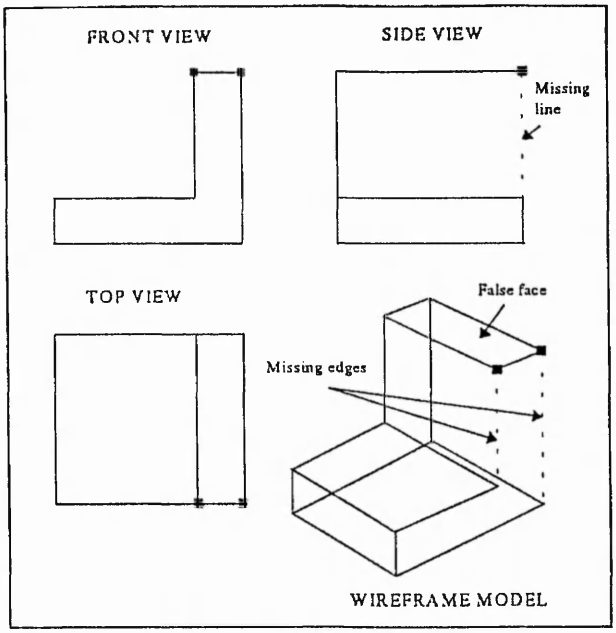


Figure 8. An example of a case where possible false faces are generated; the two vertices and three edges have projections in all three views. In this case two missing edges will be found. This enables the missing line to be identified and added to the drawing. (The full wireframe is shown here.)

(iv) A vertex with just two associated non-collinear edges, as in Fig. 8.

In cases (i) and (ii) the features are deleted and in case (iii) replaced by a single edge. In case (iv) both the vertex and the edges would also be illegal in a complete wireframe. However at this stage they are left because it is possible that there are missing edges in the wireframe as a result of missing lines in the drawing. An attempt is therefore made to find any missing edges before deleting these features.

We now proceed to look for missing edges by identifying all vertices which have only two edges connected to them. The edge finding method is repeated for these vertices but with the modification that edges are allowed which have projections in only two of the views. Any new edges found are then added on to the existing wireframe. The features (i) - (iv) are again checked and all the false features deleted this time. Many of the more frequently occurring cases of missing edges will be corrected by this method and the corresponding missing lines added to the drawing, as illustrated in Fig. 8.

CORRECTING THE DIMENSIONS ON THE WIREFRAME MODEL

The dimensions of the wireframe model may not be entirely consistent with the written dimensions on the drawing because of inaccuracies in the draughting. We now proceed to correct the wireframe model using the dimensions as follows. First, the dimensions are associated with nodes in the 2-D views for the drawing in Fig. 9. Then the associated dimensions are used to correct the positions of the corresponding vertices A-J in the 3-D model. Finally the positions of nodes in all the views are corrected, thus correcting the 3-D model and the 2-D drawings simultaneously.

Dimensions can be categorised into different types; linear dimensions (either parallel to an axis or to another axis on the drawing), radial dimensions and angular dimensions. For each type of dimension a search is made for nodes associated with the dimension. It is important that a dimension is associated with nodes rather than with a line, because in some cases there will not be a line which coincides with the dimension. In particular, radial and angular dimensions need to be associated with centre nodes that do not always have any lines connected to them. We again allow for inaccuracies by permitting a small tolerance in the search for nodes. The association takes place across views so that dimensions in one

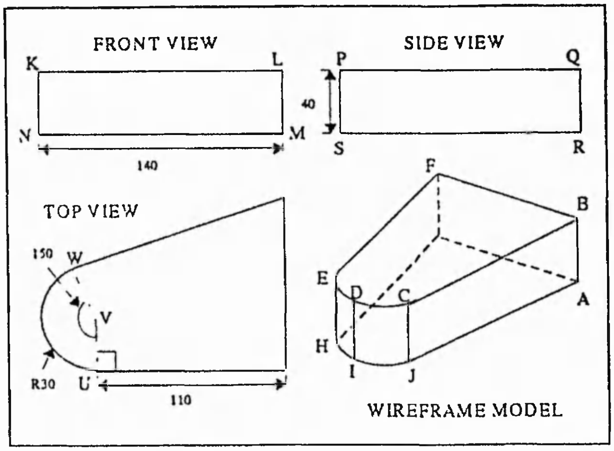


Figure 9. An example of a drawing with various linear, radial and angular dimensions. The drawing from Figure 4 is now complete with all dimensions shown. The linear dimension "140" is associated with pairs of nodes K and L, N and M respectively. The radial dimension "30" is associated with the nodes U and W with V as the centre node. The angular dimension "150" is also associated with U, W and V as the centre node. Then, once the position of vertex A is fixed, the position of all other vertices B-J can be found.

view are associated with corresponding nodes in all views, as shown in Fig. 9.

Radial dimensions must have their centre points associated with a centre node of one or more arcs on the drawing. This can be done simply by searching the relevant region of the drawing for such a node, for example node V in Fig. 9. Angular dimensions are associated with nodes along the direction of the leader lines. The node at or near the intersection point of the two leader lines is taken to be the centre node of the angle, giving node V as centre and nodes U and W on leader lines in Fig. 9.

In order to correct the whole drawing, the position of one vertex is fixed and the others are calculated from it. In Fig. 9, the vertex A is fixed and the vertices B, C, D, E and F are

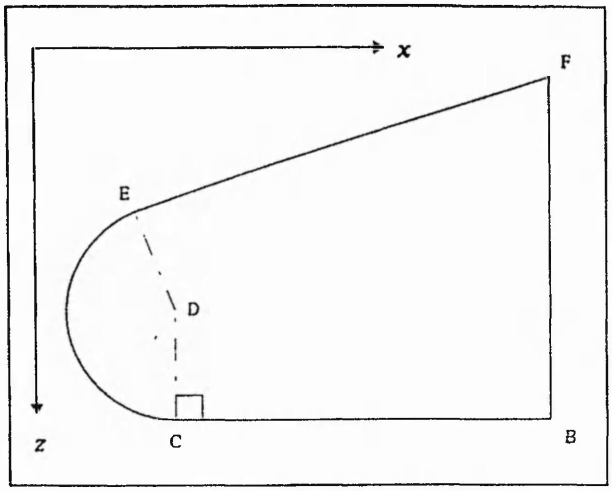


Figure 10. The wireframe model from Figure 9, viewed from above. The x and z co-ordinates of B are the same as those of A. The x and z co-ordinates of C and D can be found by subtraction and addition respectively. Node E is at a given distance from node D and at a given angle from the line DC. Node F lies on a "vertical" line from B at a given angle from another pair of nodes D, E.

calculated in order, as shown in Fig. 10. The remaining vertices can then also be calculated.

IMPLEMENTATION

2-D nodes and lines in a particular region of a drawing need to be accessed efficiently in order to match features in different views and perform various other tasks. In a large drawing searching through all lines and nodes one by one would be too slow. Therefore a square grid of cells represented by a 2-D array is superimposed on the drawing. Nodes are indexed by associating each one with the grid square that contains it. Then any nodes with a common co-ordinate will be associated with a particular row or column of grid squares. Thus, searching for nodes with common co-ordinates requires only a search across that row or along a column. Lines are indexed by associating each with every square through

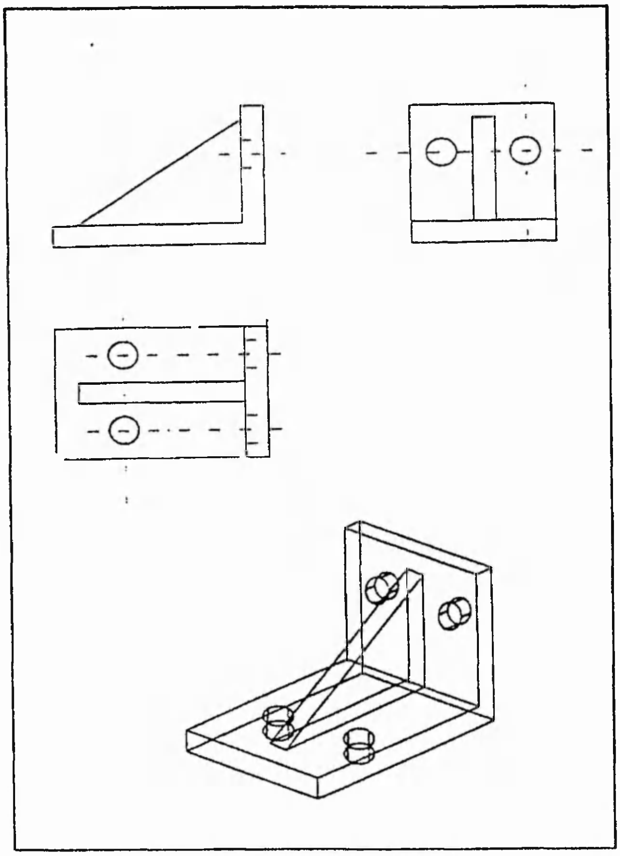


Figure 11. An example of an engineering drawing which has been corrected and reconstructed. (The dimensions have been omitted for clarity.)

which it passes. The efficiency of matching nodes and other features in different views is improved by using pointers to link edges to vertices, vertices to corresponding nodes, as well as nodes to line segments and vice versa.

Silhouette nodes are assigned to one of three types according to the axis along which the "silhouette" will be seen. This enables a more efficient search to be made for the silhouette edge.

The algorithms described in this paper have been implemented in C run under the UNIX operating system on a SUN workstation.

CONCLUSIONS

The algorithms described here are an improvement on other existing methods for automatic interpretation and reconstruction of 3-D models from technical drawings. Many errors and imperfections in the original drawing and those introduced by the scanning and vectorising processes are detected and corrected before reconstruction is attempted. During 3-D reconstruction, a certain tolerance is permitted in matching features in order to allow for inaccuracies. 3-D constraints are used to eliminate erroneous lines and replace some missing lines. Corrections are then made to the wireframe model using dimensioning and textual information and these correction are then reflected back into all three views, thus correcting all three views simultaneously. Fig. 11 shows an example of a drawing which has been corrected and reconstructed.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support for this work given under the SERC/ACME scheme, the interest and support of Pafec Limited, and the help and advice of Professor R. J. Whitrow.

REFERENCES

Aldefeld, B. 1983 On Automatic Recognition of 3-D Structures from 2-D Representations, *Computer Aided Design*, Vol 15, no 2, pp 59-63.

B. Aldefeld, B., Richter, H. 1984 Semiautomatic Three-Dimensional Interpretation of Line Drawings, *Computers and Graphics*, Vol 8, no 4, pp 371-380.

Bergengruen, O. 1989 About 3-D Reconstruction from Technical Drawings, *International Workshop on Industrial Applications of Machine Intelligence and Vision*, Tokyo, pp 46-49.

Idesawa, M. 1973 A System to Generate a Solid Figure from a Three View, *Bull. JSME*, Vol 16, no 92, pp 216-225.

Lequette, R. 1988 Automatic construction of curvilinear solids from wireframe views, *Computer Aided Design*, Vol 20, no 4, pp 171-180.

Idesawa, M., Shibata, S. 1975 Automatic Input of Line Drawing and Generation of Solid Figure from Three-View Data, *Proceedings of the International Joint Computer Symposium*, pp 304-311.

Markowsky, G., Wesley, M. A. 1980 Fleshing Out Wire Frames, *IBM J. Res. Develop.*, Vol 24, September, no 5, pp 582-597.

Markowsky, G., Wesley, M. A. 1981 Fleshing Out Projections, *IBM J. Res. Develop.*, Vol 25, no 6, pp 934-954.

Preiss, K. 1980 Constructing the 3-D Representation of a Plane-Faced Object from a Digitized Engineering Drawing, *Fifth International Conference and Exhibition on Computers in Engineering and Building Design*, Brighton, pp 257-265.

Sakurai, H., Gossard, D. C. 1983 Solid Model Input Through Orthographic Views, *Computer Graphics*, Vol 17, no 3, pp 243-252.