THE UNIVERSITY *of* EDINBURGH

COMPUTER RECOGNITION OF

OCCLUDED CURVED LINE DRAWINGS

MARK RONALD ADLER

Ph.D. THESIS

UNIVERSITY OF EDINBURGH

1977

Computer Recognition of Occluded Curved Line Drawings

ABSTRACT

A computer program has been designed to interpret scenes from PEANUTS cartoons, viewing each scene as a two-dimensional representation of an event in the three-dimensional world. Characters are identified by name, their orientation and body position is described, and their relationship to other objects in the scene is indicated. This research is seen as an investigation of the problems in recognising flexible non-geometric objects which are subject to self-occlusion as well as occlusion by other objects.

A hierarchy of models containing both shape and relational information has been developed to deal with the flexible cartoon bodies. Although the region is the basic unit used in the analysis, the hierarchy makes use of intermediate models to group individual regions into larger more meaningful functional units. These structures may be shared at a higher level in the hierarchy. Knowledge of model similarities may be applied to select alternative models and conserve some results of an incorrect model application. The various groupings account for differences among the characters or modifications in appearance due to changes in attitude. Context information plays a key role in the selection of models to deal with ambiguous shapes. By emphasising relationships between regions, the need for a precise description of shape is reduced.

Occlusion interferes with the model-based analysis by obscuring the essential features required by the models. Both the perceived shape of the regions and the inter-relationships between them are altered. An heuristic based on the analysis of line junctions is used to confirm occlusion as the cause of the failure of a model-to-region match. This heuristic, an extension of the T-joint techniques of polyhedral domains, deals with "curved" junctions and can be applied to cases of multi-layered occlusion. The heuristic was found to be most effective in dealing with occlusion between separate objects; standard instances of self-occlusion were more effectively handled at the model level.

This thesis describes the development of the program, structuring the discussion around three main problem areas: models, occlusion, and the control aspects of the system. Relevant portions of the program's analyses are used to illustrate each problem area.

ACKNOWLEDGEMENTS

I am grateful to my supervisors, Jim Howe and Sylvia Weir for their help and advice over the last four years.

I should also like to thank the other members of the Department of AI, past and present, for their friendship encouragement and cooperation especially: Bill Clocksin, Ben du Boulay, Marc Eisenstadt, Ricky Emanuel, John Knapman, Claude Lamontagne, George Luger, Marilyn McLennan, Tim Radford, Robert Rae, Andy Russell, and Richard Young.

My largest debt is to my wife, Linda, for her encouragement, patience, and love; her help in producing this thesis; and her financial support during our years in Scotland.

I also acknowledge the limited financial support from the Social Science Research Council.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 The Peanuts Universe

This work is concerned with the recognition of curved line drawings. A system was designed and implemented to interpret a restricted subset (see Section 1.3) of PEANUTS cartoons as two-dimensional representations of events that occur in the three-dimensional world. Much previous research in machine recognition of scenes has been concerned with the analysis of *static planar* objects; our domain takes us into the realm of flexible objects. We must contend with object shapes that are irregular and flexible. Our program must recognise the PEANUTS characters as they assume various positions and orientations in the scenes. Occlusion adds further complications to the recognition procedure. Provision must be made to allow the recognition process to succeed when only part of an object is visible, and not to confuse a partially occluded object with something else.

To cope with these problems, we have used a *hierarchy of models* to guide the region-based analysis of the scene. The information contained in the hierarchy can be classified into two categories:

(1) Knowledge of shapes of *regions*;

(2) Knowledge of relationships between component parts of an object. (Component parts of an object may be regions or groups of regions.)

These two types of knowledge are bound together to reflect the observed changes in the appearance of objects as the viewing angle changes, or when a flexible object assumes a different position.

The hierarchy contains four levels:

(1) Structure models
(2) Component models
(3) Description models
(4) Composition models.

The Structure and Component models deal with the relationships between regions and groups of regions. Since the relationships between body parts may change there are different models to represent various configurations. The Description and Composition models pertain to the shape of regions. They contain detailed descriptions of the expected shapes of the regions that make up the objects in the domain.

In addition to the models, procedures are provided which are invoked to deal with the problems of occlusion. These procedures are designed to examine the information in the scene at the level of the lines and junctions to decide whether or not a particular region is partially hidden from view. The method is based on the simple fact that when two objects overlap, the boundary of one object disappears at one point and then re-appears at another. By finding this pair of junctions, we can determine the occlusion characteristics for the regions concerned.

Together with models and occlusion handling we also consider the problems of control strategy. This covers a variety of control decisions concerning the method of scanning the scene and the interaction of the model information with the occlusion heuristics.

We have classified problems in this environment into three main categories:

(1) Models -- the problem of shape description and recognition;

(2) Occlusion and its effect on recognition;

(3) Control strategy for scanning the scene.

These three problems will be described in detail in Chapters 3, 4 and 5, but first we will present some examples of the types of scenes we are dealing with and the nature of the description obtained from the scenes.

1.2 Examples

As a first example of the type of scene and the resulting analysis consider Figure 1.1. The top-level information obtained from this scene is as follows:

Who: LUCY

View: FACING FRONT-RIGHT (of screen)

Body Position: STANDING

In addition to this information, the program has found a corresponding model part for every individual region or functional group of regions in the input scene. So, for instance, region R0208

Figure 1.1

corresponds to the sock on Lucy's right leg and region R0198 is labelled as her skirt.

In this scene there are four instances of occlusion not expected by the standard models:

(1) The RIGHT-ARM occludes the LEFT-ARM

(2) Both ARMS occlude the FACE

(3) Both ARMS occlude the TORSO

(4) The RIGHT-SHOE occludes the LEFT-SHOE

It is sometimes difficult for the human viewer to appreciate the difficulties involved in recognising an occluded shape. Figure 1.2a shows several of the occluded regions of Figure 1.1 taken out of context to illustrate the difficulties involved. In Figure 1.2b we see the same shapes again, this time in conjunction with an occluding region. This illustrates how valuable information about occlusion can contribute to the identification process. There are general occlusion techniques which are employed to establish that a region is occluded and to find the occluding region. With this knowledge, the recognition task of the models is simplified.

In Figure 1.3, we have another example of a PEANUTS scene. This time there are other objects in the scene in addition to LUCY. The top-level information for this scene extracted by the analysis system is given below.

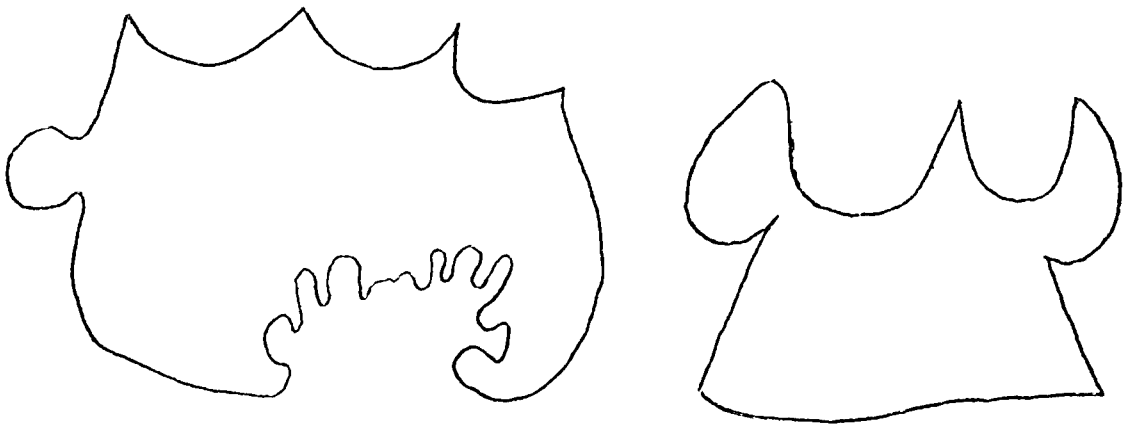


Figure 1.2a

Figure 1.2b

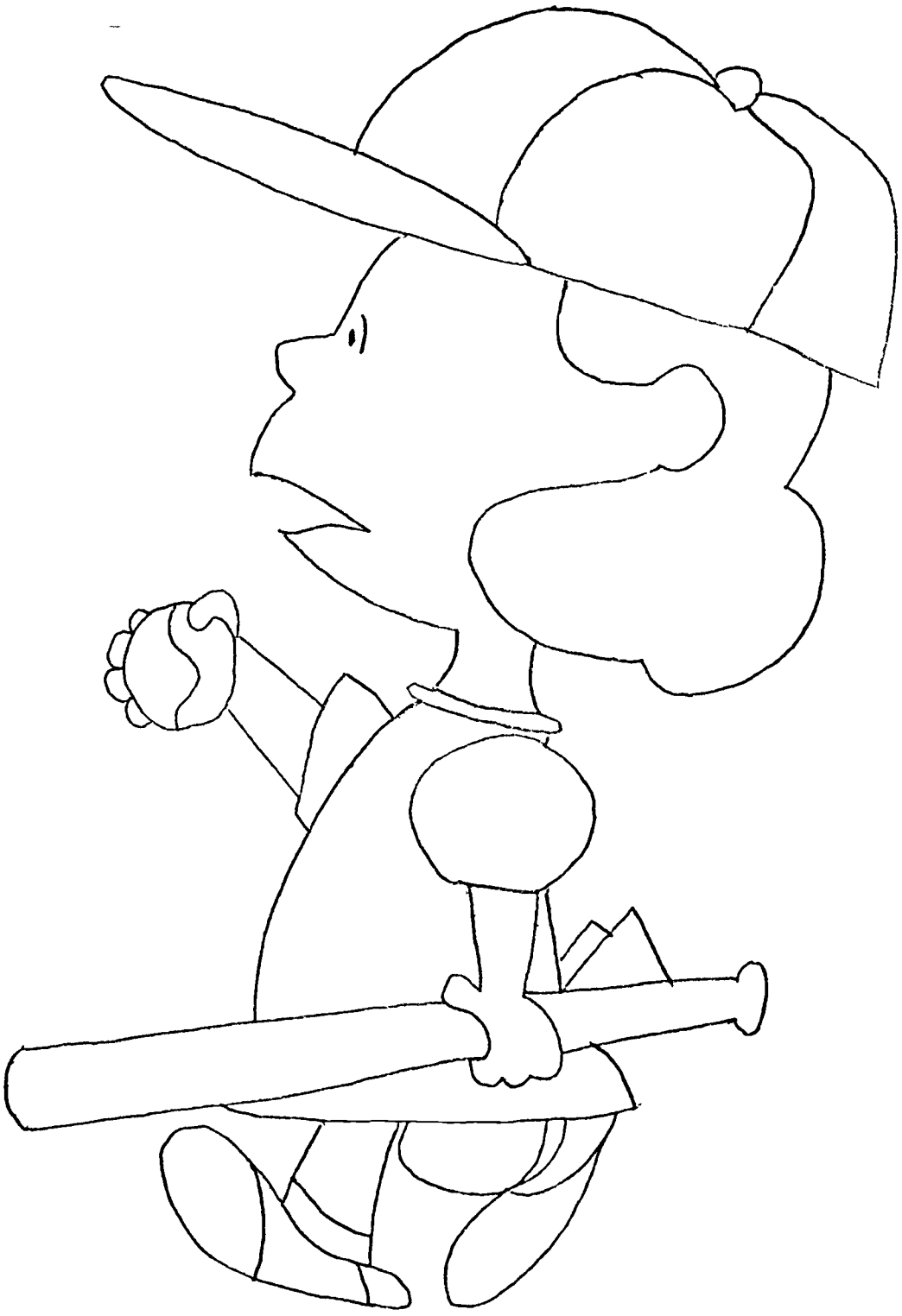


Figure 1.3

Who:	LUCY
View:	FACING LEFT (of screen)
Position:	WALKING
Objects:	BASEBALL-CAP on HEAD
	BASEBALL in RIGHT-HAND
	BASEBALL-BAT in LEFT-HAND

This scene illustrates the global-context mechanism of the program. In this case, the context is BASEBALL. The context mechanism effectively shuffles all baseball related items to the top of the list of expected items in the scene. The detection of one item of the group triggers this re-ordering of the possibilities. Shapes very similar to that of the ball and bat may have depicted other objects in a different scene, but the unmistakable stimulus of the baseball cap forces the baseball context interpretation.

Finally, in Figure 1.4 we see an instance of a scene which does not conform to the model's expectations. The region that should have corresponded to LUCY's skirt has been distorted to resemble a planar surface, such as a table-top. A closer inspection reveals that this region cannot be interpreted as a table since the surface appears to be both behind and in front of LUCY's body. The program still produces a top-level result of:

Who:	LUCY
View:	FACING FRONT-RIGHT (of screen)
Position:	STANDING

However, it adds items to its data-base to indicate that:

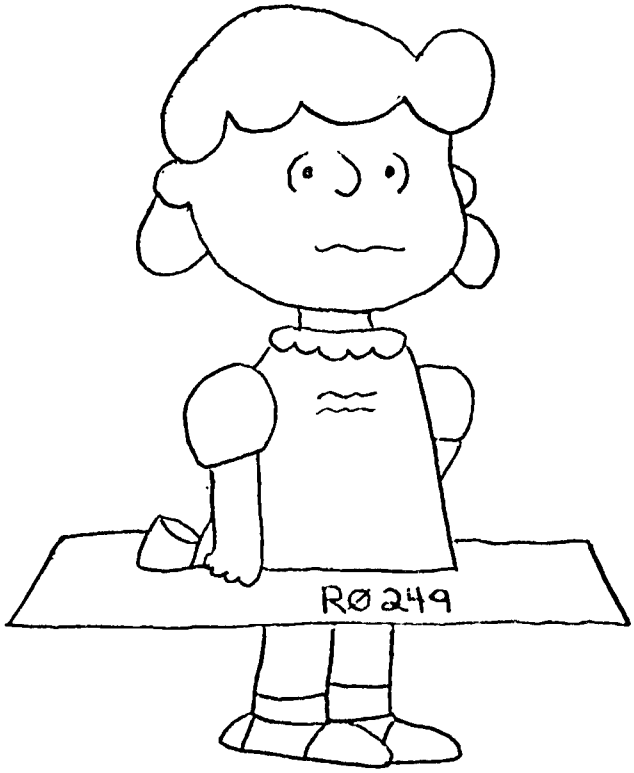


Figure 1.4

(1) The SKIRT portion of LUCY's dress could not be found.

(2) The region R0249 could not be matched to any model.

Although the entire scene did not agree with all the model parts, there was sufficient evidence to recognise LUCY from her head and torso. The success of partial results produced enough evidence to override the confusion resulting from the distortion of the SKIRT area. The model for PERSON continued the scan to find the undistorted legs in their expected place.

1.3 Further Explanation of the Domain

We chose the PEANUTS universe because it seemed an appropriate domain for the problems we wished to study, namely the recognition of irregular shapes and the effects of occlusion on the recognition process. The choice of this domain clearly takes us out of the world of regular geometric shapes. We feel that these irregular line-drawings bring us closer to the analysis of real-world scenes although there are many differences between the two domains.

A significant difference between the cartoon domain and the polyhedral domains studied in the past is the fact that very precise descriptive methods that depend on strict geometric analysis cannot be applied. The basic shapes are too irregular -- they vary from scene to scene. Although the scenes represent events in the three-dimensional world, the characters are not drawn as strict

geometric projections; rather they are cartoon characters conforming to a well-developed system of cartoon conventions.

There are certain characteristics of this cartoon world that simplify the recognition task. An obvious advantage is the simple style of the PEANUTS drawings. It is easy to isolate the closed regions which we use as the basic primitives (which roughly correspond to surfaces in the real world). A more significant benefit is related to the skill of the cartoonist. We can rely on him to capture the distinctive elements of an event and express it in a simple and symbolic manner. Therefore, we need not worry about confusing overlaps of objects or other ambiguities. There will always be sufficient information present because the cartoonist has arranged it for us.

Within this cartoon environment, we have made several further simplifications. We have not designed the system to handle all PEANUTS cartoon scenes. The program has been tested on scenes of isolated PEANUTS characters as shown in Section 1.2. The complete set of computer analysed scenes is shown in Figure 1.5. These scenes will be described in the following chapters. All scenes contain a person, usually with some other object such as baseball equipment present in the scene. The character may assume any body position. Naturally, the number of characters and objects that the system is capable of recognising is limited by the models available. (Models

of Charlie Brown, Lucy, and Violet were used for the analyses.) By adding more models the scope of the system may be extended, but the processing time for the scene will be increased (see Chapter 3). We have not considered certain types of cartoonist conventions such as the addition of "noise" to represent falling rain or snow nor have we allowed "multiple exposure" scenes which are beyond the scope of our models (see Figure 1.6).

1.4 Structure of this Thesis

The basic layout of this thesis is as follows.

Chapter 2 describes the previous approaches to the Vision/Recognition Problem. It contains a discussion divided into the same three categories that have been used to structure this whole thesis, namely: Models, Occlusion, and Control. We contrast the various alternative methods with our own and discuss the limitations.

Chapters 3, 4 and 5 form the main body of the presentation. They discuss Models, Occlusion, and Control respectively. Each section contains a description of the techniques incorporated as well as detailed examples of the analysis of selected scenes.

Chapter 6 is a discussion of the problems encountered in the development of this system. This section is also sub-divided into the three aforementioned categories.

Finally, Chapter 7 summarises our conclusions and sets out possibilities for future work.

Figure 1.5

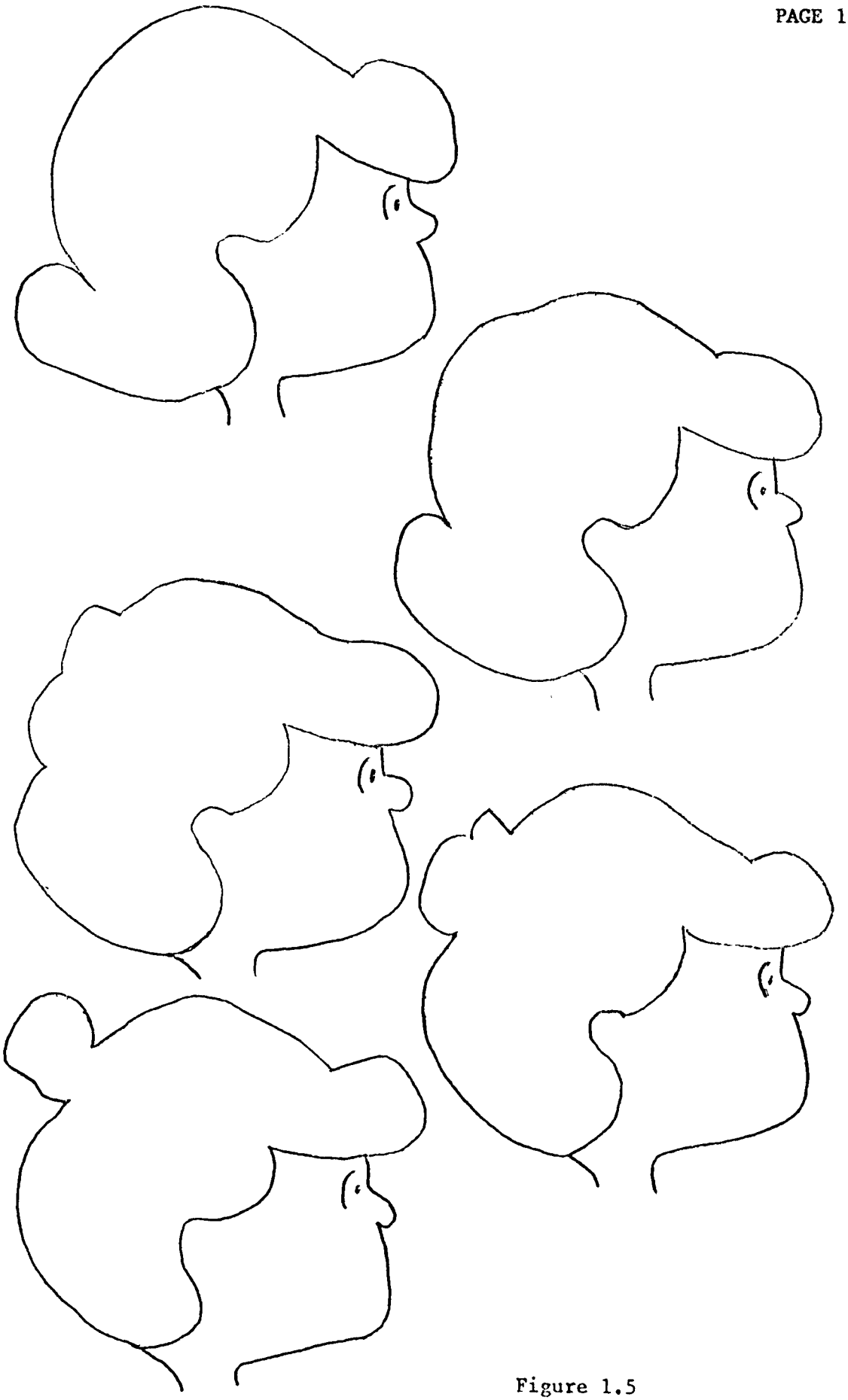


Figure 1.5

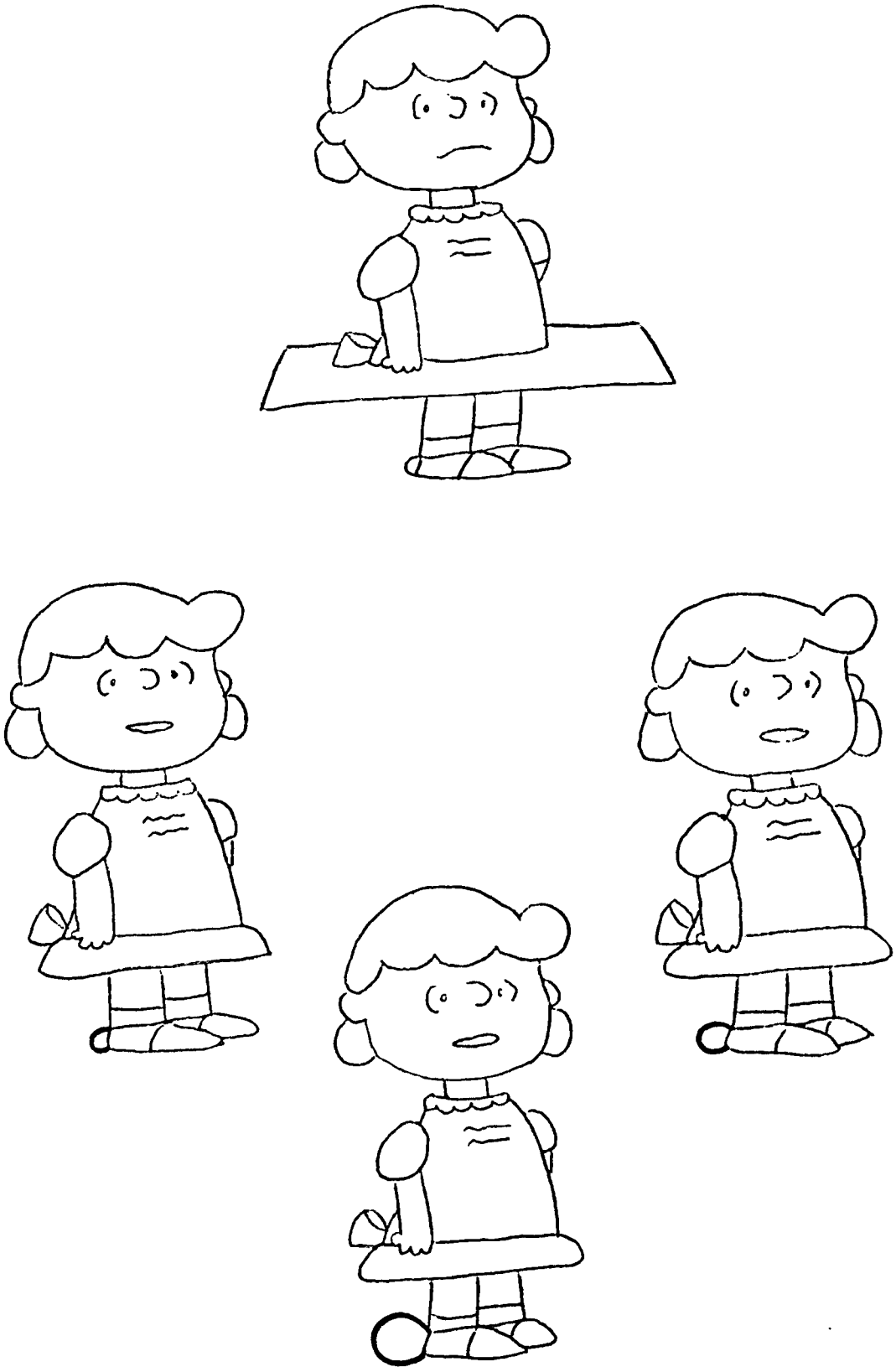


Figure 1.5

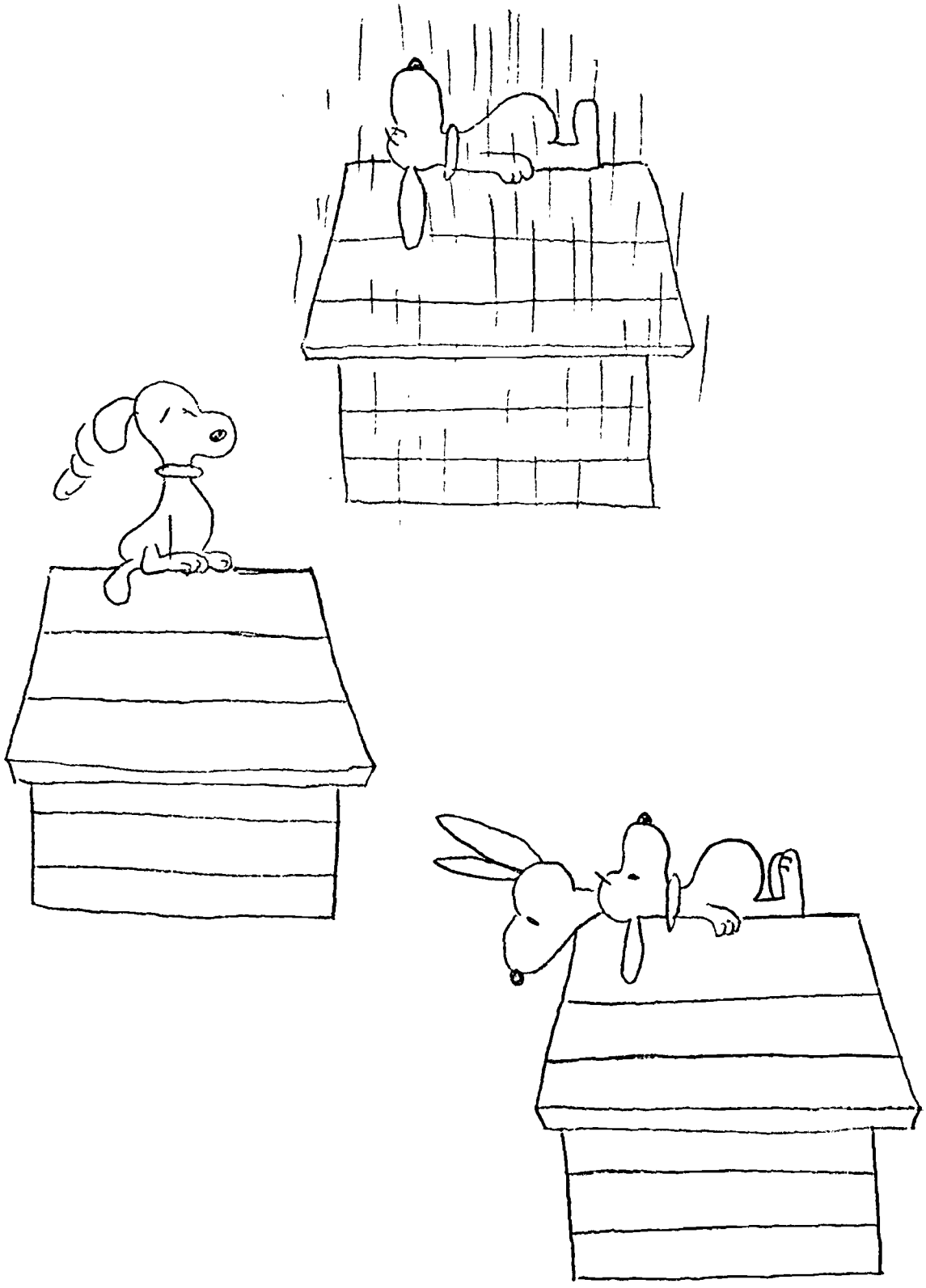


Figure 1.6

CHAPTER 2

RELATED RESEARCH

In this chapter we discuss some of the related vision research in artificial intelligence (AI). As vision in AI branches into new domains and investigates different techniques, it becomes increasingly difficult to compare the different approaches of one piece of research with another. However, there are some common sub-problems and this section will concentrate on those while pointing out how the domain and specific goals of each piece of research influence and often restrict the generality of the solution.

Our own research is concerned with the use of models and the problems of occlusion in the recognition of curved-line drawings (cartoons). We feel that the most interesting and complicated task in a vision program is the process of matching the visual data (whether they are regions, lines or points of light) to a higher-level model, i.e. recognition. Our selection of related works will reflect these prejudices. We will omit works mainly concerned with line finding and segmentation of TV images.

We have divided this chapter into two sections. The main section is further sub-divided into the three main problem areas that we use throughout this thesis: models, occlusion, and control.

The choice of models often influences the nature of the control mechanisms and the method of handling occlusion. For this reason we may discuss the same piece of research in more than one section.

The final section briefly discusses some related psychological theories of perception and serves as an introduction to the next chapter which is an explanation of our model system.

2.1 Related AI Research

2.1.1 Models

In this sub-section we discuss various approaches to the problem of coding information to be used for the recognition process. The domain chosen for each study strongly influences the types of models that are selected. We begin this survey by discussing the early research in the domain of polyhedra (the blocks world) and proceed in a roughly chronological order to cover richer domains.

While research in this domain has led to many advances in the AI vision field, particlarly in TV image interpretation and program control mechanisms, all too often the over-simplified domain constrained the interpretation. Hence the results in one domain often cannot be extended to meet the challenge of a more complicated

domain. Nevertheless, any discussion of AI vision cannot ignore these early blocks world programs and their influence on the programs of today.

2.1.1.1 Blocks world

Roberts' [1965] work on the recognition of three-dimensional solids is usually recognised as the first AI vision program. His program was able to interpret line drawings (extracted from a camera image) as three-dimensional objects. In this geometric world, Roberts applied mathematical models based on three prototypes:

1) Wedges
2) Rectangular prisms
3) Hexagonal prisms.

These models were manipulated mathematically (using translation, rotation and scaling) until their projections matched those of parts of the image. Unfortunately, this technique does not yield a unique decomposition for objects formed from the combination of two or more prototypes (see Figure 2.1).

Roberts' work initiated a series of projects concerned with the recognition of simple polyhedral scenes. For example, Guzman's program, SEE [1968] used some very ad hoc rules based on line junctions to group the regions of a scene into "bodies". Occlusion is a hinderance in this task. By occlusion we mean the effect of one object hiding all or part of another object. The term occlusion

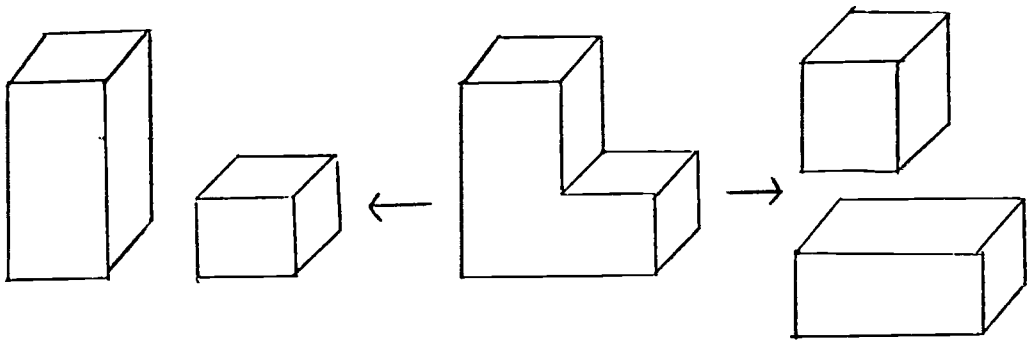


Figure 2.1 Two decompositions of an 'L'

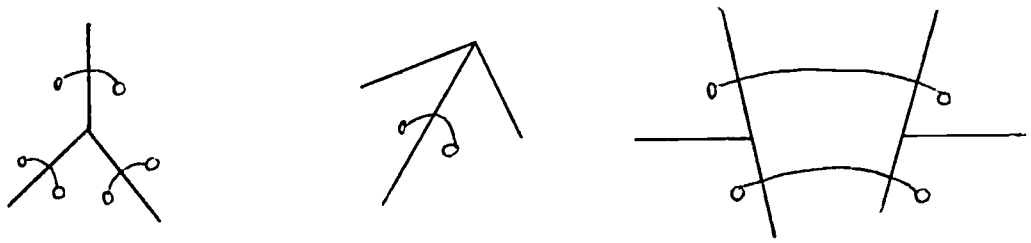


Figure 2.2 Some of Guzman's vertices with links

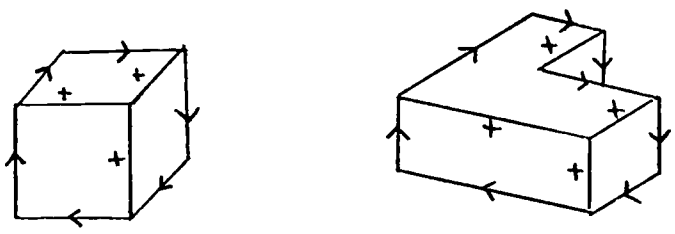


Figure 2.3 Labelled polyhedra

usually refers to only a partially obscured object. If an object is completely hidden it is usually undetected by the program unless its presence is indicated by higher level knowledge. Matching partial images to models is a difficult problem (see Section 2.1.2). The rules which guided SEE were based on the very local evidence of the vertex configurations of the blocks. The rules, or linking and inhibiting heuristics, were based on observations of typical block configuration. Basically, line junctions commonly found in un-occluded instances of blocks caused links to be formed among the regions surrounding the vertex; line junctions usually formed by occlusion (e.g. T-junctions) placed inhibiting links between the regions (Figure 2.2).

In this system, the knowledge of well-formed objects was dispersed throughout the system in terms of vertex configurations, rather than confined to prototype definitions. Guzman's rules had no formal theory to back them up; they were not always successful, tending to be too liberal in proposing links between regions.

Clowes [1971] and Huffman [1971] provided the formal theory based on global mathematical constraints of the edges and vertices of polyhedra, labelling each line according to its role in the scene. Lines could be labelled as occluded or occluding, or concave or convex edges. In the blocks world, each line has only one labelling for each consistent interpretation. (See Figure 2.3). By finding a

consistent labelling for all the lines (i.e. exactly one label for each line) the structure of the scene is established. Furthermore, if no such labelling can be found then the line drawing cannot correspond to an image of true polyhedra. It is an "impossible object" Huffman [1971].

Armed with this labelling scheme for line drawings, researchers devoted their efforts to the problem of extracting "perfect line drawings" from TV images. Much work was devoted to "simplifying" the visual task by simplifying the initial data. Shadows and texture were eliminated and only straight-edged objects were studied. The results of Waltz [1972] were influential in a re-appraisal of this attitude. His work on the interpretation of line drawings of trihedral solids with shadows showed that more information made the processing easier. Essentially, Waltz extended the labelling scheme to handle more complex scenes, i.e. those with shadows. This introduced new labellings for lines to account for the boundary between light and dark, but more importantly, it added new constraints to the allowable labelling of neighbouring lines of the scene. By applying these constraints over the entire scene in an approximation to parallel processing, Waltz was able to eliminate all inconsistent labellings quickly and efficiently. While this scheme may have applications for some problem solving domains, its applications in more complicated visual environments is severely limited. It requires a very simple domain such as the very geometric blocks world to allow its filtering of constraints to be successful

since it requires total knowledge of all possibilities; in a less constrained world total knowledge is harder, if not impossible, to capture.

Waltz's success more or less heralded the end of AI's fascination with polyhedral scenes. But there are two further pieces of research in blocks world that are interesting because of the very different ways in which this world is modeled.

Mackworth [1974] also studied scenes of perfect line drawings but his model scheme was based on a 'dual picture-graph' mapping of the image plane into a corresponding two-dimensional gradiant space. The resulting representation was more easily analysed than the original line drawing. While this was an ingenious scheme it was limited to polyhedral scenes while the Waltz labelling technique has been extended to domains of curved objects (see below).

Grape's work [1973] illustrates quite a different approach to the analysis of blocks world scenes. Grape's program uses TV images rather than perfect line drawings. His models are capable of coping with the imperfect data that is provided. Rather than using models based on lines or junctions, Grape bases his analysis on "intermediate" structures (e.g. lines with a Y-junction at both ends). There are two main advantages to using such structures. First, these larger structures provide a degree of context

information which is useful in overcoming the imperfect data. Second, such larger structures usually have properties which are considerably different from the smaller parts from which they are composed. In a sense, by using well-chosen groupings of sub-structures, the recognition process is simplified.

Paul [1977] also makes extensive use of intermediate models in the domain of pictures of a puppet constructed from polyhedra. He uses two-and-a-half dimensional models to describe the appearance of the three-dimensional puppet from various view-points. The parallel lines present in the pre-processed input to his system are grouped to form regions -- the intermediate-level structures of the two-and-a-half dimensional models. The input to the system is allowed to vary over a range of puppet representation. The shapes of the polyhedra which form the puppet are not fixed. They may be hexagonal prisms, cylinders, rectangular prisms, etc. The detailed shapes are not crucial since recognition is based on the intermediate models.

2.1.1.2 Curved objects

Not all the early vision systems restricted their domains to convex polyhedral objects. Barrow and Popplestone [1971] included everyday objects such as cups and spectacles (along with simpler geometric objects) in their domain. In this world of curved objects, the

region replaces the line as the fundamental building block used for recognition. Most blocks world programs relied on the line because of its important role in that restricted domain. In irregular visual worlds the region captures more information than the lines which form it. In isolation, lines may have several possible interpretations. They may be edges of surfaces (as they always are in the blocks world) or the point where a curved surface disappears from view. A region always corresponds to a surface, whether it is planar or curved.

The rather crude descriptive primitives used by Barrow and Popplestone in their analysis are similar to our own. (See Chapter 3). It is difficult to find general ways of describing irregular curves that allow one to match the image to stored descriptions. Most methods are either too precise (allowing minor details to hide general similarities) or too general (missing out on essential details). Barrow and Popplestone chose to use gross features such as compactness, circularity, elongation, etc. and rely on adjacency information or context to arrive at a solution. One of the drawbacks of their early system was that it had a fixed set of objects and applied a "best match" algorithm to classify the objects in the scene. This method allows occasional mismatches and of course finds an erroneous match for objects that are not included in its repertoire. The "best match" graph traversing technique employed by this system is typical of the engineering approach of some AI work. While this solution may work efficiently in a limited environment, it

is not suitable for applications in more complicated domains subject to occlusion.

The Waltz labelling technique developed for the blocks world with shadows was successfully extended by Turner [1974] to the universe of regular curved objects (conic section curves) as well as polyhedra. The modelling technique is still based on line labelling at junctions. Naturally, this extended domain requires a much larger catalogue of allowable line and junction labels. Furthermore, Turner points out that the introduction of curves introduces hidden points of transition from one type of labelling to another, i.e. not all transitions happen at easily recognised junction points (see Figure 2.4). Such problems in a world restricted to curves of simple mathematical regularity show the futility of such an approach to more irregular scenes.

Our own work stems directly from some ideas proposed by Guzman [1971] concerning irregularly curved line drawings such as those found in children's colouring books. His approach was to use available context information to disambiguate various shapes and find appropriate models and sub-models. His region description technique suffered from the opposite problem of the crude description technique of Barrow and Popplestone. Guzman proposed a system of encoding the regions shape as a concatenation of the line segments that form its boundary. Such a system makes it very difficult to compare two

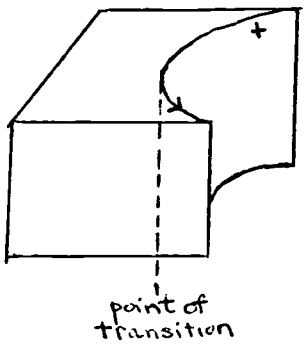


Figure 2.4 Two labels for one line. From Turner 1974

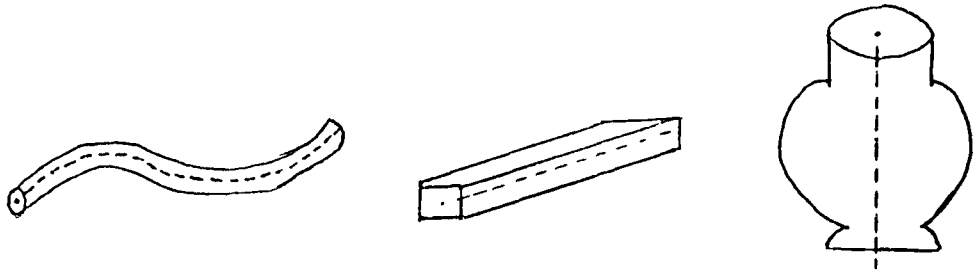


Figure 2.5a Generalised cylinders

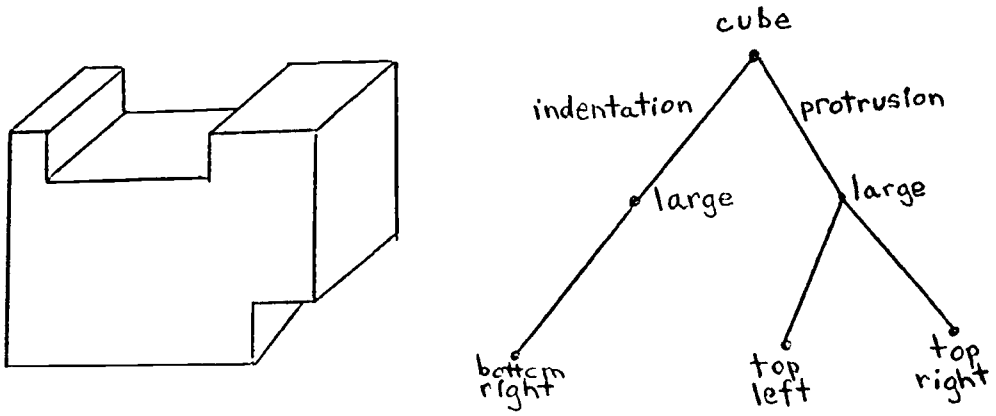


Figure 2.5b Hierarchy of description. From Hollerbach [1975]

regions that are very similar, but differ in only minor detail; the encoding scheme magnifies slight differences. While we have rejected Guzman's region description schema (his shape models), his context information ideas (relation models) strongly influenced our ideas for scene analysis.

2.1.1.3 Modelling with generalised cylinders

The nature of the shape description plays an important role in the recognition process. On the whole, once one leaves the geometrically simple blocks world the actual shape primitives are just that -- primitive, ad hoc, and rather unsatisfactory. Recent work has found the notion of the "generalised cylinder" [Agin 1972; Nevatia and Binford 1977] to be a very useful tool in the description of three-dimensional scenes. A "generalised cylinder" may be described as a cross-section (which may change shape) travelling along an arbitrary space curve which is perpendicular to its plane (Figure 2.5). Hollerbach [1975] has used this technique as the foundation for his hierarchy of shape description. By describing shapes in terms of a simple hierarchy of a fixed repertoire of prototypes, he was able to provide an adequate description of a large family of Greek vases. This domain of curved vases is, of course, well-suited for the use of generalised cylinder prototypes since all Greek vases are radially symmetric. However, Hollerbach has demonstrated the generality of his techniques by also applying them

to polyhedral shapes.

The general approach of the hierarchical description technique is to select the appropriate prototype that most accurately fits the object, and then note the differences between the data and the chosen prototype. Naturally, some differences are more important than others, so the results of the comparison form a hierarchical description. The significance of certain details varies with the prototype, so it is the model itself which contains information concerning matching. This is also true in our system.

The emphasis of Hollerbach's work is on developing useful and qualitative descriptions which emphasise the significant features at the expense of lesser ones which are usually lost in the smoothing process. Much of the success of Hollerbach's technique may be due to the close matching of the descriptive capabilities of the prototype hierarchy and the relatively simple shapes in his chosen domains. He rightly criticises earlier approaches to shape description which are either too sensitive to local features of detail (such as the Blum transform Figure 2.6a, or Guzman's chain coding scheme in Figure 2.6b) or miss them completely by using parameters based on features which are too general such as area, perimeter, or moments of area. Hollerbach's primitives are very general also. To parameterise curvature in Greek vases he uses only five curvature levels varying from strongly curved to very straight. For his domain this is

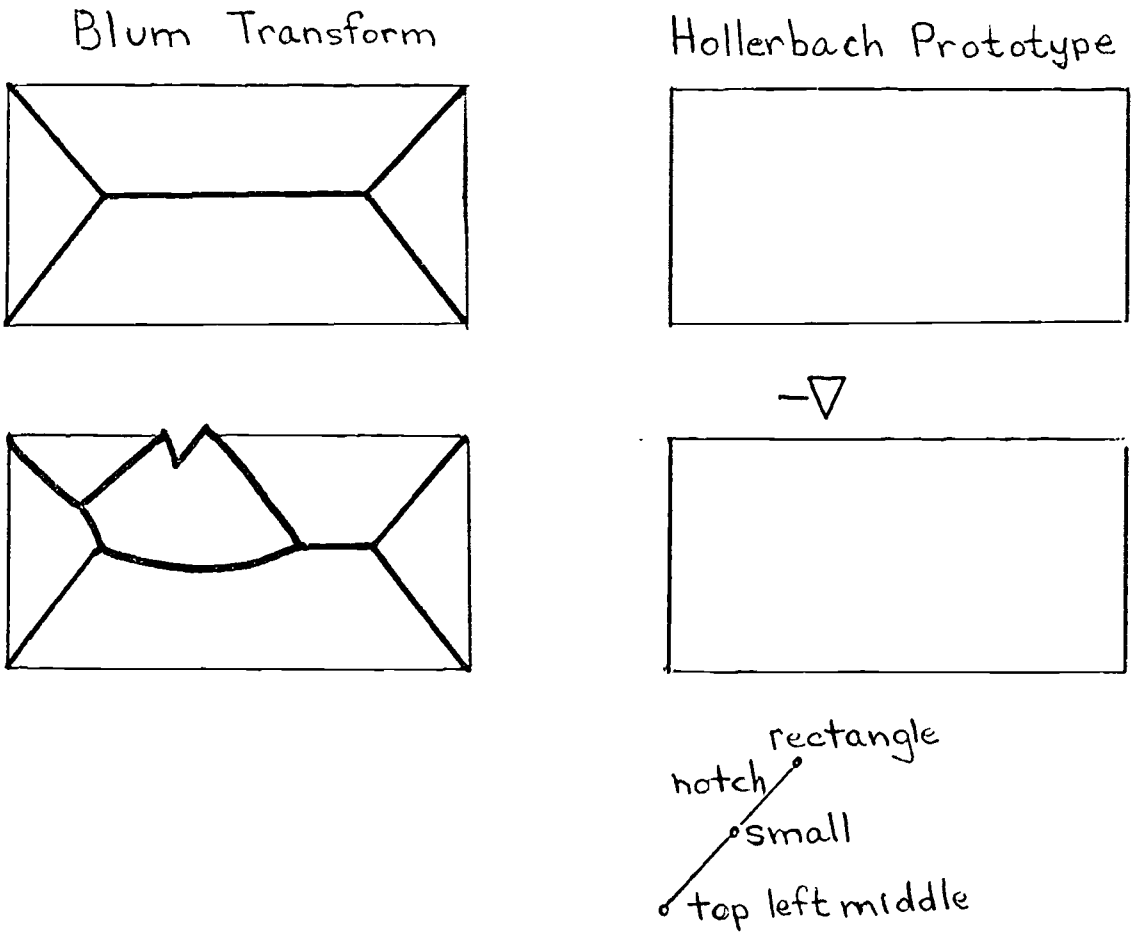


Figure 2.6a Comparison of Blum's Transform technique and Hollerbach's prototype modification

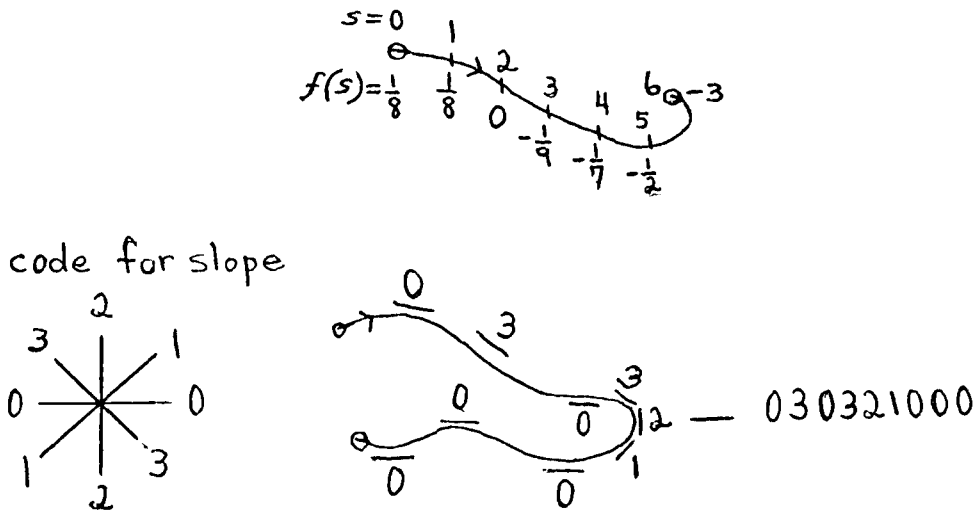


Figure 2.6b Guzman's chain coding scheme
From Guzman [1971]

adequate and he argues that it corresponds closely to archeological descriptions; however, we feel it is a mistake to assume that visual primitives must have counterparts in language.

Although Hollerbach only applied his technique to describe very regular geometric shapes, his overall approach is very promising. The use of prototypes in conjunction with a method of cataloguing differences seems to be a fundamental part of the visual process. Generalised cylinders seem to have wide application as a means of characterising gross three-dimensional shapes or volumes and their orientations. More research is needed to supplement this with information concerning finer details of shape, texture, etc. While Hollerbach's work was primarily a study of perfect line drawings using single cylinders of known axes and orientation, Agin [1972] and Nevatia and Binford [1977] have approached the problem of segmenting a scene into a series of such cylinders, i.e. determining the size, axis, and orientation of generalised cylinders in real scenes (i.e. from TV input). Agin extracts three-dimensional depth information by scanning the scene with a laser, and grouping the lines thus obtained by linking the internal points by a "minimal-maximal distance" method. The axes are determined by the midpoints of the segments; circular cylinder cross-sections are fitted to the axis point estimates and the cylinder is extended as far as possible. His routines work best on objects describable as a single generalised cylinder. Problems of occlusion may cause certain "cylinders" to be missed or merged with neighbouring parts; the cross-section finder

gets confused in the neighbourhood of T-joints (see Section 2.1.1). Nevatia and Binford [1977] have extended Agin's work. Using the same low-level laser ranging techniques, they have refined the segmentation algorithm and they can now handle cases of occlusion where the objects are not in contact. In their terminology "generalised cones" are used to extract the "shape description" of a scene. Their use of the word shape refers to the gross structure of the scene rather than the description of the outline of the object. From this they form a connection graph of the scene which they then match to stored models based on the very primitive descriptors of connectivity, type (long or wide), and whether or not it is conical. Since their objects have such varied stucture, these descriptions are sufficient. Furthermore, the connectivity graph enables the matcher to deal with flexible objects such as the toy dolls that they have used. (See Figure 2.7).

Marr and Nishihara [1975] have also adopted the generalised cylinder representation and have built up a sophisicated theory of spatial modification to account for rotations and other re-arrangements of the standard model in a domain of stick-figure creatures formed from pipe cleaners. In Marr's theory objects are modelled as stick figures (similar to the aforementioned connection graphs) and are organised in a loosely hierarchical manner (see Figure 2.8). To get more detailed information about the object, one climbs down the data-structure. In conjunction with these, they use an "Image-space Processor" to map images into their flexible three dimensional

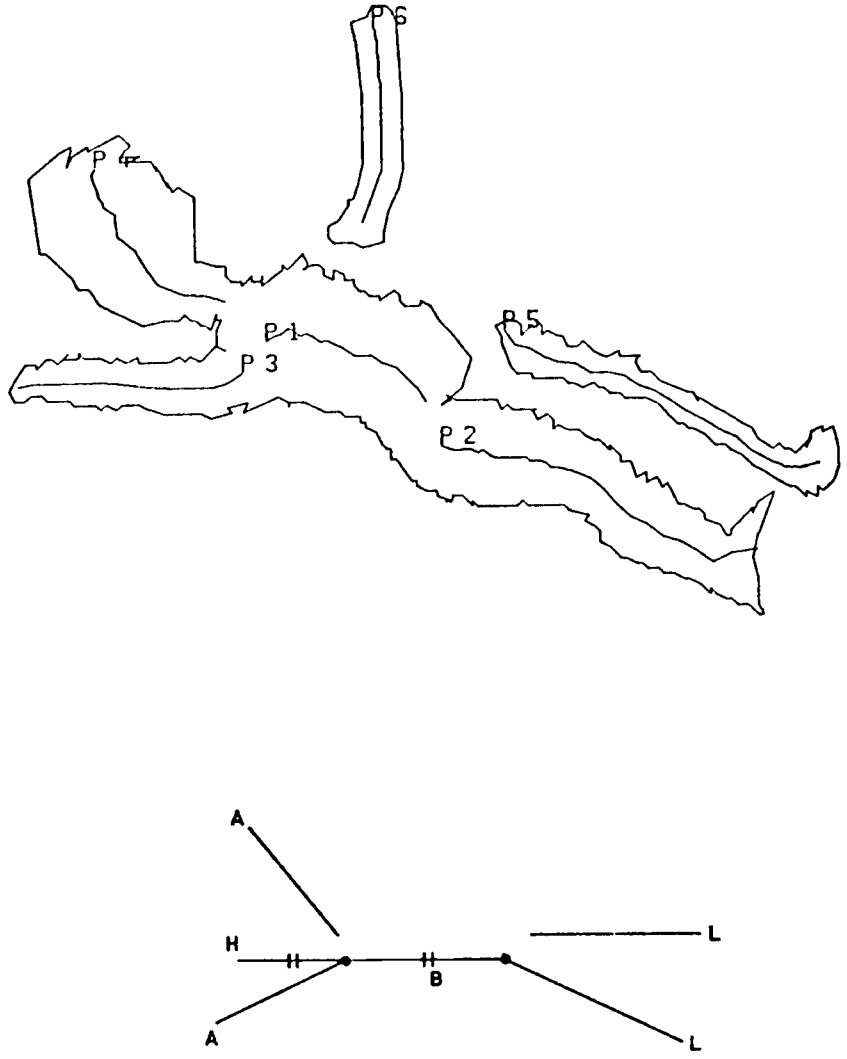
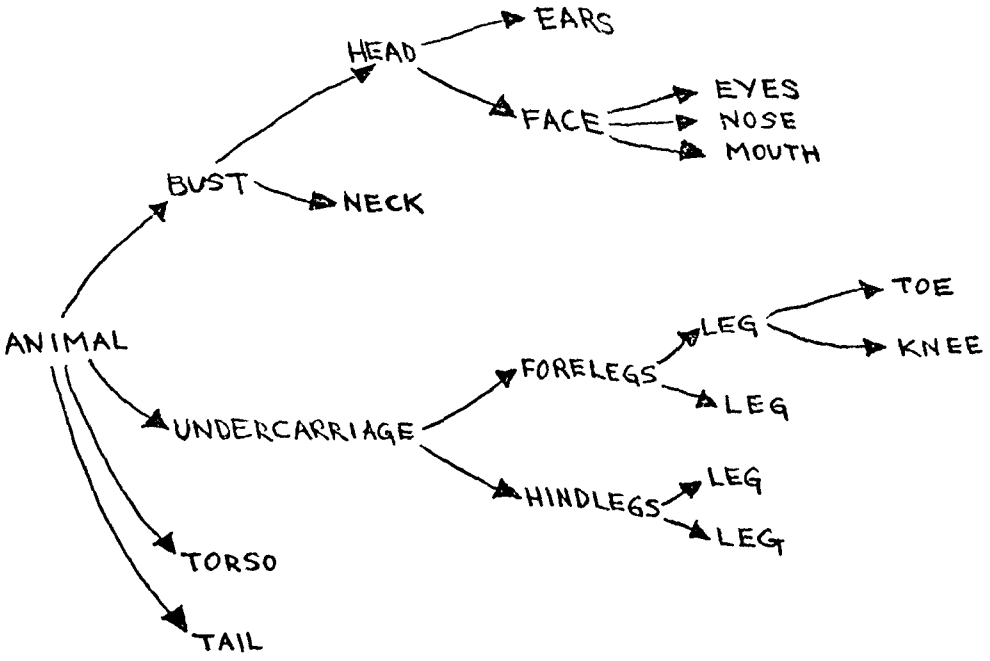
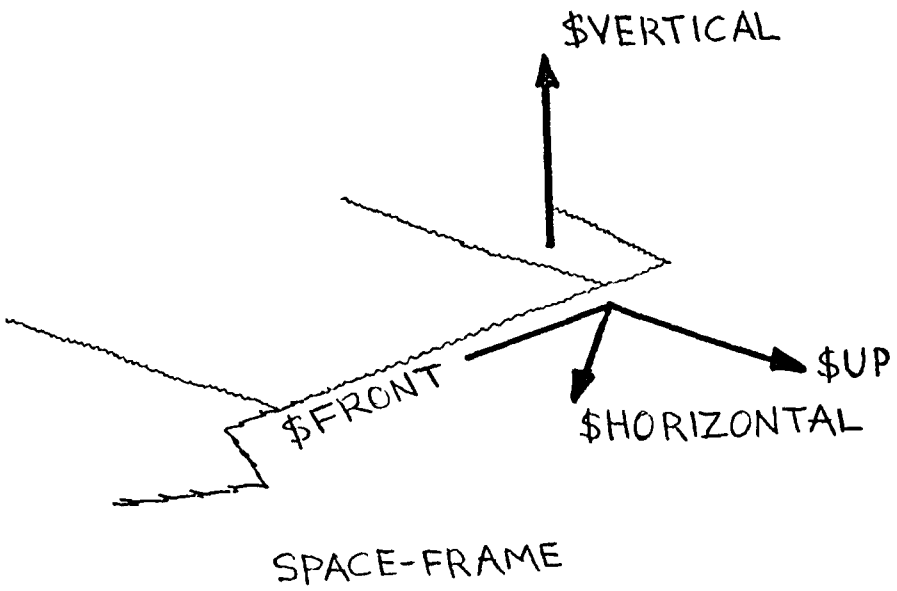


Figure 2.7 A view of a doll and the corresponding connection graph. From Binford and Nevatia [1977]



(a) Hierarchical representation of stick figure horse



(b) Rotation axes of the image-space processor

Figure 2.8 From Marr and Nishihara [1975]

models. The image-space processor is a device for transforming a vector between object-centred and viewer-centred coordinate systems. They claim that the mechanics of their system conform to the psychological data available on human vision -- specifically the mental rotation experiments of Shepard and Metzler [1971].

We should point out that the Marr/Nishihara theory as described is applied to a very simplified universe. Segmenting a two-dimensional image of a real scene (or a "primal sketch" [Marr 1975]) with occlusions into generalised cylinders will be much more difficult than for pipe cleaner animals on contrasting background. Furthermore, the generalised cylinder representation only captures the underlying structure of an object; there are so many other attributes which may be just as important. Nevertheless, Marr and Nishihara make a good case for a sophisticated vision system to at least contain an image-space processor such as they describe.

2.1.2 Occlusion

It is often difficult to recognise an object which is partially obscured by another object. By applying knowledge of the domain and by exploiting redundant information in the scene, it is possible to obtain an interpretation for the scene. By extracting more information from the scene by using better descriptors or providing a richer domain (e.g. adding shadows) one can lessen the effects of

occlusion. Likewise, by incorporating more knowledge of occlusion in the model system (e.g. by cataloguing the junction configurations caused by occlusion) the detrimental effects of occlusion on the recognition process are reduced.

The progression of occlusion handling techniques applied to line drawings of blocks world scenes serve as good examples of these principles. Guzman [1968] used some ad hoc rules based on typical junction configurations which are usually caused by occlusion. Figure 2.9 shows how T-junctions are used to inhibit the linking of regions which are adjacent in the scene but correspond to faces of different polyhedra.

The line labelling techniques of Huffman [1971] and Clowes [1971] were a significant advance over Guzman's approach. The improved performance can be attributed to:

1) Improved use of context information. The requirement of providing a labelling that is consistent throughout the scene ensures that a global interpretation will be achieved. Guzman's interpretations could be misled by local cues.
2) Richer labelling scheme. Guzman's rules only served to link or unlink the regions surrounding line junctions. The Huffman/Clowes labelling offers four possible interpretations for lines (edges): concave, convex, occluding, or occluded. The use of a richer labelling scheme provides more detailed

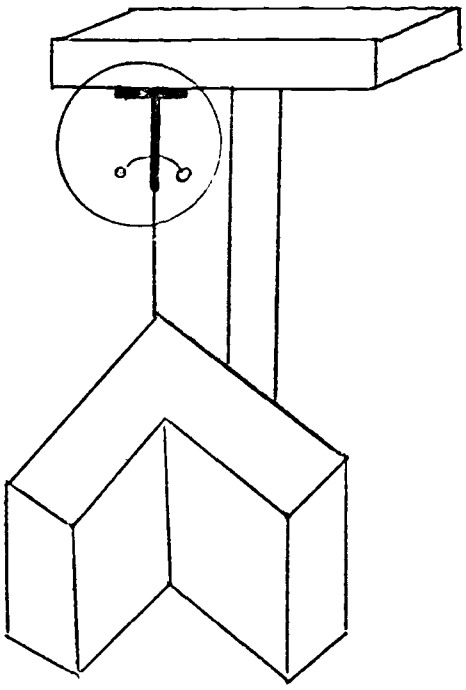


Figure 2.9 T-junction inhibition of links

knowledge of the scene.

3) More complete knowledge of occlusion. The labelling scheme is based on a careful study of all the possible views of polyhedra and the effects of occlusion. In this simple domain, all the types of occlusion can be catalogued.

Waltz's study of scenes of blocks with shadows is an example of enriching the domain to simplify the interpretation procedure. Essentially the shadows offer information about the scene from the point of view of the light source. Occlusion becomes less of a problem because of the additional information in the scene. In order to use this information, the Huffman/Clowes line-labelling algorithm had to be extended to account for lines arising from shadows. The resulting explosion of possible line labellings is adequately controlled by exploiting the stronger constraints provided by the richer domain. (See Section 2.1.3).

These same principles have been used outside the blocks world. The effects of occlusion are more severe when one leaves the geometric domains because it is more difficult to capture all the information in the scene.

Nevatia and Binford [1977] use three-dimensional range information provided by a laser scanning system rather than limiting their input to two-dimensional views. A generalised cylinder representation can

be constructed for objects which are occluded by (but not in contact with) another object.

Baumgart's [1975] approach to three-dimensional geometrical modelling has strong links with computer graphics. He enriches the domain by using several different views of the same object. His system handles hidden surfaces and occlusions -- especially self-occlusions. Polyhedra are used to model the complex three-dimensional shapes obtained from several TV images of the same object. His techniques for handling hidden surfaces are closely modelled on the Warnock [1969] algorithm commonly used in sophisticated computer graphic display programs. The available three-dimensional information is used to decide which surfaces are behind another. These "hidden" surfaces are removed from the display image. While such modelling is quite adequate for display purposes, it is not very suitable for recognition, i.e. matching such a polyhedral model to a stored description. The generated polyhedral images may vary from one trial to another making comparisons difficult. Ohlander's research [1975] on natural scenes handles not only occlusion, but shadows and highlights as well. He incorporates knowledge of occlusion based on local evidence in his system. The region-based analysis is founded on principles of continuity, similarity, and proximity rather than on knowledge of what the region in the scene represents. A case analysis of occlusion possibilities guides the analysis. Regions designated as shadows or highlights are eliminated, occluding regions are removed one by one and remaining region boundaries are extended

to correct for the occlusion. The actual modelling of objects and the recognition process are not described. While this general technique may indeed work in a variety of cases, one misses the interaction with specific models to govern the filling in of missing information. Boundary extension is suitable only for straight lines and simple curves.

Minsky [1975] outlines a method of handling occlusion based on incorporating higher level knowledge in the recognition system. He presents an example of a chair partially hidden by a table and suggests that in a chair-table frame of knowledge such common occurrences of occlusion might be predicted.

Paul [1977] uses such a technique in his domain of puppets. In certain views an arm may be occluded by the puppet's torso. Knowledge of such a typical configuration is used to locate a partially hidden arm or "explain" the absence of a complete occluded one.

As the domains chosen for analysis become more complex, we feel that such higher level knowledge will come into wider use. In our system high-level knowledge is used in conjunction with low-level techniques to tackle the problems presented by occlusion in a scene. (See Chapter 4).

2.1.3 Control Strategies

There are a number of terms that are commonly used to classify the overall strategy of scene analysis. Such terms as bottom-up, top-down, middle-out, parallel, and serial may serve to broadly classify a strategy, but usually a program uses a combination of these ideas.

The choice of domain influences the nature of the models used and these in turn influence the control strategy adopted. In Waltz's analysis of the blocks world (unlike Roberts) there were no isolated models; local descriptions of edges and vertices along with a set of combination rules were used in their place. Instead of concentrating the knowledge in a model of block, the information was dispersed. This structuring of knowledge allowed Waltz to pick up all the information in a bottom-up manner and then interpret it with respect to the whole scene using his filtering technique. Possible labellings for lines at each junction are checked for consistency with possible labellings at neighbouring junctions. Inconsistent labellings are eliminated. This filtering technique effectively provides a parallel interpretation of the data.

The technique of optimization is another method of achieving a degree of parallel processing (e.g. Fishler and Elschlager [1973], Yakimovsky [1973], and Hinton's work on "relaxation" [1976]). In

this approach a large number of structures in the scene are examined simultaneously. Rather than filtering possibilities as Waltz did, an attempt is made to optimize the value of a function. As a result, different labels are assigned to different parts of the image (e.g. points, lines, regions). The optimization technique finds the best overall interpretation of the scene (with respect to the function being evaluated). This technique will yield a labelling even if the scene under analysis is outside the domain under consideration (e.g. Yakimovsky's face recognition program would probably find a face in a picture of the moon).

A contrasting approach is that of Kelly [1970] on recognition of faces. His primarily top-down or model-based approach has much in common with our own work. His models consisted of procedural templates which were applied to an extracted outline of a head. Each feature model had information about gross features by which they might be recognised (i.e. two dark areas for nostrils) as well as their expected position within prescribed limits relative to other attributes. By applying crude tests based on expected locations for facial features he was able to locate first the most easily obtainable features and then with more and more reliability the remaining features. So by finding the eyes, the location of the nose was more accurately predicted. This technique has its drawbacks. By seeking features in a predetermined and contexturally dependent order, one risks complete failure if one of the features is obscured. (Our system like Kelly's is based on a contextually dependent

sequence of scanning. However, we have included the capability of restarting our analysis at another point and coping with problems caused by occlusion or missing model parts. See Chapter 5.)

Tenenbaum [1973] in his work on analysis of office scenes also uses goal-directed feature-extraction based on local context-dependent attributes. As with most real-world scenes the descriptions used are rather crude and ad hoc. However, Tenenbaum claims that an exhaustive shape description is not necessary, contextual relations and multi-sensory data (range, colour, texture) should simplify the recognition process. Supplied with sufficient knowledge of the current context, Tenenbaum proposes that a program should be able to use problem-solving techniques to locate a telephone in an office. Shirai [1975] describes a system (also working with office scenes) that recognises complex objects based on edge information extracted from TV pictures. The models are two-dimensional and represent "typical views". While this may be sufficient for some classes of scenes, it too seems very context-dependent. Vertical lines may unambiguously represent books in scene of an office in California, but in Edinburgh they may just as well represent a radiator. This illustrates the close link between models and analysis techniques.

As AI vision has progressed, so has the role of the model. Originally a model contained shape information about objects in the world that could be paired with objects in the image domain. Now

more sophisticated models are used which contain not only shape information, but also relational information or context information. They are not only used for matching purposes, but also have the capability of determining the search strategy or even guiding the analysis. While single isolated models may have been sufficient for simple or very restricted worlds, there is a need to have several models (possibly related) to describe more complex objects or even for different views of the same object. Minsky [1975] has proposed a system of "frames" to handle complicated environments. He describes a frame as a bundle of knowledge about some familiar object or grouping of objects. This knowledge has default values that may be modified upon inspection of the environment. A frame may also be altered by the attachment of other sub-frames, e.g. by joining a "statue frame" to a "horse frame" we may keep the essential horse appearance information, while cancelling all animate characteristics and modifying size and colour parameters.

The models or frames influence the analysis of the scene, and the analysis of the scene may alter the frame or cause new frames to be invoked. This sophisticated type of interaction takes place in our own analysis and our model hierarchy is similar in many respects to the proposals of Minsky. There are differences, of course. In particular, we do not allow the attachment of one model system to another; such sophistication did not prove necessary in our environment.

Minsky's ideas have not been implemented and are not sufficiently detailed on many points, but they have influenced many AI researchers in both vision and natural language. There has not yet been a vision program to fully explore all his notions. Freuder's system [1976] was influenced by some of Minsky's ideas. Freuder makes use of what he terms "active knowledge" to guide the analysis of a scene. If the handle of a hammer is located, the system gives advice regarding the location of the hammer head. The particular knowledge of the handle location is combined with the general knowledge about hammers. Although he makes great claims for his heterarchical system (based on some of Minsky's ideas), it is difficult to evaluate the versatility of his program since it is only applied to scenes of a single hammer.

Kuipers [1975] has outlined how Minsky's frame ideas may be applied to the problem of recognition. He presents six steps that are needed in such a working system:

(1) Representing the hypothesis
(2) Manipulating the hypothesis
(3) Selecting the next observation
(4) Evaluating the observation
(5) Selecting a new hypothesis
(6) Translating knowledge to the new hypothesis.

Our system has portions corresponding to each of these steps. The most important point is the strong interactions between models and control structures. We believe this is essential as a general mechanism for machine vision.

2.2 Related Psychological Theories

2.2.1 Theories of Perception

Although our model system was not designed as a strict implementation of psychological hypotheses of visual perception, it was influenced by some theories on cognitive psychology. In the words of fellow researcher John Knapman, we are "informed but not constrained" by these theories.

A typical AI approach to visual perception may be described as a process of constructive hypothesis testing. As we try to interpret images as objects we are guided by expectations or internal models. These models are invoked by certain fragments of a scene, and thereafter the interpretation of the image is strongly influenced by this model. Clowes [1973] refers to this process as "seeing-as".

As far as low-level vision is concerned, research by Hubel and Wiesel [1965] with the visual cortex of cats has shown that there are mechanisms for the signalling of complex information on the retina directly to single cells in the cortex. The stimulus features used were stripes of various orientations. There is no evidence yet that there are receptive fields for curved lines or even simple shapes like squares or circles. It is likely that such figures would be signalled by combinations of cells, or by some higher level process.

Recent work by Marr [1975] suggests that processing of visual information can be done at a much lower level than most previous AI programs have done. Marr performs a uniform low-level analysis of an image. His results (called a primal sketch) are in the form of symbolic assertions about changes in intensity that occur in the image. The assertions are of two types: about the presence of edges and bars. Basically, an edge indicates the boundary between two regions of differing intensity while a bar is a region with no change in net intensity (e.g. a line). See Figure 2.10. Although Marr's stimuli are restricted to single static objects, a primal sketch may be obtained for any scene. McLennan [1975] is using primal sketch data in her study of scenes of house plants.

There still remains the problem of what to do with all these assertions at the next stage in grouping. Although there is a great deal of low-level information available, there is very little structure. Low-level grouping rules are inadequate to segment the scene so higher level models are necessary. It is this higher level knowledge that has been of interest to us -- the function of the brain, not the eye. Gregory [1974] states:

> "if the brain were not continually on the lookout for objects, the cartoonist would have a hard time. But in fact, all he has to do is present a few lines to the eye and we see a face. The few lines are all that is required for the eye -- the brain does the rest: seeking objects and finding them whenever possible."

The need for higher level knowledge is exhibited in the

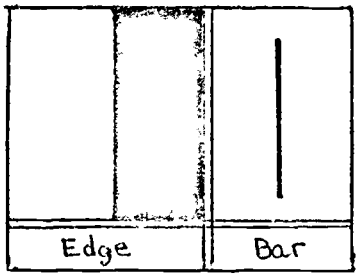


Figure 2.10 Kinds of contrast changes

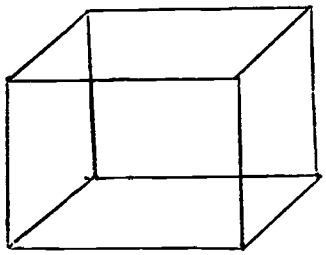


Figure 2.11 Necker cube

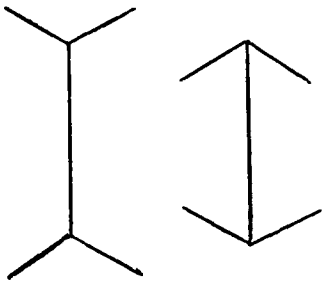


Figure 2.12 The Müller-Lyer illusion

interpretation of ambiguous scenes. The same retinal image may be seen as two or more different objects. For example, see Figure 2.11, the Necker cube.

Gregory's work on inappropriate constancy scaling serves as an illustration of how higher level knowledge (or models) may influence the interpretation of an image. Gregory uses this theory to explain many perspective illusions. See Figure 2.12. The perceptual system compensates for changes in the retinal image with viewing distance and viewing angle. Smaller images may be seen as more distant; ellipses are usually interpreted as circles. The perceptual illusions arise when cues in the image trigger the scaling compensation mechanism inappropriately.

Such perceptual models may vary from culture to culture. Gregory contrasts people in Western urban cultures with the rural Zulus. While we in the West interpret perspective cues in such situations as rectangular rooms, and the long parallel lines of railways and roads, the Zulus live in a "circular culture" and experience few straight lines or corners. The Zulus are not as affected by the same illusionary figures as Western people. Their models emphasise different features. Deregowski [1972] has observed that most rural Africans have great difficulty perceiving or interpreting depth in two-dimensional drawings although the cues we find familiar, such as size and linear perspective are available. Although the objects are

recognised the spatial relationships could not be determined.

These examples show that the interpretation of the patterns on the retina depend on much more than pure data. We "see-as" objects -- not patterns. While patterns have little if any meaning on their own, objects have a wealth of descriptors not necessarily limited to visual characteristics. By seeing abstract objects as instances of more familiar concrete objects, we can impose our knowledge of past events -- our memory -- to make predictions.

> "We do not perceive the world merely from the sensory information available at any given time, but rather we use this information to test hypotheses of what lies before us." [Gregory, 1973].

Such a hypothesis-and-test schema may seem inappropriate in most visual tasks where there is sufficient redundancy of data to ensure a correct interpretation (a conclusion rather than an hypothesis). However, in data-deprived situations such a scheme appears to be the best approach, and this is the basis for our program's analysis of its visual data.

2.2.2 Theories of Shape Recognition

There is very little psychological evidence concerning the description of shapes. Several models of form perception have been considered by psychologists (see Haber and Hershenson [1973]). The template model cannot adequately deal with invariances of size and position, nor can it cope with segmentation of the image. Clearly a

more interactive process is required. Eleanor Gibson [1969] proposed feature models as a method of recognition. The visual form is broken down into features. As we increase our ability to discriminate shapes, these feature lists become longer. A third alternative is the constructive model, or a feature model with relationships included. In this model, the configuration evokes a schema which is the "data-base" for the operations which determine the percept. If a particular pattern of segmentation is inconsistent with the model, it is rejected and an alternative interpretation is made.

Appropriate segmentation and the use of relationships to construct a meaningful image have been shown to play an important role in remembering an image. Haber and Hershenson [1973] report on a pair of studies concerning picture memory by Wiseman and Neisser, and Freedman and Haber. The experiments investigated the need for the perceiver to impose a coherent organisation on the image data. In these two studies, the subjects were presented with pictures of faces which were to be remembered. The faces were drawn with very high contrast -- only the shadows and highlights were apparent (see Figure 2.13). In some scenes it was more difficult to see a face than in others. In all cases, sometimes a face could be seen and sometimes it could not. The subjects were shown a selection of such scenes and asked if they could see a face in it. Later they were tested for recognition with a mixed set of "new" and "old" pictures (all of faces). If the perceiver saw an organised face, both at the first presentation and at the time of testing for recognition, then there

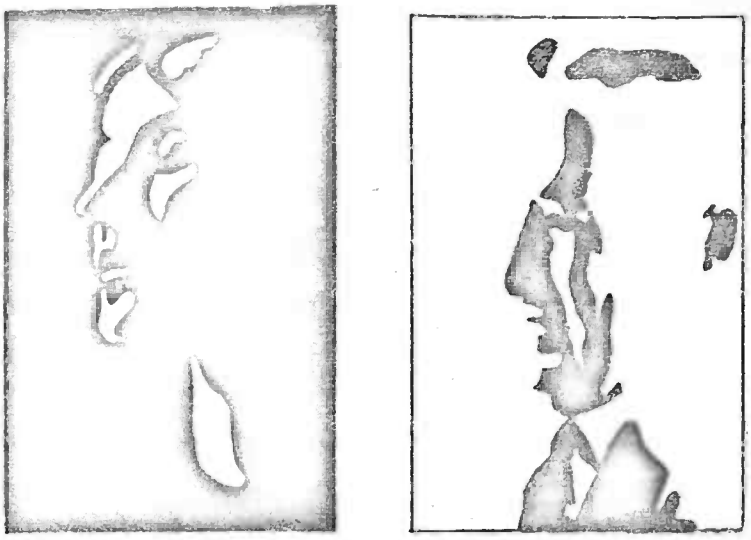


Figure 2.13 Two faces shown in highlight.
From Haber and Hershenson [1973]

was a very good chance that he would remember it. It was argued that without the organised structure, the perceiver would have to remember a large number of meaningless blobs. This illustrates the fact that the retinal image is not the only factor in scene recognition. The stimulus must be integrated with an internal model. Our program can be seen in a very similar light. Just as the shapes in the highlighted figures are difficult to describe, so are our region shapes. However, in the context of the appropriate model, the scene has meaning. The shape of the individual regions has been de-emphasised. The relationships of the composite parts with respect to the model is of much greater significance.

Our model can be seen as a combination of the visual feature model and construction model described above. The relational information of our model hierarchy corresponds to the construction model; the shape information is little more than the simple feature model described above.

The next chapter describes our model system in detail.

CHAPTER 3

MODELS

3.1 What is a Model?

As we have shown in Chapter 2, models can take many different forms and be used in different ways and at different levels. They may be simple templates to be matched to the image, or as in Waltz's case, the model information may be dispersed throughout the system - used as a set of constraints to eliminate impossible interpretations. For the purpose of this paper, we shall restrict the term model to mean a hierarchical organisation of shape and relational information which can be used to interpret the scene based on the selection of appropriate sub-model possibilities.

3.2 How the Models are used

In most cases it is the model that controls the analysis of the scene. If the appropriate object model is invoked, it can step through the scene region by region, guided by information about how the regions interconnect to form meaningful images. In this way, a full description of the scene can be constructed, matching each region in the scene to each part of the corresponding model (barring occlusion or other complications).

Shape information guides the local analysis, matching regions to shape descriptions in conjunction with relational information that guides the global analysis finding the next region for examination. In the very irregular world of cartoons, we were unable to find a suitable general method of describing the varied shapes. Instead, we specify a series of shape feature tests for each object that must be recognised by the system. Each test contributes to a score for the acceptance or rejection of the model-to-region match.

In this way only the essential features of each particular object need be specified. Within the hierarchy features are not used to generate model possibilities in a bottom up manner. Instead, each model uses its own tests to confirm whether or not it matches a specified region. The shape description tests are based on purely local information; without more global knowledge or context information this interpretation method of applying simple tests would fail because a region is often open to several different interpretations when studied in isolation. The number and type of tests varies with the complexity of the shape. A SOCK is easily tested with:

	Value	Score	Tolerance
HEIGHT	250 units	15	50%
WIDTH	600 units	15	20%
AREA	14 units*	30	50%
SHAPE	RECTANGULAR	40	--

*The units of area are 10,000 square linear units

where the height of a standard PEANUTS character is taken to be 6500 units. Each test contributes to a score for the acceptance or rejection of the model-to-region match. A more complex shape like LUCY's HAIR might be tested with:

	Value	Score	Tolerance
HEIGHT	1350 units	15	10%
WIDTH	3000 units	15	10%
AREA	230 units	20	10%
LOBES (lower)	4	15	25%
LOBES (upper)	2	15	
SURFACE-MARKINGS	0	20	

The purpose of these tests is not to decide if the region in question matches the model; rather, it is to decide if there is sufficient evidence to accept a candidate region. The expectation of what the region should be, provided by the relation model, allows us to accept a result on rather flimsy evidence.

Such a method of analysis has both advantages and disadvantages. The obvious advantage is that there are fewer checks or tests required before assigning (at least temporarily) an identity to a region. Naturally, this saves processing time: it is a short cut. The human visual system might take a similar short cut, that is recognise an object on the basis of a fleeting impression and an expectation rather than performing a detailed analysis. Only if an inconsistency

is discovered is the detailed analysis necessary to correct a misperception. The disadvantage is that if the wrong model has been chosen the analysis may proceed much further than is desirable before any inconsistency is noticed due to the high tolerances of the system. At such points knowledge of model similarities may be useful to allow partial results to locate the correct model.

3.3 The Model Hierarchy

As previously noted, our original conception of using separate models for local (shape) and global (relational) descriptions was based on Guzman's ideas [1971]. He proposed a separation of these two classes of description: a set of models for shapes and another set for the spatial relationships between them. While his scheme may have worked fairly well for static objects such as hats, blocks, and houses, for flexible objects the complexity of the description becomes too great. Flexible joints, self-occlusion and the variety of possible positions suggest a more complex set of models is needed. The discussion which follows will centre on the PERSON model hierarchy, since it is both the most frequently used model as well as being the most complex one. There are also Structure models for baseball caps, bats, gloves, etc.

Within the model, there is no distinction between what an individual is wearing and his actual body. Since the analysis is region-dependent, the system would be puzzled by a naked body because

it would only find one region instead of the distinct regions representing different articles of clothing. (PEANUTS characters are rarely seen without clothing.) The essential cues for recognition would be missing. Alternative models could of course be added to deal with such special cases, but the current PERSON model could not be modified to work in such a case. The model depends on the clothing to segment the PERSON into functional units such as head, arms, legs, etc. The recognition of the whole depends on the ability to locate and recognise these separate body parts, and thus build up a total description of the scene in terms of corresponding sub-models. In most PEANUTS scenes SNOOPY is portrayed as having almost human characteristics and assumes human postures. Since his body is not segmented, our models are not applicable. The models could easily recognise the character as SNOOPY since his head is a separate region. But it could not provide a description of the positions of his body parts. SNOOPY is not included in our subset of the PEANUTS world.

There are a variety of different characters that must be identified. The complete model for PERSON must be capable of analysing a general PERSON as well as recognising the details that characterise each individual. These differences between characters of the same sex in the PEANUTS world are for the most part restricted to hair styles and head shapes. The shapes of heads vary in quite subtle ways and have proven too difficult to distinguish with the coarse shape tests; however, the hair styles are easily distinguishable by using tests

for characteristic features and these are the best evidence for character identification. In PEANUTS cartoons the characterisation on the basis of hair style corresponds to identification by facial features in the real world. We may use other cues to recognise people (e.g. clothes, height, weight), but faces usually provide the best evidence. In cartoons, hair style is the key to recognition.

Since each object may appear in one of several orientations in a particular scene, a model must have some reference to orientation in the three-dimensional world. In geometric worlds, different orientations could be described by altering the equations to reflect different perspectives. We use a series of 'frames' [Minsky 1975] each representing a different view to achieve the corresponding understanding in the 'two-and-a-half dimensional' space of the cartoon world. See Figure 3.1. Different views of an object have their corresponding sub-models or frames. The complete model system is composed of the set of all these possible frames for different views.

In many ways this frame technique is well-suited to the cartoon universe. The number and scope of the different frames is limited by the simplicity of the PEANUTS environment. The cartoonist only uses about six views of a head selected from the vast number of possible head orientations. Standard body poses are used to convey a variety of body positions. There is another benefit from the crudeness of the shape descriptions. Since our descriptions are not very precise,

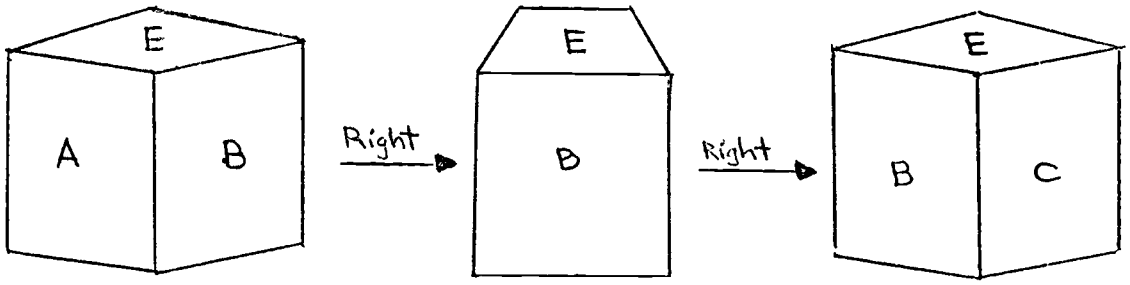


Figure 3.1 Different frames for different views

they are not necessarily specific to one particular recognition task. The same description may apply over a small range of similar body configurations. For instance, all the arms in Figure 3.2a can be recognised by the same procedure. The chef's hat seen in Figure 3.2b will also pass this test. This emphasises the important role of relational information to ensure that the descriptive tests are only applied when and where they are needed. A final degree of complexity must be introduced because of the flexible nature of the characters' bodies. It is this flexibility that can cause self-occlusion and can alter the arrangement and shape of the various body components. Again the frame idea is applied here to cope with this problem (along with the occlusion routines). More than one frame may be supplied for a specific view such as LEFT to reflect the variety of possible region configurations. The flexibility of the bodies is also handled by allowing a slight mis-match between sub-frames, e.g. a HEAD facing FRONT-RIGHT may match a TORSO facing either FRONT, RIGHT, or FRONT-RIGHT. To summarise, the model for a person must handle:

(1) A variety of characters

(2) A variety of orientations

(3) A variety of body poses

(4) Some "actions" (e.g. WALKING, STANDING)

Each frame represents a possible combination of these variations, and any frame decision affects both the local and global aspects of the analysis, i.e. both the region shapes and their connectivity. With this overview in mind, we are now ready to describe the system of models to handle the complexity of the problems described above.

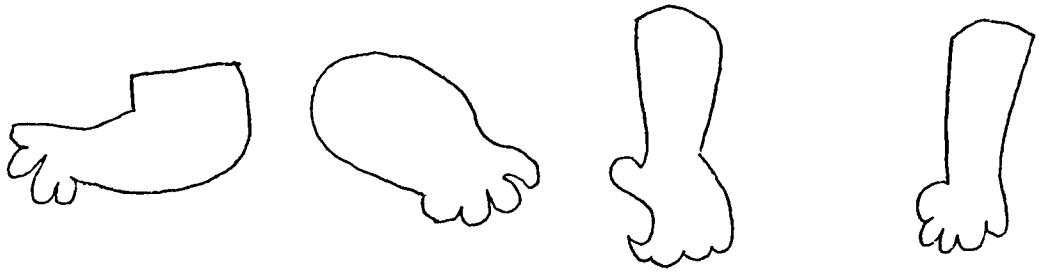


Figure 3.2a

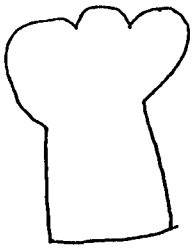


Figure 3.2b

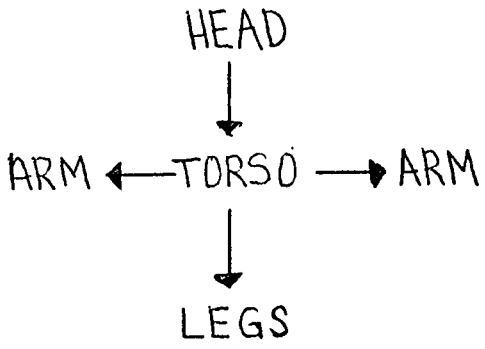


Figure 3.3 Structure model

The models are divided into four classes which form a hierarchy of relational and shape information models:

(1) Structure
(2) Component
(3) Description
(4) Composition

Of these four, the first two classes of models are primarily used to generate relational information, while the last two are for shape information. The control structure that guides the analysis uses this hierarchy to effectively blend the descriptive and relational functions so one cannot describe the effects of one class as either purely descriptive or purely relational. For this discussion, we shall again use the most complicated PERSON model for illustrative purposes. Simpler objects may contain dummy models in parts of the hierarchy.

3.3.1 The Structure Model

The Structure model represents the overall appearance of the object in terms of the possible body parts (see Figure 3.3). Each body part is composed of a region or group of regions that change their shape and interrelationships as a unit. Examples of such units are HEAD, TORSO, ARM, etc. By dealing with these functional body units as separate sub-models (see Component models, Section 3.3.2) the recognition task can be simplified. The Structure model concept is derived from Guzman's [1971] relational model. The major and important difference is that the components of the model were carefully chosen to represent functional body units. There may be up to eleven regions that combine to form the LEGS node of this

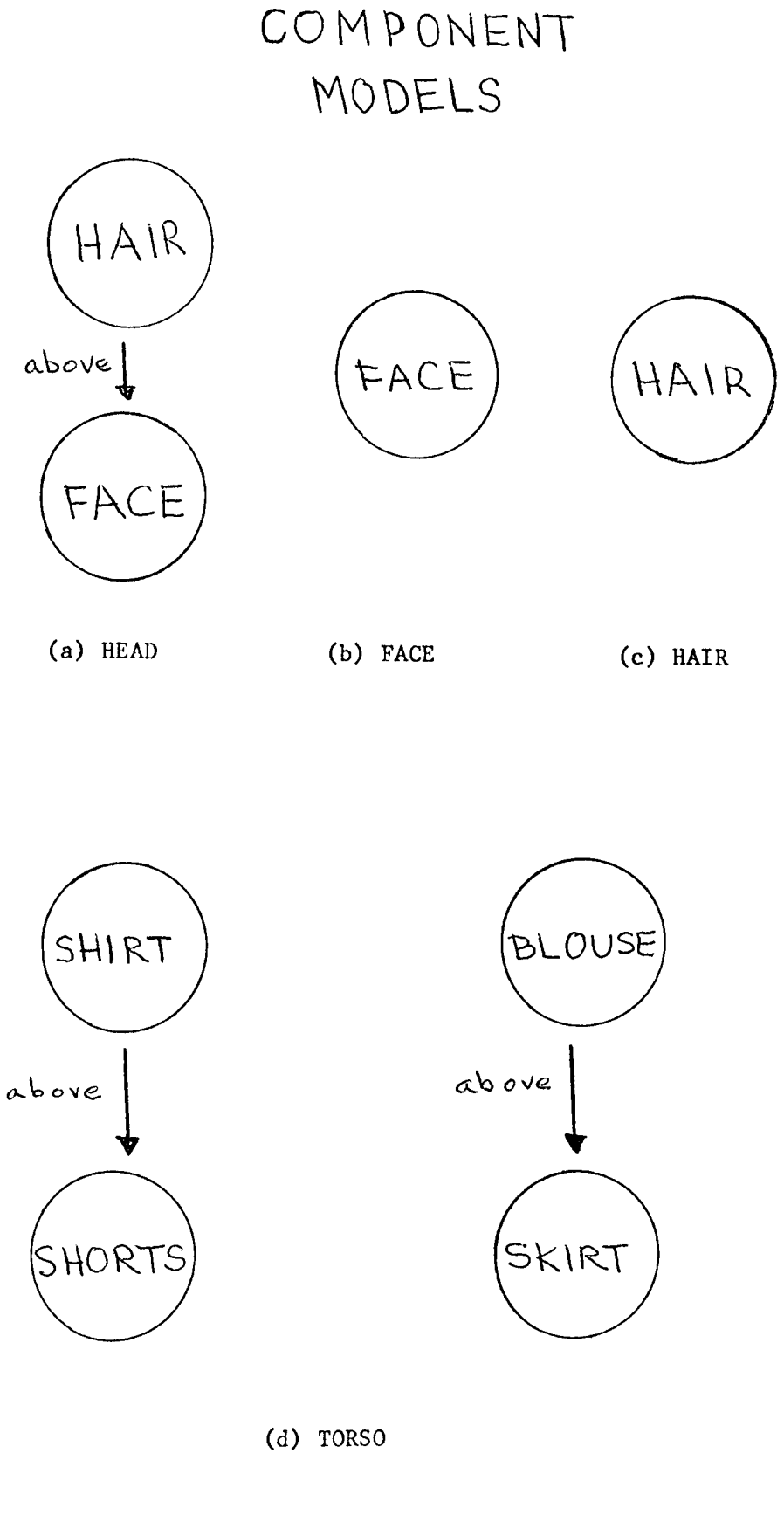
Structure model, while an ARM may be composed of only one. By structuring the information in the hierarchy we can defer the recognition of these units to separate recognition experts. Notice, too, that while ARMS are treated separately, LEGS are not. This is because the cartoonist uses the legs together to convey the action. In this cartoon world, the physical possibilities are limited. One cannot be STANDING on one leg, while SITTING or RUNNING on the other.

While the simple model depicted in Figure 3.3 proved sufficient to handle some scenes, complex issues such as occlusion, distortion, and missing model parts required a more sophisticated approach. The naive model merely reflected the relationships between various body parts and suggested search directions for the location of neighbouring body units. The more sophisticated model modifies its suggestions to reflect the state of the current analysis and knowledge of possible body positions. For instance, if the TORSO was found to be facing toward the RIGHT, then the expected position for the ARMS is modified accordingly. Furthermore, if there is an interruption to the normal processing flow, the Structure model can offer suggestions for re-starting the analysis. These details will be left for later chapters; the simple model will suffice for the present discussion.

3.3.2 The Component Model

Like the Structure model, the Component model contains relational information. For each of the terminal elements of the Structure model there is at least one Component model to describe how the individual regions fit together to form the functional body unit. For example, a HEAD may be represented by Figure 3.4a. This is the most common representation of a PEANUTS head, but there are other possibilities. Charlie Brown has no hair so his HEAD is represented by the simple Component model shown in Figure 3.4b. Similarly, the back of the head is represented as in Figure 3.4c. The purpose of having more than one Component model is to have separate recognition procedures for the different possible configurations of the cartoon bodies. The procedures are general enough to cover several body positions due to a very tolerant interpretation of directional information.

One can see that although the Component model illustrated above appears to display purely relational information this is not strictly true. Since there are several Component model possibilities, the selection of one particular model has a bearing on the shape information. A good example of this is the Component model for TORSO (Figure 3.4d). It should be apparent that SHORTS will have a different shape than a SKIRT. The TORSO model not only gives the relationships between the nodes of SKIRT and SHORTS, or BLOUSE and SKIRT; it also determines the shape information by its selection of appropriate sub-models.



(a) HEAD (b) FACE (c) HAIR

(d) TORSO

Figure 3.4

As the positive results of the descriptive tests flow back up the hierarchy to the Component model, they change the context of future selections of descriptions. The discovery that a character's SHIRT is facing toward the right indicates that his SHORTS will be facing right also. Such context effects are not limited to the Component model; they are in turn passed to the Structure model when the Component model has successfully completed its analysis.

To summarise, the main purpose of the Component model is to break down the functional groups of regions into their component region parts so that the shapes of the regions in the scene can be tested. The variety of possible alternative representations for the functional body units is handled by a collection of alternative Component models.

3.3.3 The Description Model

The Description model does nothing more than perform the shape tests required to determine if an individual region is an instance of the required shape. Usually the Description models are called by the Component models to examine the regions which fit together to form the functional body units. (But see the Composition models, Section 3.3.4.) The weighted sum of the shape test results is used as a measure of confidence of the model-to-region match. If the resulting sum is below a specified threshold, the match is rejected. (This is not strictly true. Suspicion of occlusion will invoke the occlusion

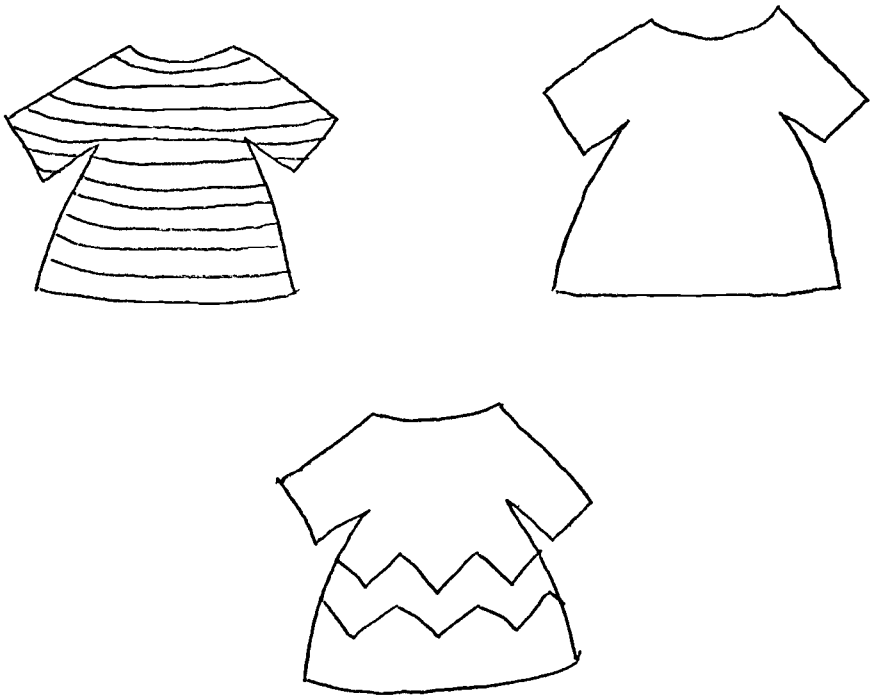
routines. This is discussed in Chapter 4. For the purpose of this chapter we will not confuse the issue with occlusion handling at this point.) The rejection of one Description model causes another possible model to be invoked. The process continues until either the appropriate model has been found, or all possible models have been rejected. The latter possibility usually indicates that the wrong Component model has been invoked. If this occurs, the Component model is rejected and an alternative Component model is applied. (Again this is not strictly true. See Chapter 6 for details about missing model parts.)

A successful Description model-to-region match may help to limit the future searches through the possible alternatives in the hierarchy. Each selection must be consistent with past results. By establishing a character's identity (i.e. matching a HAIR Description model to the appropriate region) one can eliminate all the models which are not appropriate for that person.

3.3.4 The Composition Model

At the lowest level of the hierarchy a description is matched to a single region. However, due to the variety of possible representations for classes of objects, some objects may be shown as a single region in one scene, but composed of multiple regions in another. Figure 3.5 illustrates this problem.

(a)



(b)

Figure 3.5

Since our analysis is region-based, the sub-division of a shoe into three separate regions or the diyision of a shirt into regions representing stripes confuses the shape analysis. Our solution has been to add another layer of relational information to the hierarchy: the Composition model. (See Figure 3.8). If an object may be represented by more than one region in a scene it is described by a Composition model containing the relational information which explains how the regions fit together. There are also Description models for each of these regions.

The LEGS Component model accesses both types of SHOE model without regard to the detailed region gathering analysis that might have to be performed. In typical cases, the descriptive tests are very crude. To recognise the saddle-oxford SHOE of Figure 3.5a only size is of any importance. The shape varies extensively from scene to scene. The relationships between the three regions is the critical factor.

In practice, only one layer of Composition models was necessary for the PEANUTS scenes analysed. By using several such layers one may incorporate more knowledge of structural organisation in the model. More complicated domains may require several such layers.

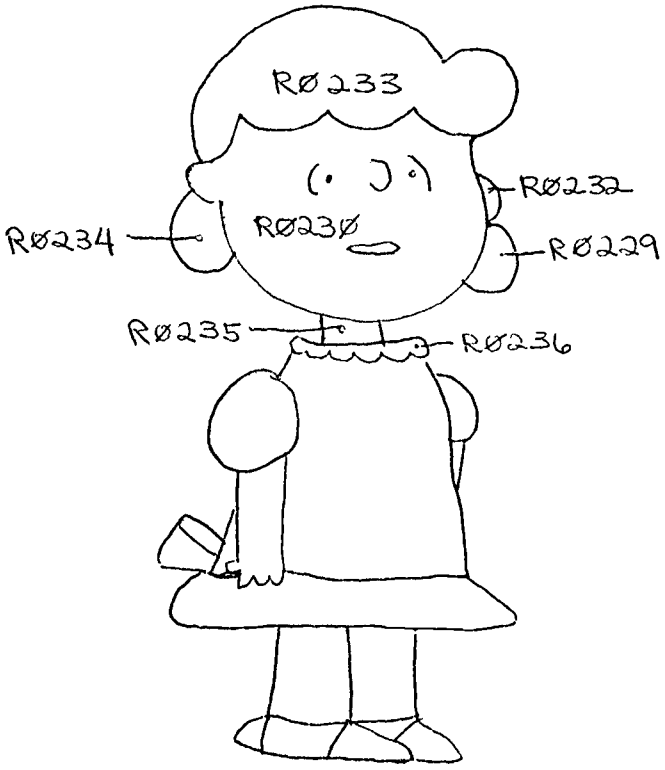


Figure 3.6

3.4 The Model Hierarchy in Action -- An Example

With only a discussion of the hierarchy itself, and not a discussion of the occlusion heuristics and control mechanisms, it is not possible to go into a completely detailed commentary of the system's analysis. In this section we will stress the role of the model hierarchy and ignore the other issues. More complicated scenes will be examined in later chapters.

The following analysis refers to Figure 3.6.

We begin this annotated analysis at R0233 at the top of the picture with the analysis being driven by the PERSON model. Refer to Chapter 5 for control details leading to this point.

Analysis of HEAD: The Structure model for PERSON indicates that HEADS are usually located at the top of the body (a head-stand would be an exception). With this in mind, the Structure model turns control of the analysis over to the HEAD Component models with a call which for our present purposes we will describe as:

```
(FETCH
     '(COMPONENT-MODEL     ;Get all Component models
     HEAD                  ;for HEADS.
     R0233                 ;Match them to R0233
     !?who                 ;Try all characters' HEADS since
                           ;we do not know who it is yet.
     !?view                ;Try all HEAD views
                           ;since we do not know what
                           ;direction they are facing
     TOP                   ;This is a suggestion by the
                           ;TOP-LEVEL of the PERSON
                           ;Structure model
     !>score)              ;If a match is found, return a
                           ;confidence score.
```

Each Component model has a header slot which may match this call. At this early point in the analysis, all the HEAD Component models are eligible. They are gathered into a list of possibilities and tried in succession. This list is ordered so that the most likely alternatives are tried first.

The first Component model tried is the most common, the basic HAIR above FACE model which is appropriate here. Just as the Structure model passed control to the Component model to deal with the HEAD analysis, the HEAD Component model calls on the HAIR and FACE Description models to act as experts on the detailed analysis of the regions concerned:

```
(FETCH
     '(DESCRIPTION-MODEL   ;Get all the Description models
     HAIR                  ;for HAIR.
     R0233                 ;Still working from region R0233
     !?who <restricted>    ;Still do not know who this is
     !?view <restricted>   ;nor do we know the view
     HEAD                  ;suggested by HEAD Component model
     !>score))             ;request for confidence score
```

Again, the sub-models are gathered into a list and tried one by one. The "<restricted>" label on the unbound parameters who and view indicate that although we do not know which Description model is appropriate, we can eliminate some possibilities. As we mentioned above, there are various Component models for HEADS. A back view of a head consists of only a view of the HAIR, no FACE can be seen. So the scenes are restricted to non-back views. In a similar manner, this Component model is not valid for CHARLIE-BROWN (he has no hair region), so possibilities for who have also been restricted. These excluded possibilities are represented by other Component models.

Finally, we reach the stage of the actual region-to-shape comparison. Various HAIR Description models are invoked, one by one, until one which matches the region is found. The system includes models for Charlie Brown, Lucy and Violet each facing in several directions.

Let us examine the winning model. Its header slot looks like:

```
(DESCRIPTION-MODEL    ;This is a Description model for
HAIR                  ;HAIR
!>region              ;It takes a given region
LUCY                  ;to match to a shape description
                      ;for LUCY
FRONT-RIGHT           ;facing the front-right of the screen
!>suggestor           ;as suggested by the given
                      ;model and will
!<score)              ;return a confidence score.
```

In this instance region is set to R0233; and suggestor, HEAD. The standard simple tests that are applied in this model are:

	Value	Score
maximum extension in X direction:	3000	10
maximum extension in Y direction:	1350	10
area:	230	20
no surface-markings		10

(see discussion of FACE)

The values mentioned above are allowed to vary within a range of fifteen percent. The critical tests for LUCY's hair pertain to the lobed parts of the region. The presence of these features is the strongest confirming evidence of a proper description-to-region match; therefore a disproportionate contribution to the total confidence score comes from the lobes, both at the top and the bottom of the region. By convention, the PEANUTS characters can never look directly to the front of the screen, their noses and shoes point either to the right or left. LUCY's hair varies with direction also. The small lobe on the top of her hair is in the direction she is facing, the slightly larger lobe on the bottom of the region is on the opposite side (see Figure 3.7). These particular tests then are shown below:

	Value	Score
LOBES (below)	3-4	20
LOBES (above)	2	20
(large and small)		
Threshold:		70

The total score then for this series of tests is 90 out of a 100. The threshold represents the minimal score necessary to accept the

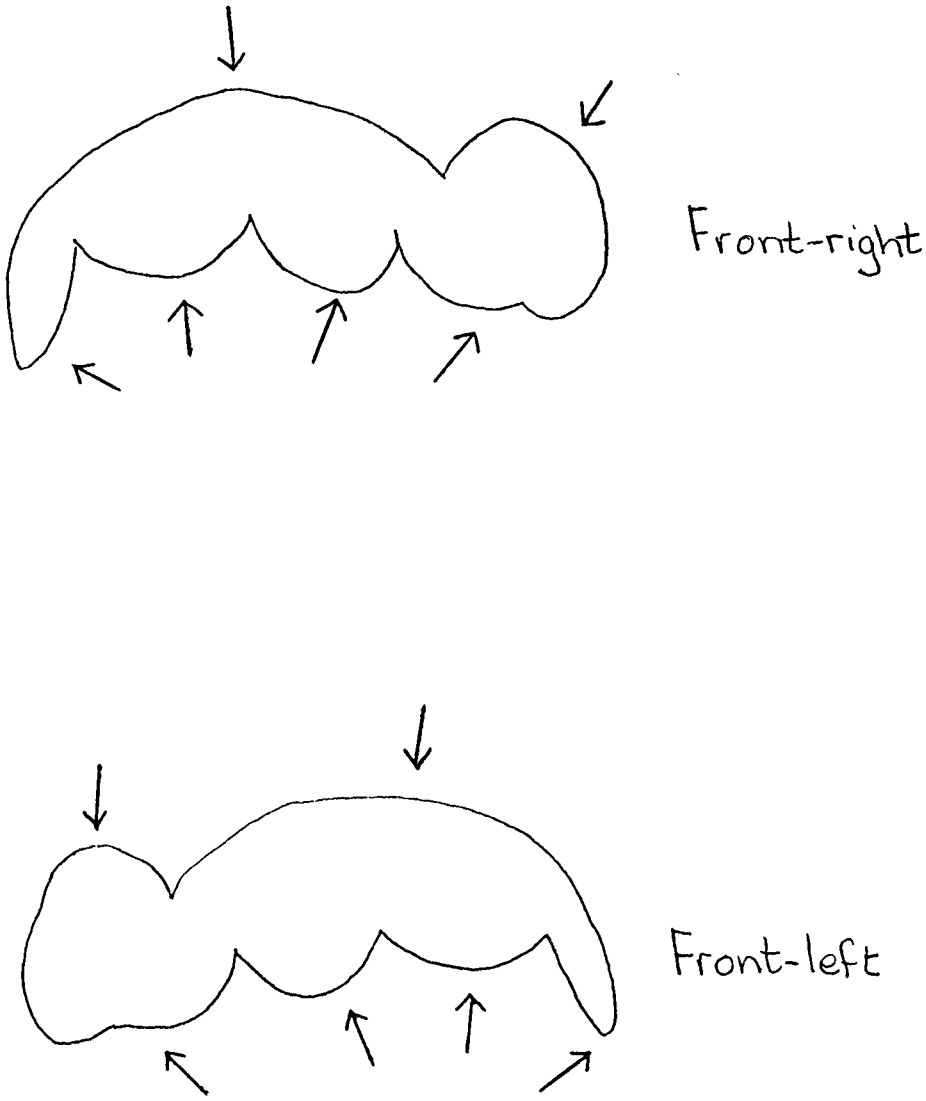


Figure 3.7

match. The missing 10 points will be contributed by the two remaining regions that also represent LUCY's hair, regions R0234 and R0229 (see below). The success of the Description model is signalled by the addition of the filled-in header to the data-base:

(DESCRIPTION-MODEL HAIR R0233 LUCY FRONT-RIGHT HEAD 90)

Auxilliary Description models needed to deal with the remaining hair regions are added to the data-base and the Component model is informed. In a true three-dimensional analysis these remaining hair regions might be dealt with as parts of the main hair region separated by occlusion. While the T-junction heuristics we developed could determine that R0234 and R0229 are occluded, the best approach in this two-dimensional world is to incorporate the information into the model system (see below).

Control returns once again to the HEAD Component model. The next step is to identify the FACE region in its expected place below the HAIR. The relation models provide directional information in terms of above, below, left, and right. For the PEANUTS world these have proved sufficient. If several regions meet the direction relationship, each in turn is matched to the allowable models. This time the call for appropriate Description models may be made more specific by the information provided by the successful HAIR model:

```
(FETCH
 '(DESCRIPTION-MODEL      ;Again find a Description model
        FACE              ;for FACE this time
        R0230             ;for the region below R0233
        LUCY              ;for the character LUCY
        FRONT-RIGHT       ;facing front-right
        HEAD              ;as suggested by the HEAD
        !>score))         ;and return a score.
```

This time only one model will match this request since all the variables (except the score) have been found. This particular face is built up out of three separate regions:

R0230	Main face
R0232	Ear
R0235	Neck

In a side view of the FACE all these parts are represented in a single region; in this particular view we need three. To keep a consistent structure to the model system over all the possible configurations, we make use of the Composition model at this point. Instead of a Description model, a Composition model is returned. This model returns the required Description models for the main face region:

```
(FETCH  '(DESCRIPTION-MODEL      ;Model type
        FACE-PART                ;for main face
        R0230                    ;region to match
        LUCY                     ;character
        FRONT-RIGHT              ;view
        FACE                     ;suggested by FACE
        !>score))                ;confidence score slot.
```

The facial features (eyes, nose, mouth) are the essential features to confirm the model-to-region match, although size must also be considered. These features are not coded as regions but surface-markings (see Chapter 5) contained by the bounding region. The tests then for the matching Composition model are:

	Value	Score
X extension	3000	15
Y extension	1900	10
area	385	25
Surface markings:		
eyes		20
nose		20
mouth		10
Threshold:		65

The imbalance in the score values reflects our personal evaluation of the relative contribution of each test to the overall recognition process. We have no set rules for selecting these figures. The successful Description model returns:

(DESCRIPTION-MODEL FACE-PART R0230 LUCY FRONT-RIGHT FACE 100)

The FACE Composition model may now locate the EAR and NECK regions with respect to the face yielding:

		Value	Score
EAR:	X	220	20
	Y	320	20
	area	3	30
no surface markings			30
	Threshold:		60

(DESCRIPTION-MODEL EAR R0232 LUCY FRONT-RIGHT FACE 100)

The tests here are quite simple and the allowable variation is greater than normal. Almost anything in the general area with approximately the right size will be seen as an EAR. It is the relational information which forces the interpretation. Similarly for the final Description model:

		Value	Score
NECK:	X	565	20
	Y	275	20
	area	17	25
no surface markings			20
	rectangular		15
	Threshold:		60

(DESCRIPTION-MODEL NECK R0235 LUCY FRONT-RIGHT FACE 100)

The total for the Description model is a weighted average of its three Description sub-models.

((4* FACE-PART + EAR + NECK) / 6)

The three regions are merged into super-region S0300 for convenience. The result then from the FACE Composition model is returned to the HEAD Component model:

(COMPOSITION-MODEL FACE S0300 LUCY FRONT-RIGHT HEAD 100)

As a final step, the Component model searches for the two remaining HAIR regions. (This task is signalled by the presence of the auxilliary Description models).

These are located on either side of the FACE region S0300. Again it is size and location rather than particular shapes that are tested. The data-base entry for HAIR is revised as a super-region of sorts; S0301 is formed of R0233, R0234, and R0229, and the new entry is:

(DESCRIPTION-MODEL HAIR S0301 LUCY FRONT-RIGHT HEAD 100)

N.B. This super-region is tagged as a list of non-contiguous regions, rather than the normal merger of connected regions that we typically call a "super-region".

These results are combined in the Component model which returns its result:

(COMPONENT-MODEL HEAD S0302 LUCY FRONT-RIGHT TOP-LEVEL 100)

and control passes back to the Structure model.

A similar procedure must now be followed to analyse the TORSO which is a more complicated structure, having eight regions in this particular instance. Occlusion processing is required within this model, but we will defer details of this until the next Chapter.

The Structure model "points" to the region below the HEAD, R0236 and calls for the TORSO Component models:

(COMPONENT-MODEL TORSO R0236 LUCY FRONT-RIGHT HEAD !>score)

There are only two Component models for this view: the male and the female. The identity of this character has already been determined, so the female TORSO is applied. In turn the call to the BLOUSE Description model is made:

```
(FETCH (DESCRIPTION-MODEL
            BLOUSE R0236 LUCY FRONT-RIGHT TORSO !>score))
```

which in turn calls the necessary Description model. All these layers of the model may seem unnecessary at this point in the discussion, especially for the TORSO. This part of the body generally offers us no new information concerning the scene; it merely connects the interesting portions of the body: the head, arms and legs. The value of the hierarchy is seen to its best advantage when filtering through possible options rather than merely calling up pre-determined sub-models. Nevertheless, we will quickly follow the analysis of the TORSO by the appropriate sub-models in the hierarchy outlining both the descriptive tests and the relational information that is used. The Composition model for blouse (FRONT) calls four Description models: collar, shell, sleeve-left, and sleeve-right, with an optional possibility for the bow regions. The structure of their relationships is illustrated in Figure 3.8. The tests for the Description models are provided below:

	Test	Value	Result
COLLAR:	X	1100	20
	Y	190	20
	area	13	30
	lobes	4-5	30
		Threshold:	60
SHELL:	X	1800	20
	Y	2000	20
	area	280	30
	surface-markings	2-3	30
		Threshold:	60
SLEEVE:	X	800	20
	Y	800	20
	area	46	30
	no surface-markings		30
		Threshold:	60
BOW-MAJOR:	X	550	20
(larger part)	Y	480	20
	area	14	30
	no surface-markings		30
		Threshold:	60
BOW-MINOR:	X	245	20
(inside part)	Y	450	20
	area	4	30
	no surface-markings		30
		Threshold:	60

As we have mentioned, the occlusion details are described in the next Chapter. A few points are worthy of discussion at this stage, however, because the problems caused by occlusion affect the model hierarchy's recognition task. There are two classifications of occlusion problems:

1) Unpredicted occlusion - handled by T-junction pairing heuristics

2) Predicted occlusion - handled by model selection and variation.

In most cases, either method of solution may be used. Some common

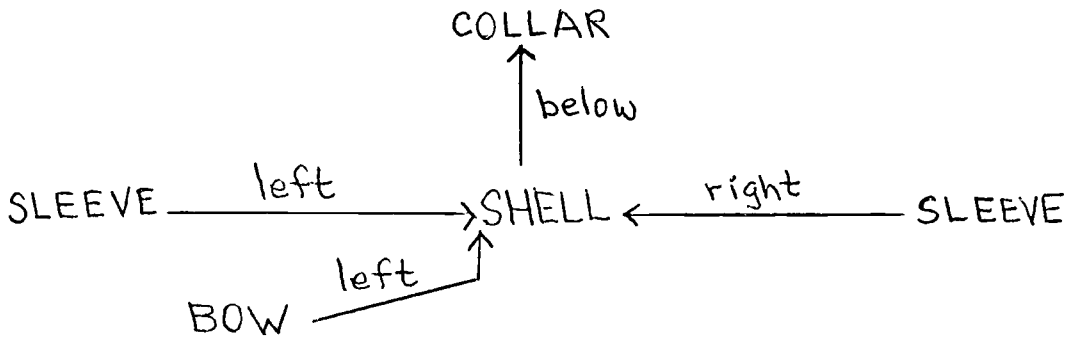


Figure 3.8 Composition model

forms of occlusion that often involve complicated analysis have been "compiled" into the model to make the processing easier. In the two following Chapters on Occlusion and Control, we will discuss this issue in more detail. For the current analysis we will merely indicate where occlusion handling was necessary and what the results of the analysis indicated.

We now resume the analysis within the BLOUSE Composition model at the examination of the COLLAR by the appropriate Description model. (Refer to Figure 3.9). There are no problems here and the model succeeds:

(DESCRIPTION-MODEL COLLAR R0236 LUCY FRONT-RIGHT BLOUSE 100)

Below the collar the Composition model expects to find the main body of the blouse designated the SHELL, but this is an occluded SHELL. Nevertheless, it is easily recognised; only the area parameter is below the specification of the Description model. This discrepancy does not bring us below the acceptance threshold, so the occlusion processing (in this case) is postponed. The following results are returned by the model:

(DESCRIPTION-MODEL SHELL R0239 LUCY FRONT-RIGHT BLOUSE 70)

The analysis continues with the right sleeve (easily recognised):

(DESCRIPTION-MODEL SLEEVE R0240 LUCY RIGHT-ARM BLOUSE 100)

The left sleeve is occluded. It fails both area and width test sufficiently to call in the occlusion heuristics which easily determine that the region in question, R0237, is occluded by R0239.

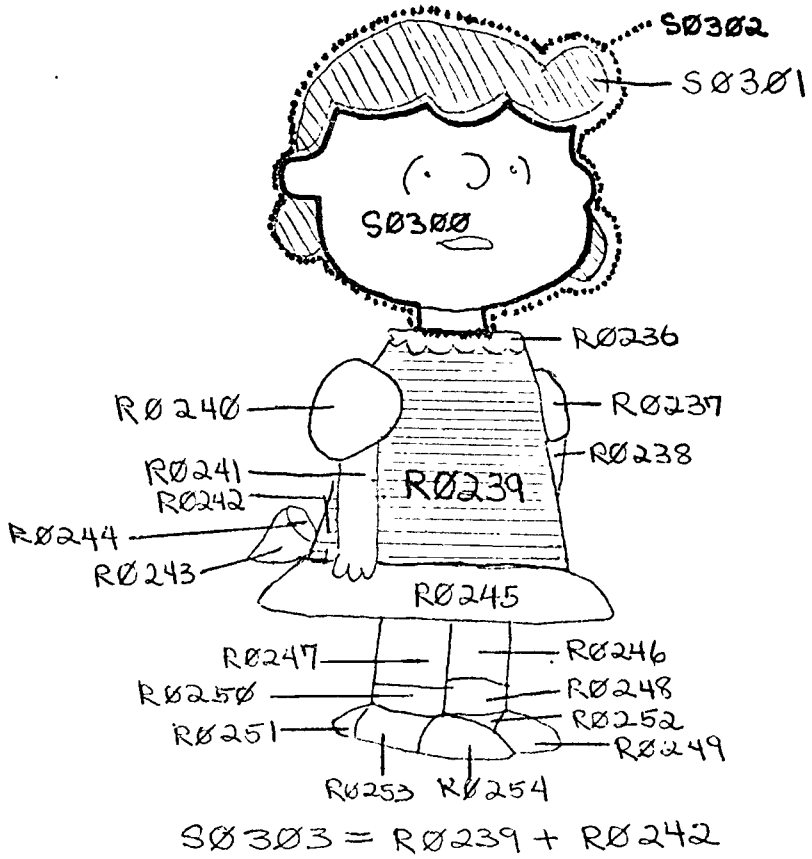


Figure 3.9 Refer to text for a complete list of region identities, pages 92-93

The acceptance threshold is lowered and the match is allowed:

(DESCRIPTION-MODEL SLEEVE R0237 LUCY LEFT-ARM BLOUSE 50)

Next, the bow regions are sought, and for the first time in the analysis, the wrong region is selected for the match. The arm region R0241 is selected instead of the correct one R0243. This is done on the basis of region adjacency only. It is the shape tests that determine not only if the proper model has been selected for matching, but also if the appropriate region has been chosen. Since the region is too large occlusion cannot be the immediate cause of the region-to-model match failure. We must resort to other means of solving this dilemma, which we refer to as the "missing model part" problem.

In other words, the BOW is yet to be found in the scene so the BLOUSE Composition model cannot complete its analysis. We have experimented with various techniques to deal with such problems. One method is shown here. (This common self-occlusion problem is handled with the Component model in later versions in an analogous manner to the LEGS analysis below.)

The problem is despatched to the Troubleshooter (see Chapter 5) which uses special case knowledge to handle the problem based on the available information from the scene, namely:

1) The "view" of the character, i.e. FRONT-RIGHT;

2) The name of the missing model part: BOW;

3) The region R0241 (that was in the expected place for the bow) and its size/shape characteristics;

4) The partial information obtained to this point, i.e. the results of successful model analysis.

The Structure model provides further useful information. In this case, the ARM is connected to the SLEEVE. This solves part of the present difficulty. R0241 is matched to the ARM Component model and subsequently the ARM Description model confirms the match:

(DESCRIPTION-MODEL ARM R0241 RIGHT VERTICAL TORSO 100)

There is still the problem of the bow and the occluded SHELL region R0242. Based on knowledge of common character configurations, the solution is now at hand: the vertical right arm in a character facing toward the right may cut off part of the SHELL, in this case R0242. Super-region S0303 is formed and the score value for the shell is increased:

(DESCRIPTION-MODEL SHELL S0303 LUCY FRONT-RIGHT TROUBLE-SHOOTER 80)

Now the analysis of the bow can proceed. It is in the expected position to the left of the SHELL:

(DESCRIPTION-MODEL BOW-MAJOR R0243 LUCY FRONT-RIGHT BLOUSE 100)

(DESCRIPTION-MODEL BOW-MINOR R0244 LUCY FRONT-RIGHT BLOUSE 100)

The BLOUSE Composition model returns control to the TORSO model.

(COMPOSITION-MODEL BLOUSE S0304 LUCY FRONT-RIGHT TORSO 95)

The final TORSO step involves the skirt. Although there is slight occlusion by the ARM, this goes unnoticed. The region is recognised with:

		Value	Score
SKIRT:	X	3100	20
	Y	680	20
	area	140	30
no surface-markings			30
	Threshold:		70

(DESCRIPTION-MODEL SKIRT R0245 LUCY FRONT-RIGHT TORSO 80)

So the TORSO Component model is complete, the results are gathered together:

(COMPONENT-MODEL TORSO S0305 LUCY FRONT-RIGHT HEAD 97)

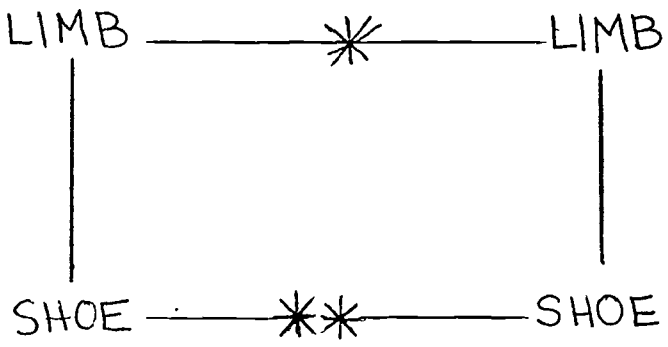
The right arm has already been located, so the Structure model need only find the left arm. The occlusion routines come again to the rescue and R0238, although severely occluded, is found as the missing arm.

(DESCRIPTION-MODEL ARM R0238 LEFT VERTICAL TORSO 20)

The final task of the Structure model is to initiate the call to the LEGS Component model:

(FETCH '(COMPONENT-MODEL LEGS R0247 !>type FRONT-RIGHT TORSO !>score))

The LEG Component model breaks down the structure of the legs as illustrated in Figure 3.10. The order of the analysis plays an important part in the analysis. By handling the nearer unoccluded SHOE, the processing is simplified. Various sub-models are available to account for the various possible leg configurations which are used



✳ A region representing the background may appear here

✳✳ Often one SHOE will occlude the other

Figure 3.10

extensively by the cartoonist to convey motion and body positions. (Further examples of leg analysis follow this one).

```
(FETCH '(COMPOSITION LIMB R0247 !?type FRONT-RIGHT LEGS !.>score))
```

The LIMB breaks down into two regions called the CALF and SOCK. These are easily found. The shape tests for these rectangular regions are the now familiar X, Y and area. The SHOE for female characters is always formed of three regions. For the standard STANDING position, the HEEL portion of the far foot is usually hidden. This fact has been incorporated into the model instead of using occlusion routines.

The following are the results of the model matches:

```
(DESCRIPTION-MODEL CALF R0247 STANDING FRONT-RIGHT LIMB 100)
(DESCRIPTION-MODEL SOCK R0250 STANDING FRONT-RIGHT LIMB 100)
(COMPOSITION-MODEL LIMB S0306 STANDING FRONT-RIGHT SKIRT 100)

(DESCRIPTION-MODEL MID-SHOE R0253 SADDLE RIGHT SHOE 100)
(DESCRIPTION-MODEL TOE R0254 SADDLE RIGHT SHOE 100)
(DESCRIPTION-MODEL HEEL R0251 SADDLE RIGHT SHOE 100)
(COMPOSITION-MODEL SHOE S0307 SADDLE RIGHT LIMB 100)

(DESCRIPTION-MODEL CALF R0246 STANDING FRONT-RIGHT LIMB 100)
(DESCRIPTION-MODEL SOCK R0248 STANDING FRONT-RIGHT LIMB 100)
(COMPOSITION-MODEL LIMB S0308 STANDING FRONT-RIGHT SKIRT 100)
```

(DESCRIPTION-MODEL MID-SHOE R0252 SADDLE LEFT SHOE 50)

(DESCRIPTION-MODEL TOE R0249 SADDLE LEFT SHOE 100)

(COMPOSITION-MODEL SHOE S0309 SADDLE LEFT LIMB 75)

The second MID-SHOE is marked as occluded. The missing HEEL is accounted for by the model.

(COMPONENT-MODEL LEGS S0310 STANDING FRONT-RIGHT SKIRT 96)

The analysis is complete. The top-level information obtained can be summarised as:

The character is LUCY (determined by HAIR - consistent with TORSO and SHOES) She is STANDING (determined by LEGS) facing the FRONT-RIGHT of the screen (determined by HAIR consistent with TORSO, ARMS and LEGS).

All regions are accounted for; a region by region accounting is shown below.

R0229	Part of LUCY's HAIR
R0230*	Main part of FACE
R0232	EAR
R0233	Main part of HAIR
R0234	Part of LUCY's HAIR
R0235	NECK
R0236	COLLAR
R0237	SLEEVE - occluded
R0238	ARM - occluded
R0239	Main SHELL of BLOUSE
R0240	SLEEVE
R0241	ARM
R0242	Part of BLOUSE SHELL
R0243	Outside of BOW
R0244	Inside of BOW
R0245	SKIRT

---

* 00231 refers to the outer closure of the regions, i.e. the silhouette of LUCY in this case.

R0246	CALF (part of LEG)	
R0247	CALF	
R0248	SOCK	
R0249	TOE of SHOE	
R0250	SOCK	
R0251	HEEL of SHOE	
R0252	Middle part of SHOE	
R0253	Middle part of SHOE	
R0254	TOE of SHOE	
S0300	FACE	[R0230,R0232, R0235]
S0301	HAIR	[R0229, R0233, R0234]
S0302	HEAD	[S0300, S0301]
S0303	SHELL	[R0239, R0242]
S0304	BLOUSE	[R0236, R0237, R0240, R0242, R0244, S0303]
S0305	TORSO	[R0245, S0304]
S0306	LIMB	[R0247, R0250]
S0307	SHOE	[R0251, R0253, R0254]
S0308	LIMB	[R0246, R0248]
S0309	SHOE	[R0249, R0252]
S0310	LEGS	[S0306, S0307, S0308, S0309]

The analysis of the LEGS in the previous scene was straight-forward, and we hurried through the analysis. We use two other examples of legs to clarify the issues involved, paying more attention to the relational aspects this time.

First let us examine Figure 3.11. Assuming R0198 has just been analysed we have three choices for regions "below" it to initiate the LEGS Component model: R0193, R0194, and R0196. Guided by previous information that the character is facing FRONT-RIGHT, we would choose to do the left-most leg first, since it is most likely to be unoccluded, R0194 then is proposed as the starting point for the right LEG and the analysis proceeds through the hierarchy to label that region as the CALF. The Description model for LIMB searches for a region below the CALF, and R0208 is matched to SOCK. There are two

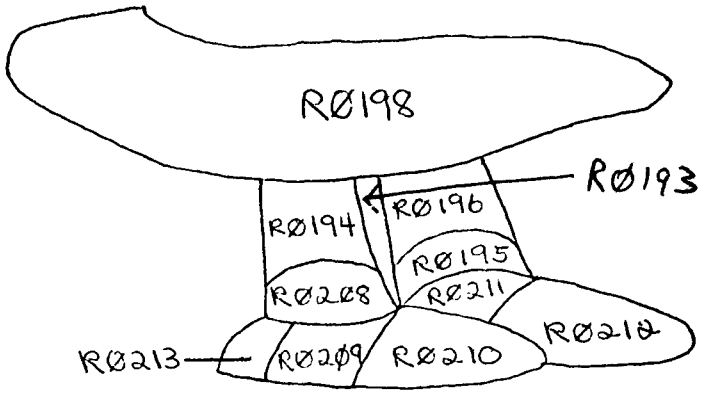

RØ198
RØ193
RØ194
RØ196
RØ195
RØ208
RØ211
RØ212
RØ213
RØ209
RØ210

Figure 3.11

regions below R0208. One of them will be passed as the starting point to SHOE. If the wrong one is selected, the SHOE model will fail and the alternative region will be selected. The SHOE Description model accepts regions R0209, R0210, and R0213 and control returns to LEGS with half of the structure accounted for. There are two possible starting points for the second leg: R0193 and R0196. The model tries R0193 (the left-most one) first, this fails the test, but R0196 succeeds and the other leg is analysed as before. In this scene, R0193 represents the background showing through the LEGS. There is an optional slot in the LEGS Component model for this common case. Any region between the two legs is labelled background but is not considered as part of the LEGS although it is labelled within that Component model.

Finally, consider the different configuration depicted in Figure 3.12. Such a configuration is labelled as WALKING legs rather than STANDING. They also indicate that the character is facing towards the LEFT. The STANDING model we have shown fails on this region configuration, as we see below.

As before, the near leg is tackled first since it has a better chance of being unoccluded. Region R0316 passes the threshold of acceptance as the CALF part of the leg, but R0318 is too large to be interpreted as a SOCK, the LIMB Description model fails, as does the STANDING LEGS Component model which called it. The correct WALKING LEGS model

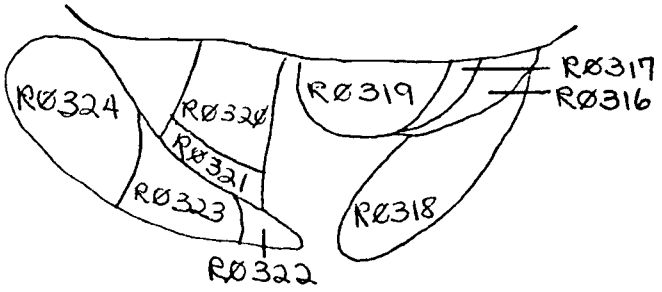


Figure 3.12

Figure 3.13

is applied next. This proceeds in the opposite direction (left to right) for convenience (i.e. maintaining the CALF-SOCK-SHOE order of analysis). The analysis of the leg on the left proceeds as before sharing the same sub-models as the STANDING LEGS Component model for LIMB and SHOE. The bent leg is handled by a separate BENT LIMB Composition model which calls on its own Description models for these altered regions. Within this BENT LIMB Composition model, analysis proceeds horizontally from R0319, the region closest to the first leg, which is labelled as the CALF on this LIMB and the distorted region R0317 is allowed as a SOCK. As in the previous STANDING LEG analysis, the SHOE is assumed to have the HEEL portion occluded.

The analysis of the SHOE then starts from the right of the limb (continuing the horizontal analysis for this leg) to R0316, labelled as the MID-SHOE. Finally R0318, below the MID-SHOE is recognised as the TOE portion. The WALKING LEGS Component model and its sub-models have successfully completed the analysis. In this last example, we see how the Description models can be shared among related Composition models, while still acting with the relational information to select the correct model for the particular scene.

Figure 3.13 illustrates one leg taken from the complete scene of LUCY shown in Figure 3.6. In this sequence the HEEL portion is enlarged and we present the analysis of this portion of the scene by the appropriate SHOE Description model. As we have already shown, the

particular model employs three Description models to discover the mid-portion, heel and toe of the shoe. For this model, size and shape are of secondary importance, the spatial relationships are the crucial factor. Examination of the tests these models employ (see above) show that the actual region to shape tests have been weakened in three ways:

1) The size tests allow a wider margin of error;

2) The acceptance threshold is set much lower;

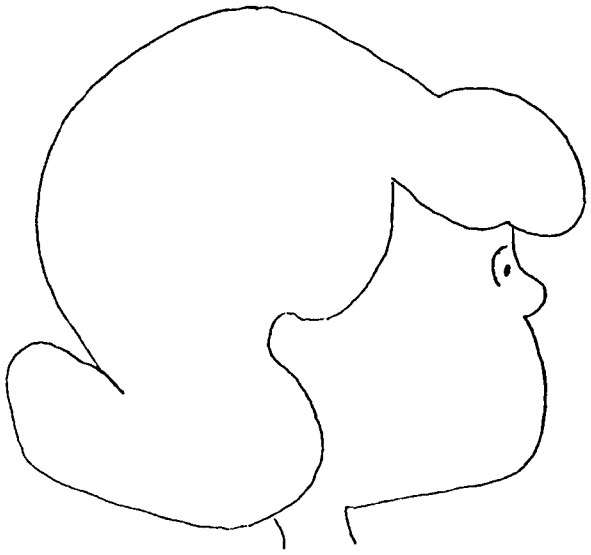
3) There are no specific feature tests.

This leaves the burden of acceptance on the relationship between the regions.

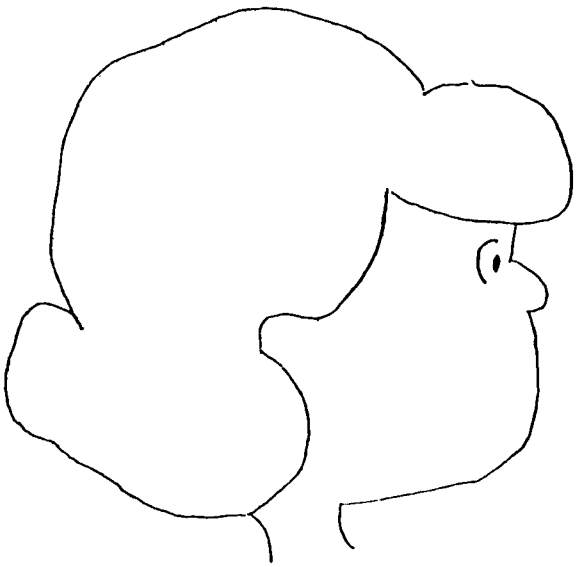
It is not necessary to run the analysis from the top of the picture to the bottom, but it does make the analysis less complicated if an easily identified region is chosen. In some sense, starting at the top of the scene reflects the author's prejudices when viewing the scene; but it also provides important information that eliminates a great deal of trial and error processing. Once the analysis has chosen a successful starting position, it proceeds through the scene using the context of recognised regions to identify their immediate neighbours. For the PERSON model, this tends to impose a top-to-bottom strategy (see Section 6.3).

As a final example for this section, we discuss an experiment which demonstrates the power of the simple yet effective tests that the

Description model uses. Figures 3.14a and 3.14e representing LUCY and VIOLET respectively were entered in the standard manner (see Chapter 5). The intermediate scenes were created by interpolating the corresponding lines in equal steps between the two scenes. These five separate scenes were presented to the HEAD Component model shown in Figure 3.4a. (Since these were isolated heads and not full scenes, the preliminary processing normally performed under program control was pre-arranged). These two particular characters were chosen for their similarity in the hope that the intermediate scenes would test the reliability of the Description models involved. The results of this experiment were consistent with the analysis of the same scenes by human observers. The tests used by the two Description models are shown in Figure 3.15. These two characters were chosen for this test since they are the most similar looking characters in the PEANUTS universe. The chief distinguishing features are that LUCY's hair is flipped up in the back, while VIOLET wears her hair in a pony tail. The results of the test are shown below. The curve extracting procedure is based on the psi-s mapping which maps a closed curve into a function of slope versus arc length. Smooth curves map into straight lines. The crucial low-level data characteristic in this sequence is the inflection point, where the outline reverses its direction. Such a point can be found in both Figures 3.14a and 3.14b. In Figure 3.14c, VIOLET's pony tail is beginning to emerge. The "flip" has been flattened out and the smooth curve over the back of the head is interrupted. This region no longer matches the LUCY HAIR Description model, nor does it yet

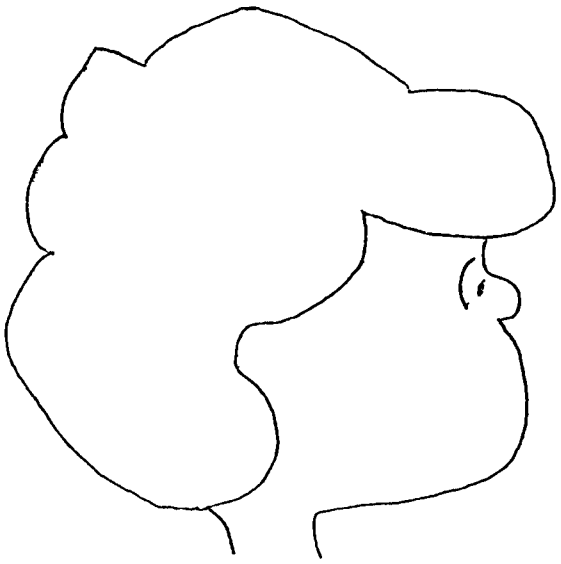


(a)

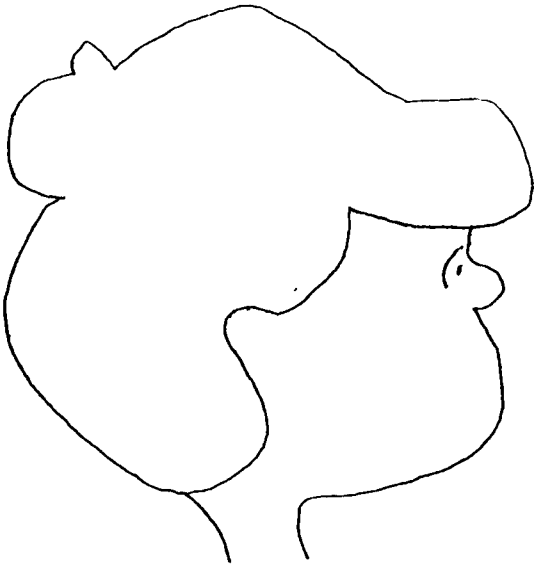


(b)

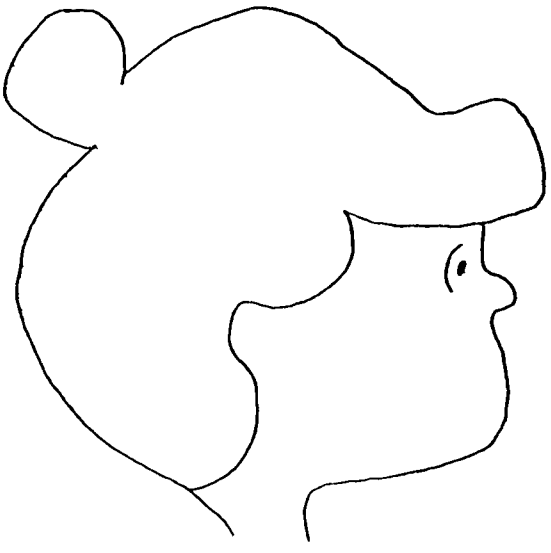
Figure 3.14



(c)



(d)



(e)

Figure 3.14

SIMPLE TESTS USED FOR LUCY/VIOLET HAIR MODELS

LUCY TESTS	VALUE	SCORE
AREA	450	20
X	3000	20
Y	2700	20
"FLIP"		40

VIOLET TESTS	VALUE	SCORE
AREA	390	20
X	2840	15
Y	2500	15

The pony tail segment (as defined by the twin inflection points) is tested separately:

PRESENCE OF SUB-REGION		20
AREA	30	10
X	620	10
Y	740	10

Figure 3.15

conform to the VIOLET HAIR Description model. For VIOLET we require the pony tail, recognised by the twin inflection points and the "smooth" curve between them. The interpolation procedure has not mapped LUCY's inflection point into one of VIOLET's, resulting in the humped shape of the middle scene. This same problem occurs in the fourth scene, Figure 3.14d, which still is not recognisable as either character, by computer or human subjects.

Scene a: Correctly identified as LUCY FACING RIGHT

scores: HAIR 100

FACE 100

HEAD 100

Scene b: Identified as LUCY FACING RIGHT

scores: HAIR 100

FACE 100

HEAD 100

This result was consistent with human observers of the scene. The interpolation has not caused gross changes in the original scene. Note that all scenes (3.14b, 3.14c, and 3.14d) are the results of equally spaced interpolations between scenes a and e. The changes are not very noticeable, because the distinguishing features have not been severely altered.

Scene c: Unidentified

This middle scene has neither the qualities of LUCY nor VIOLET. The Description models selected by the Component model all failed, so the Component model itself failed.

scores: LUCY HAIR 40

VIOLET HAIR 50

Scene d: Unidentified

Due to the interpolation of the points of inflection, VIOLET's pony tail is not sufficiently pronounced.

scores: LUCY HAIR 40

VIOLET HAIR 50

Scene e: Correctly identified as VIOLET FACING RIGHT

scores: HAIR 100

FACE 100

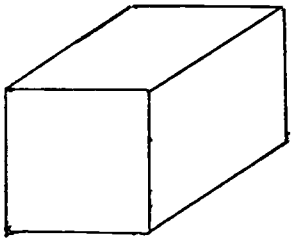
HEAD 100

This simple experiment demonstrates that within an established context the rather crude shape tests which depend on specific features can not only distinguish between similar models, but also reject some scenes which have no corresponding model. The specific features used in the models were chosen prior to this experiment, and not arranged with prior knowledge of the nature of the interpolated scenes.

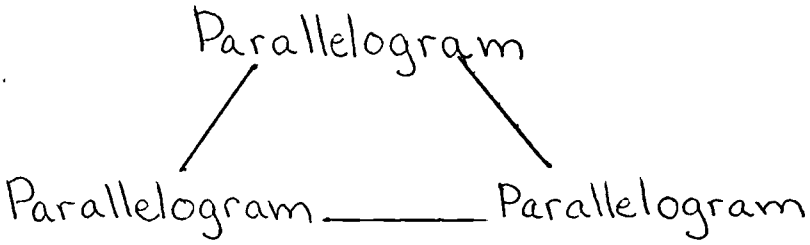
The matching of the system results and the performance of human

subjects seem to indicate that the features chosen for the Description models are the "correct" ones, i.e. the same features that we use to establish the character identity. We do not wish to lay too much emphasis on the rejection of the intermediate scenes by the program. It would certainly be possible to arrange the acceptance of a non-representative shape by the simple models we use. But, this is contrary to the purpose of the program, which was designed to recognise legitimate scenes. To use a simple example from the blocks world [Guzman 1971], in Figure 3.16a we see a two-dimensional representation of a block which might be recognised by the simple relational diagram shown in Figure 3.16b. This simple model would also call the structure in Figure 3.16c a block since the shape and relational constraints of the model are satisfied. If the model is only applied in the appropriate domain, it will be sufficient; if it is to distinguish such impossible scenes from legitimate ones, a more sophisticated model system must be employed. The same is true in our situation. Some "nonsense" scenes may be rejected, others would pass without being noticed. This is the consequence of an inadequate means of fully specifying the curved shapes with which we must deal.

(a)



(b)



(c)

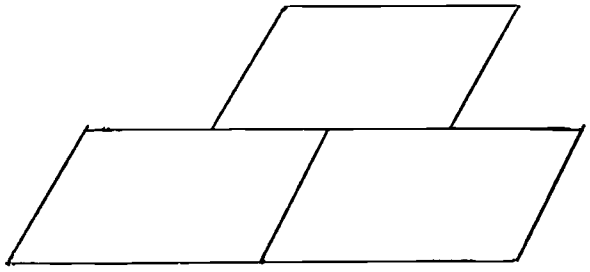


Figure 3.16

CHAPTER 4

OCCLUSION

4.1 Why Occlusion is a Problem

Occlusion is caused by one object or part of an object obscuring another object from view. It is important to be able to discover which objects are occluded because this alters their perceived shape. In geometric worlds, straight lines or even simple curves may be extended to fill in the gaps in a contour produced by an occluding object. Occlusion is merely an inconvenience. Figure 4.1 shows how occlusion can be handled in the blocks world with little difficulty.

The cartoon shapes of the PEANUTS world are irregular and the direction the hidden lines take cannot be as easily predicted. If there was a satisfactory method of describing arbitrary shapes that could be used for prediction, it would simplify the analysis. However, we have already described the difficulties in finding such a method. The system is dependent on the gross features determined by the outline to identify the regions. Occlusion alters just those things that our simple Description models examine: the area, height, and width of a region. Rather than rejecting a match when the score falls below the designated threshold, it first attempts to establish the existence of an occluding region. If such a region can be located, the acceptance threshold is lowered to allow the match to succeed. It is prepared to accept a weak match if it can find a

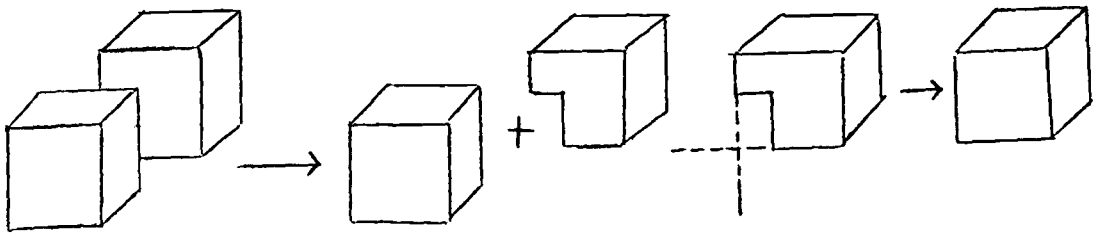


Figure 4.1 Replacing lost information

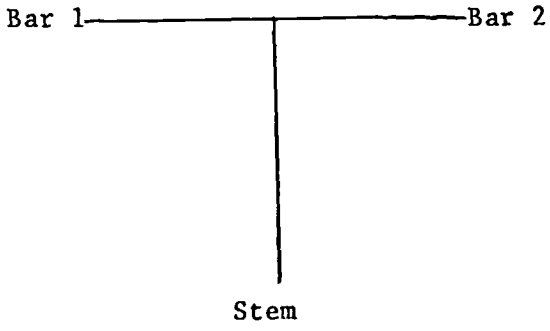


Figure 4.2

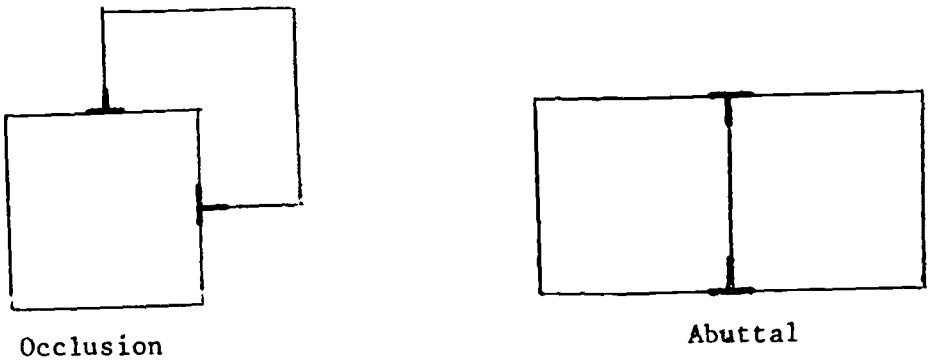


Figure 4.3 Two situations in which T-joints are found

reason (occlusion) for the low score. It only seeks an occluding region if the failure of the description tests are consistent with an occlusion hypothesis, i.e. if a region is too large, it is rejected immediately -- occlusion could not account for that difference.

4.2 The "T-joint"

In order to detect occlusion, the system uses heuristics based on the presence of T-joints. The T-joint is a powerful tool in the study of occluded objects. It is usually formed by two lines which meet with one line continuing through the vertex while the other is terminated (Figure 4.2). T-joints occur in two situations: (See Figure 4.3)

(1) Occlusion. In this situation, a T-joint implies that the boundary of one object has gone under another object, or else that a boundary that was "under" has re-appeared. Thus T-joints typically occur in pairs. The idea that finding the correct pair of T-joints provides information about region occlusion is used as the basis for a powerful pairing heuristic. (See Section 4.3)

(2) Abuttal, i.e. contact with no overlapping.

Because of this second possibility, it would be too impractical to apply the pairing heuristic to all the T-joints in the scene. So, it is used only when a region cannot be matched to a shape description and occlusion is suspected to be the cause of this failure.

It should be pointed out that although we use the term "T-joint" to refer to the junction of three lines, the lines need not be straight nor must they meet at right angles. In this cartoon world almost every line is a curved one. Figure 4.4 illustrates one of the difficulties of working in this curved world. It is sometimes difficult to determine the proper orientation for such "T-joints". While the examination of an isolated junction may leave the choice ambiguous - by making the decision in conjunction with the pairing heuristic (described below) a unique global interpretation is usually achieved. (See Section 4.4.3 for further discussion.)

4.3 The Pairing Heuristic

We have already mentioned that we can determine the occlusion relationship between two regions by finding the appropriate pair of junctions: (see Figure 4.5)

(T1) The one where the boundary of the occluded object disappears.

(T2) The one where the boundary re-appears.

In this example, by pairing the T-joints T1 and T2 we can extract the occlusion information that region RA occludes region RB. We have already pointed out that because of abuttal, we cannot apply the pairing heuristic uniformly over the scene. This case of simple occlusion may illustrate another reason why the occlusion routines are only invoked when recognition of a region fails. This is simply a heuristic -- not a fool-proof method. It is based on the topology

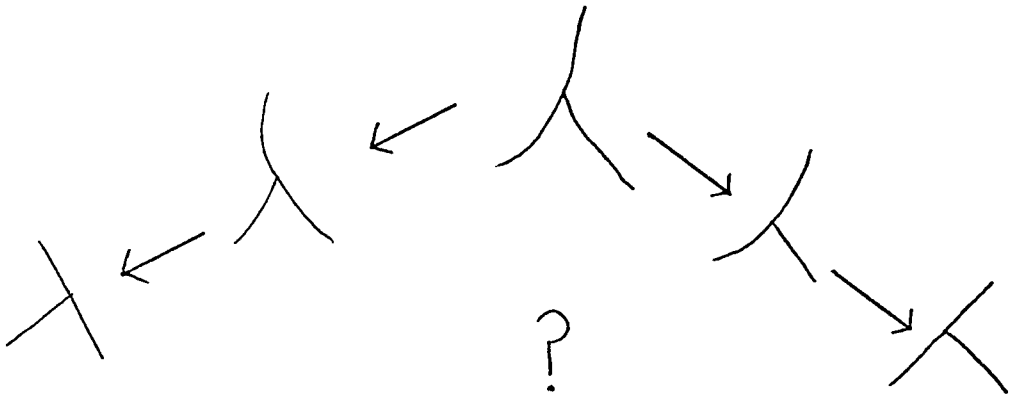


Figure 4.4

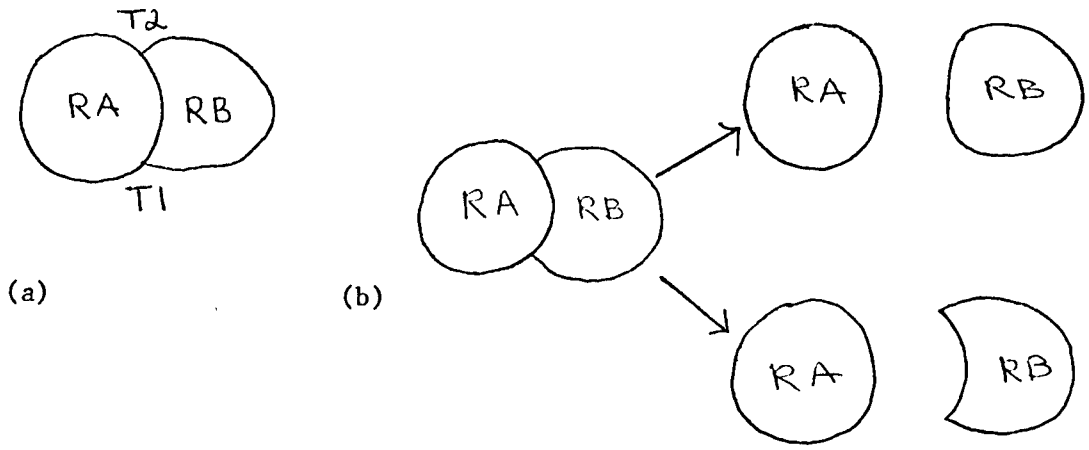


Figure 4.5

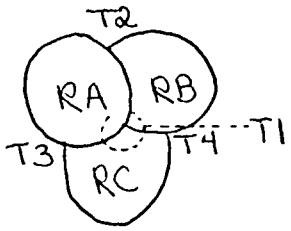


Figure 4.6

of the scene alone, not any high-level knowledge such as models of shapes. The underlying assumptions are that (1) regions represent relatively flat surfaces and that (2) the boundaries of regions are relatively smooth, at least at points of occlusion. By stating that RA occludes RB we are making the assumption that RB's shape is smooth and continuous. In the irregularly curved world that assumption may be invalid. In Figure 4.5b we see two possibilities for the unoccluded shape of RB. In the curved world, the abuttal of two objects may exhibit the same local evidence as occlusion. By attempting to recognise RB before applying the occlusion routine, we minimise this problem.

Cases of occlusion may be more complicated than the case of "simple occlusion" illustrated in Figure 4.5a. Figure 4.6 illustrates a case of multiple-occlusion, i.e. there is more than one layer of regions between the viewer and the object in the scene. In this case, there is a level 2 occlusion of region RC. It is the union of regions RA and RB that hides region RC. From this example, one can see that purely local clues are insufficient evidence for choosing the correct pair of "T-joints". The configuration of the junctions T1 and T3 in Figure 4.6 corresponds to those of T1 and T2 in Figure 4.5, but in this case the pairing is not valid.

The pairing heuristic applies more global knowledge of lines and junctions in the scene to select not only the correct interpretation

of T-joint orientations, but also the appropriate pairings to solve the occlusion problem by locating the occluding region. Applying the heuristic involves the selection of two appropriate junctions on the boundary of the possibly occluded region. These "T-joints" are selected on the basis of local occlusion clues. If we view a single T-joint in isolation (Figure 4.7), we can make certain assumptions about the three regions surrounding the junction. If region RB is the region we suspect is occluded, then region RA is a good candidate for the occluding region since the stem of the T-joint disappears "behind" it. The object is to find two T-junctions (see Figure 4.8) according to the pairing rules:

(1) The T-joints have the occluded region below opposite bars.

(2) Both T-joints have the same region (the occluding region) above their bars.

However, due to the multiple occlusion mentioned above, such local evidence is not sufficient to confirm the occlusion. The heuristic requires us to trace along both the stem and internal bar of one of the T-joints along the boundary of the region and reach the corresponding stem and bar of the other junction according to certain restrictions:

(1) There must be no unpaired T-junctions along the path.

(2) While tracing along the boundary, one cannot step from the bar of an intervening T-joint to its stem or vice versa.

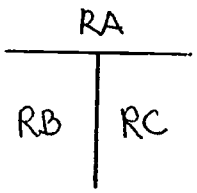


Figure 4.7

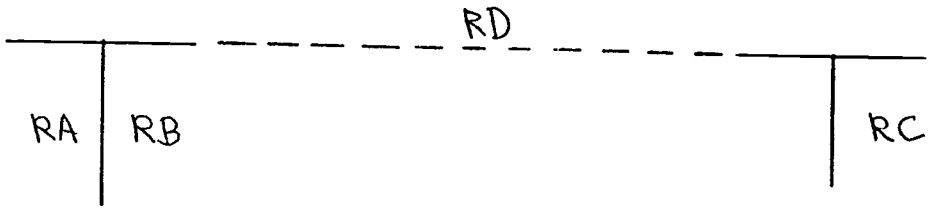


Figure 4.8

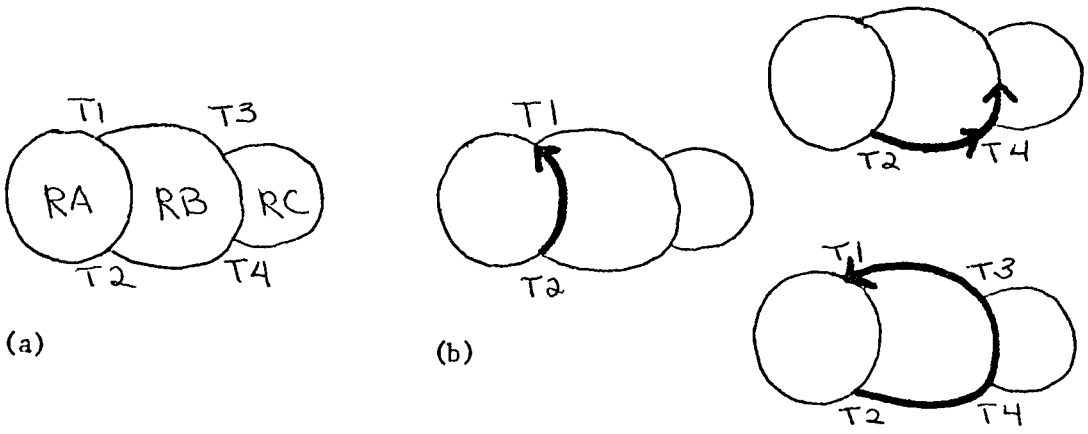
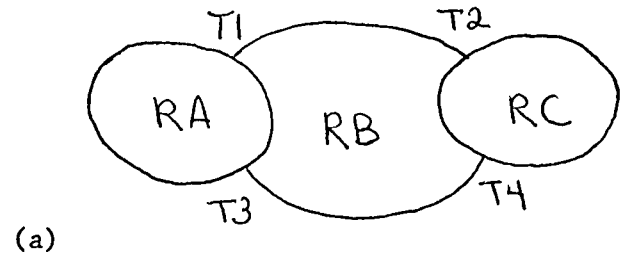


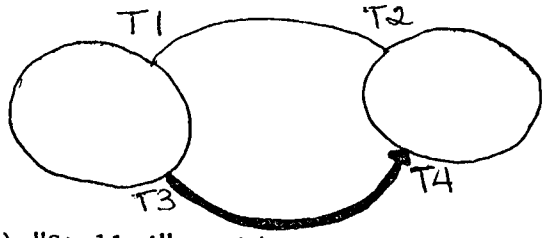
Figure 4.9

These restrictions may appear confusing at first glance; a few illustrated examples should clarify the use of these heuristic rules. To illustrate a simple case of tracing T-joints, examine Figure 4.9a. We wish to show that region RB is occluded. Junctions T1 and T2 are the only junction candidates for pairing according to the rules mentioned above and illustrated in Figure 4.8. The path from bar-2 of T2 to bar-1 of T1 is clear so the bar trace is trivially completed (Figure 4.9b). There are no intervening junctions. The stem trace from T2 runs into T4's bar-1 and continues to bar-2. This bar-to-bar step is permitted; the outline of RB is not disturbed at this point. The stem trace continues through T3 in the same manner and reaches T1. The trace is complete, confirming the predicted pairing of T1 with T2 and their implication that RA occludes RB.

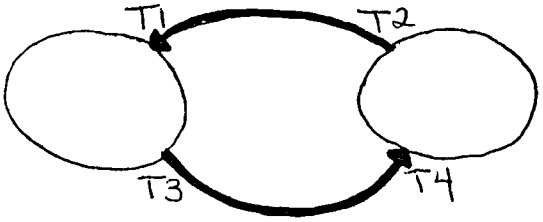
The restrictions mentioned above relate to problems associated with multiple occlusion. We see the application of both restrictions in the application of the heuristic to Figure 4.10. In this scene, RB is occluded by both RA and RC. As initial selection of T1 and T3 again trivially completes the bar trace, but the stem trace violates the second restriction. To continue the trace would mean stepping from the stem of T4 to bar-1 of T4. Instead of abandoning this pairing we establish a sub-goal. We label the original trace as "stalled" at T4, and now try to set up a pairing for that junction. There must again be no unpaired T-junctions along the trace path. The pairing of T2 and T4 is applied. The trace of this pairing is allowed to interact with the T1-T3 trace. Pairing junction T4



(a)

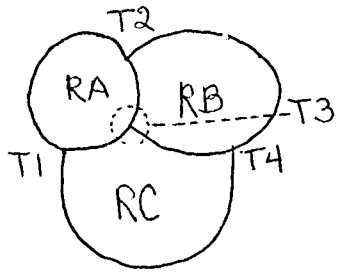


(b) "Stalled" at T4

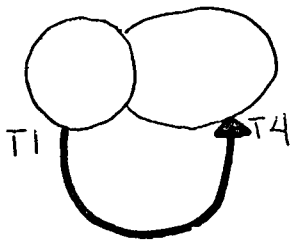


(c) Interaction of "stalled" stem trace

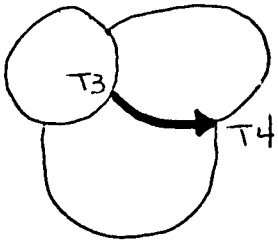
Figure 4.10



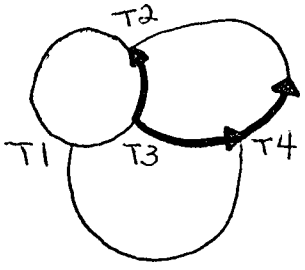
(a)



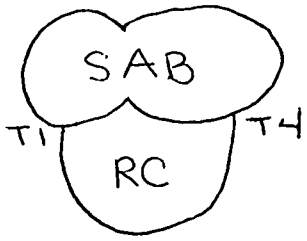
(b)



(c)



(d)



(e)

Figure 4.11

satisfies the stalled stem trace from T3. The stem trace from T2 finally gets us back to the stem of T1 -- the destination of the original tracing. The bar trace of T2 and T4 was again trivial. RB is occluded by both RA and RC. We see from these examples that the bar trace corresponds to the outline of the occluding region along the boundary of the occluded region. The stem trace follows the unoccluded contour of that region.

4.4 Further Heuristic Details

The following discussion concerns the complications arising from multiple occlusions. The casual reader may wish to skip the following discussion of the details involved and turn to Section 4.5 for an example of the use of the occlusion routines applied to PEANUTS scenes.

4.4.1 Dealing with Multiple Occlusion

The necessity for such complicated tracing procedures becomes apparent when we consider cases of multiple occlusion. The local topological cues for the selection of the pair of T-junctions are insufficient. The tracing algorithm provides more global confirmation. Consider Figure 4.11a. If we wish to determine if region RC is occluded, then our pairing rules suggest the (incorrect) pairing of T1 and T3. While it is true that RA occludes RC, a more complete description is that the conjunction of RA and RB occludes RC. The heuristic will eventually obtain this correct solution.

Since this example is not symmetric, the intermediate analysis will be different depending on whether we trace from T1 to T3 or vice versa. We will give details of both analyses. First let us consider the stem trace from T1 to T3. The trace stalls at T4. Since there is no junction to pair with T4, the original pairing must be rejected (Figure 4.11b). Likewise, if the stem trace was in the other direction, i.e. from T3 to T1, we would be forced to violate the second tracing restriction at T4: we cannot step from the bar of a T-junction to its stem (Figure 4.11c). Since the T1-T3 pairing is not valid, multiple occlusion is indicated. The solution involves finding a partner for either of the original pair (T1 or T3) for reasons that will soon become apparent.

The pairing of T3 and T2 is proposed and this tracing succeeds (Figure 4.11d). The result, that RA occludes RB allows us to notionally join the two regions into one <u>super-region</u> SAB as in Figure 4.11e. We have eliminated one layer of occlusion; T2 and T3 have been removed. Now the pairing of T1 and T4 may be proposed and this time the trace succeeds. So RC is occluded by SAB which is the union of RA and RB. If the orientation of all the T-junctions can be determined and there are no cases of abuttal, this scheme may be applied recursively to handle any level of occlusion.

4.4.2 Inadequacies of the Heuristic Technique

While this heuristic is very helpful, there are cases when more information is needed to solve the occlusion problem. In Figure 4.12a the heuristic may be applied to yield the following occlusion information:

(1) RA occludes RB;

(2) The union of RA and RB occludes RC;

(3) The union of RA and RB occludes RD.

The use of some form of model is needed to decide if RC and RD should be joined, i.e. which of Figures 4.12b or 4.12c is the correct interpretation.

There are also cases where the heuristic fails completely because the underlying assumptions of the heuristic are not valid. One common case is that of a hand holding an object. Since the hand surrounds the object, it is both occluding and occluded at the same time.

In Figure 4.13, we humans see a hand holding a ball. But in attempting to apply the pairing heuristic, the system is confused. It can pair neither T1 nor T2. This case violates the assumption that the regions represent relatively flat surfaces. In Figure 4.14, we illustrate a case where abuttal at T3 would fail a T-junction based occlusion analysis. The method we adopt in such instances is

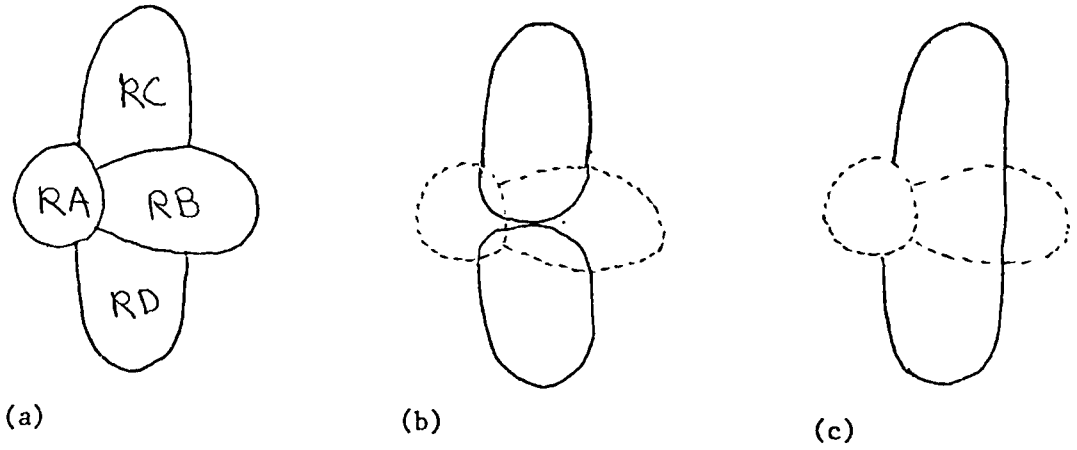


Figure 4.12

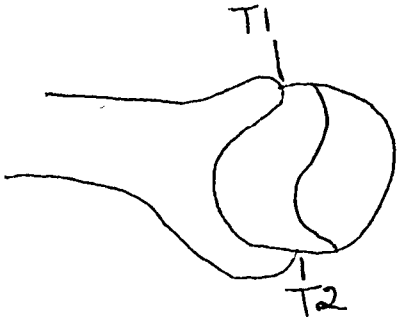


Figure 4.13

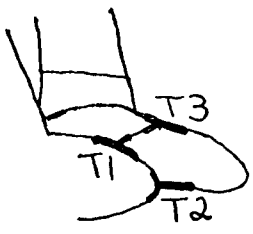


Figure 4.14

quite simple. We associate the necessary additional knowledge with the recognition model. By placing the burden of special case occlusion handling on the specific models concerned, we allow the general occlusion heuristic to deal with cases that meet its criteria of applicability.

Certain common forms of occlusion are treated as abuttal to ease the burden on the occlusion heuristics. As an example of this, consider Figure 4.15. Technically the upper part of each LEG is occluded by the SHORTS, and by the SOCK which in turn is occluded by the SHOE. Such occlusion is quite a common occurrence in PEANUTS cartoons. Originally, we developed another pairing heuristic to handle this case based on the pairing of T-junctions in the configuration illustrated in Figure 4.16. By tracing the stem and both bars of the T-junctions (as described above) we could deduce that either (1) RC occludes RD, or (2) RD occludes RC. The exact interpretation would be supplied by the model. RA and RB are considered to be in the background.

Two factors led to the exclusion of this second pairing heuristic from the present system:

(1) Efficiency. By studying the system as it churns through the scene, it became obvious that it would be more efficient to model the objects as they usually appeared, rather than in their natural isolated unoccluded state. SOCKS for instance, do not

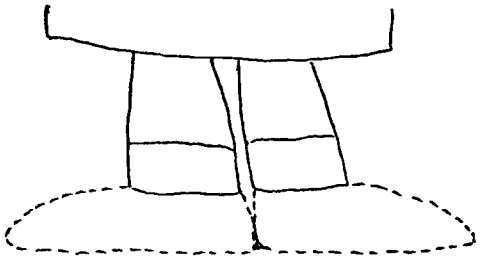


Figure 4.15

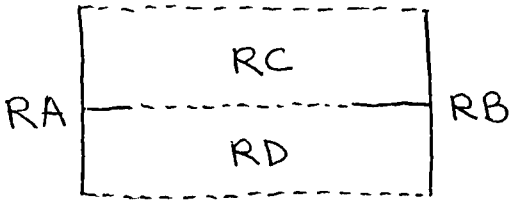


Figure 4.16

(a) (b)

Figure 4.17

appear as in Figure 4.17a but as in 4.17b. That is, socks appear in the PEANUTS context when they are being worn. Since they are worn under shoes, they are always represented by a rectangular form. If the model expects this type of sock, it could save all the time and effort of finding the appropriate pair of T-junctions and tracing the paths of bars and stems that each application of the occlusion heuristic entails.

(2) Personal Introspection. On further reflection, it seemed very likely that human observers do not run through a string of deductions regarding such repeatedly observed examples of occlusion. We learn that a LEG is hidden by a SOCK and a SOCK, in turn, by the SHOE. While we certainly have more complete models of LEG and SOCK, they are not applied in this context. Instead we use a more relevant form (which simplifies the analysis) conforming to our learned cartoon conventions.

So, since (as was stated above) we must in any case rely on the model to interpret the results of this second pairing heuristic, we chose to alter the model and eliminate this occlusion technique. The results are a much simpler and more efficient scanning of the scene with minimal loss of generality in the PEANUTS environment for the system's particular task, i.e. matching regions to models.

There is always a possibility that a new scene may contain an instance of occlusion that cannot be handled by either existing

special case knowledge or the general occlusion routines. In defense of the existing system we raise two points:

(1) The system has the ability to re-start its analysis to gain partial information from the scene if complete recognition is not possible. (See the next chapter.)

(2) These occlusion problems are related to the cartoon environment in which we are working. We are depending on the cartoonist whose intention is to convey information as simply as possible and not to confuse us.

The techniques work for the task we have chosen as well as for the examples. We recognise that more complete three-dimensional world knowledge would have to be incorporated into the system to achieve an understanding of the scenes in a more complete sense.

4.4.3 Orienting a T-junction

Throughout this section we have referred to the problems of pairing T-junctions with little regard to the problem of orienting the junctions that are the intersections of three or more curves and interpreting the results as a "T-junction". We postpone the details of the control flow involved to Section 5.5, but will present the basic method here. Of course, not all junctions must be interpreted as T-junctions. We need only bother with those that border on an occluded region (or one that we suspect is occluded).

A close look at the junctions (Figure 4.18) reveals that the "lines" are really an ordered series of points. The first step is to calculate the angles that each line makes with the junction point. In Figure 4.18 we have (for the curved world) an almost ideal T-junction. We look for a difference in angles of approximately 180 degrees for the cross bar with the remaining third line acting as the stem forming an approximate right angle. In this ideal case there is only one possible interpretation -- line lc must be the stem.

Unfortunately there are more difficult cases, involving two (and rarely, three) possible interpretations, as in Figure 4.19. In this case either lb or lc may be interpreted as the stem. One must resort to more global considerations to obtain the correct answer. The system will allow either interpretation (one at a time). The initial selection of the orientation is guided by the relative values of the angles around the junction. Usually, requiring a consistent pair of junctions will allow only one interpretation to filter through.

In Figure 4.20, both junctions T1 and T2 have two possible interpretations. By application of the pairing heuristic to learn if region RB is occluded, the only pairing possible is with junctions T1 and T2. This forces the second interpretation (illustrated in Figure 4.20) on both junctions. Naturally in the context of a scene analysis with multiple occlusion possibilities the selection of the "proper" orientation is more difficult. At times more than one

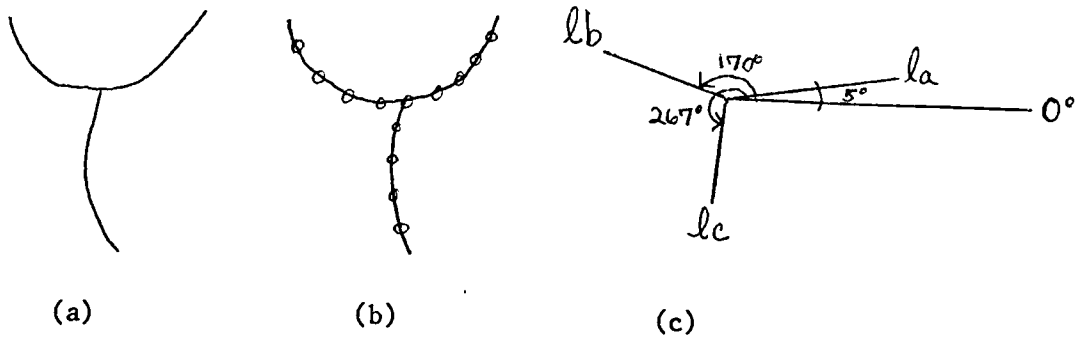


Figure 4.18

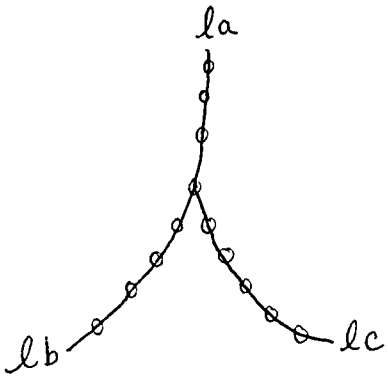


Figure 4.19

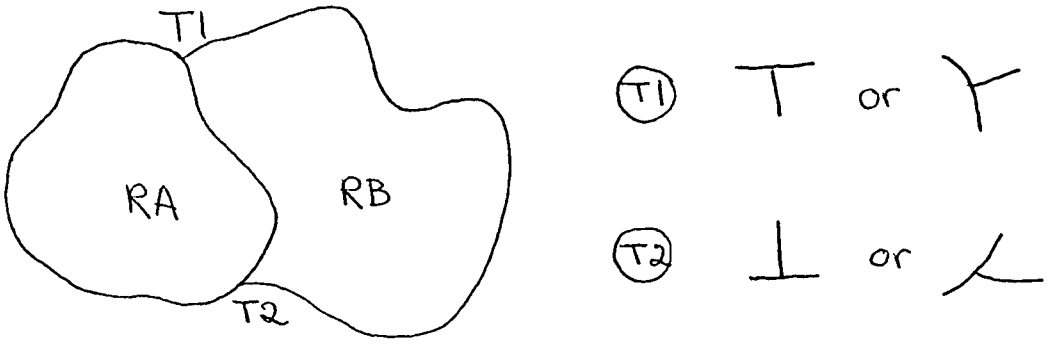


Figure 4.20

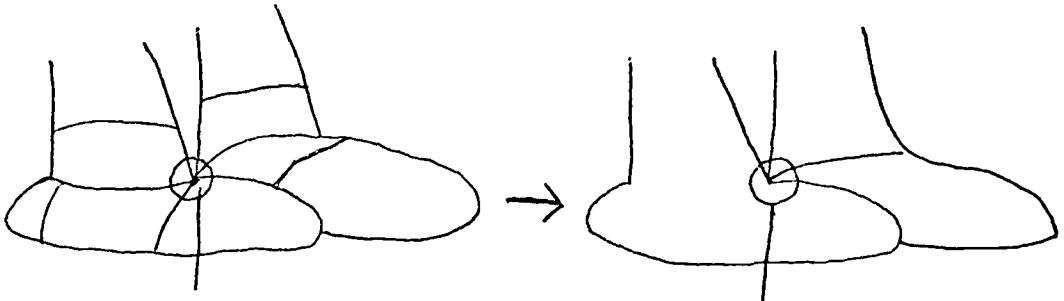
interpretation may be possible. The system adopts the first consistent interpretation it can achieve.

Some junctions consist of more than three lines. To obtain necessary occlusion information such multi-junctions must be decomposed into T-junctions. This task may be facilitated because some of the surrounding regions may have already been identified. By forming super-regions of identified region groups some of the lines can be eliminated. See Figure 4.21. It is not as hard to find the T-joints in multi-junctions as one might suppose. See Figure 4.22. In most cases, the number of possibilities of consistent interpretations is small. One T-junction may be selected, a pairing trace executed, and in the event of failure, another alternative may be selected. The control structure facilitates these trial and error techniques.

4.5 Examples

4.5.1 Interaction with Models

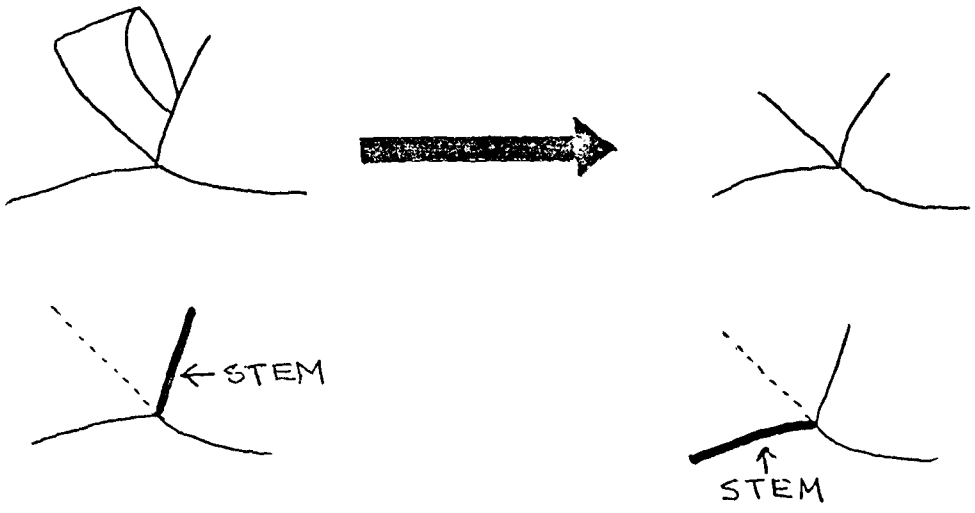
Before presenting a detailed example of the application of the occlusion heuristics in the analysis of a scene, we shall summarise the effects of occlusion on the analysis and briefly describe the interaction of the models with the occlusion heuristics. Occlusion can affect the scene in several ways; naturally some are easier to deal with than others. Let us examine the various types of problem with which it contends:



6-line junction

4-line junction

Figure 4.21



1st ALTERNATIVE

Figure 4.22 This 4-way junction may be interpreted as a 3-way junction in 8 ways. Each line may play the role of the "stem" in two ways. In only two of these are the angle relationships close enough to the 90-90-180 of a true "T-joint".

(1) The simplest case is that of slight occlusion. One region barely overlaps another region. If the occlusion is very slight and none of the essential shape characteristics have been altered, then the Description model will accept the region without even realising that it is occluded. If the shape test results fall below the acceptance threshold in a way consistent with the possibility of occlusion then the occlusion heuristics are applied. A successful junction pairing lowers the acceptance threshold and the match is allowed. The analysis continues as normal.

(2) Rarely, more complicated instances of occlusion may actually split one region into two sections. Merely establishing occlusion may not be sufficient to allow the match despite the lowered acceptance threshold. Neither smaller region alone meets the shape requirements of the whole. By applying a crude measure of shape (the enclosing X-Y envelope) the system can locate candidates for the missing sections (see Section 5.5). With all the pieces contributing to the shape tests the threshold can be reached and the analysis may proceed.

(3) Finally, there is the problem of a region which has been occluded beyond recognition. In such cases, the Description model must rely on the the context evidence of the model hierarchy. It allows the proposed match, but the score that reflects the success of the match will be very low. Future failures which might be caused by an incorrect match at this point will cause the system to fall back to this point to

reconsider.

For the case of an entirely occluded region there can be no proposed match between Description model and region because there is no region. This case is handled at a higher point in the hierarchy, usually at the Component model stage -- at the point where the region to Description model pairing is established. A region can be completely occluded in three different ways:

(1) Self-occlusion -- predicted. This case is seen in Figure 4.23. LUCY's right arm ARM is behind the TORSO. This is not unexpected. The STRUCTURE model "knows" that the TORSO usually at least partially occludes the ARM when the character is in this orientation.

(2) Occlusion with overlapping. In Figure 4.24, the NECK region that the FACE Description model requires for this HEAD orientation is completely hidden by LUCY's ARMS. However, it is not only her NECK that is occluded (completely), but also her FACE (partially). The facial occlusion provides the essential information that is needed to resolve this problem. The area where the neck should be is completely behind the ARMS which have already been classified as occluding objects. The system concludes that the NECK is behind them also and abandons its attempt to locate the NECK region in the scene.

(3) Occlusion with no overlap. This type of occlusion appears as the abuttal of two regions (see Figure 4.25). The occluding region just meets the region adjoining the hidden portion. Not

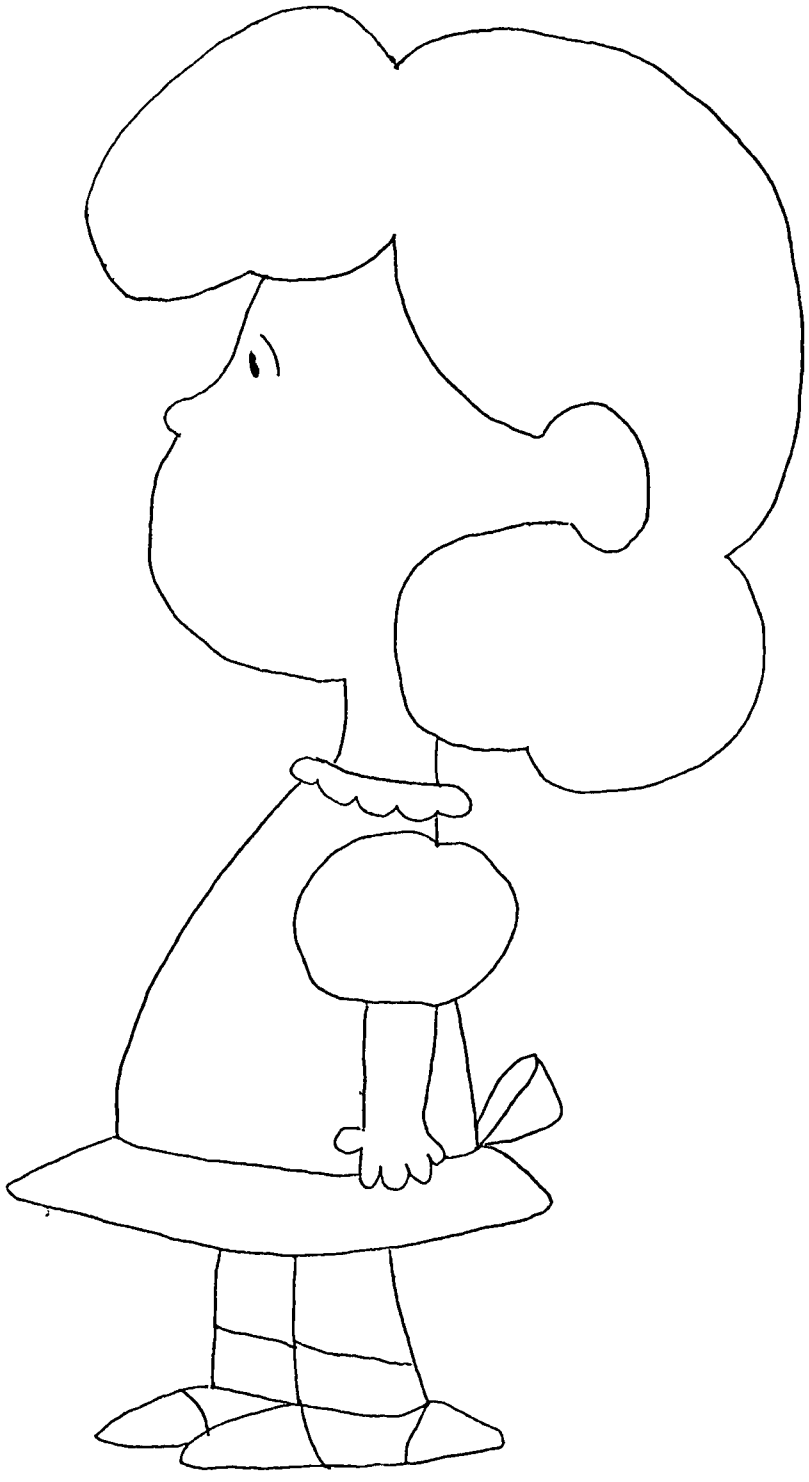


Figure 4.23

Figure 4.24

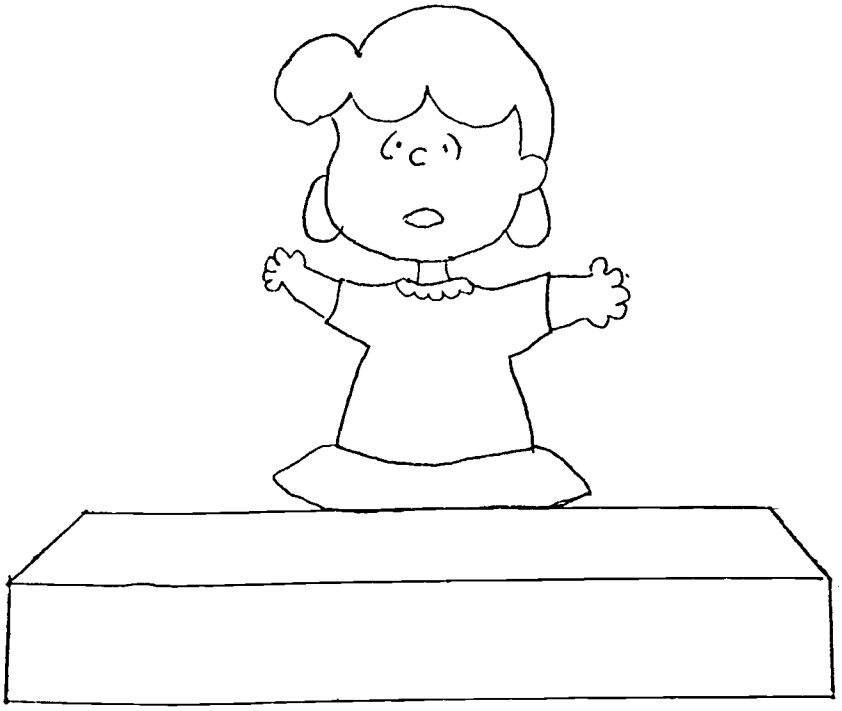


Figure 4.25

surprisingly, this type of occlusion rarely occurs in the cartoon scenes. Probably because of the ambiguous interpretations that may be possible, the cartoonist offers us more clues than are present in this scene. Any missing model part is signalled by the partial occlusion of a neighbouring region. Faced with such an occlusion, our system has been designed to attempt to find the missing parts, but on failure of this to return the partial results obtained through the successful analysis of the parts present in the scene. Section 5.4 explains this phase of the program.

(4) There is one more type of occlusion -- that of occlusion without altering the region outline. Figure 4.26 illustrates a baseball in a baseball glove. Although the centre of the glove is hidden behind the ball, the outline of the glove is not altered, so recognition is not affected. This type of occlusion has not been analysed by the system, but could be accommodated. One minor change to the system would be required: the addition of a new region relationship parameter to indicate that a region was contained within the boundary of another.

We now present a scene containing occlusion that was successfully analysed by the system. As in the previous chapter, we will stress that portion of the analysis guided by the pairing heuristics rather than by models or more global control mechanisms.

Figure 4.26

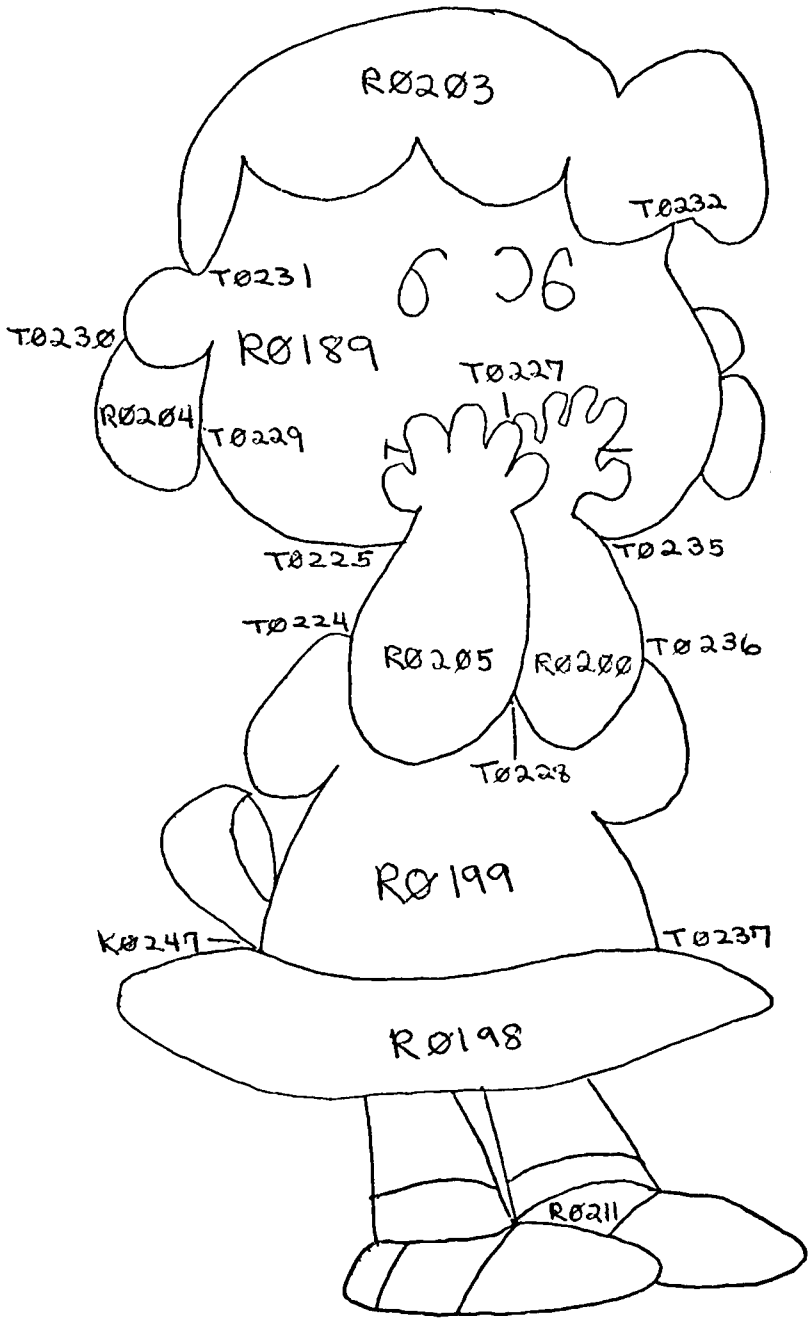


Figure 4.27

4.5.2 Examples of Occlusion Analysis using T-junction Heuristics

The first scene (Figure 4.27) we chose to analyse has several interesting examples of occlusion and led to major modifications to the original techniques we conceived. The scene was selected for the multi-level occlusion by both arms of the face region, and the total occlusion of the neck region. The junction data was far more ambiguous than we predicted; also the full extent of the global consequences of local decisions had not been realised. (See below. Different selections of a single junction orientation produces two different interpretations by forcing subsequent choices.)

We begin our discussion with the occlusion analysis of region R0189 after the recognition of R0203 as LUCY's HAIR facing FRONT-RIGHT. The system searches for evidence of occlusion in terms of T-junctions. There are about a dozen junctions on the border of the region; the first problem is to select candidate pairings from over a hundred possibilities. In this case, the occlusion by the arms does not alter the bounding rectangle, so there are no clues pertaining to the direction of the occluding region in relation to region R0189. To our sophisticated human visual system the "back-to-back" T-junctions T0225 and T0235 seem to be obvious candidates. However, in this universe of irregular curves, the "back-to-back" formation does not possess the important value that it did in the blocks world of Guzman. The stems of the junctions are not co-linear, and in this particular instance of multi-level occlusion, the configuration of

regions surrounding the junctions dis-allows the pairing. If more sophisticated shape descriptions were available, the obvious gap in the face boundary (see Figure 4.28) left by the outline of the hands would probably be detected. Since we depend on the examination of pre-determined features, such an approach cannot be applied.

The strategy we adopted was to provide suggestions for the most likely direction in which to find the occluding region. In the cartoon world, FACES are usually occluded from below. As we shall explain later, this is not a necessary piece of knowledge -- merely a good method of limiting the trial and error decisions needed for the analysis.

The first pairing candidates with the correct region configuration are T0227 and T0225. A successful pairing trace would indicate that R0205 occludes R0189. (This pairing choice involves the selection of an orientation for T0227). The trace succeeds, but the stem trace fails at T0235. There are no other appropriate pairings to try so multi-level occlusion analysis is the only alternative. The nature of the error indicates that region R0200 is an excellent candidate for occlusion. The pairing of T0227 and T0228 is proposed (forcing the necessary orientation on T0228) and this time the trace succeeds yielding the fact that R0205 occludes R0200. A super-region is formed from these two so that now the pairing of T0225 and T0235 can be proposed.

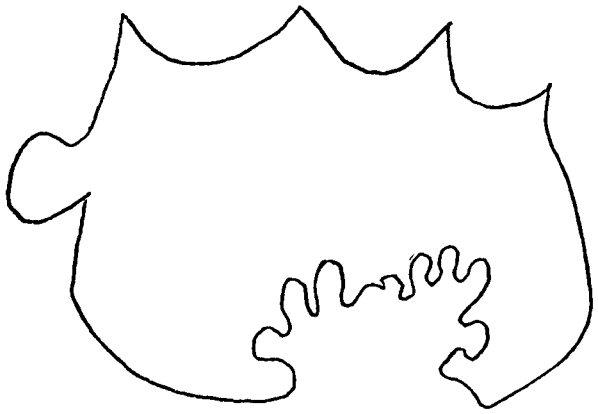


Figure 4.28

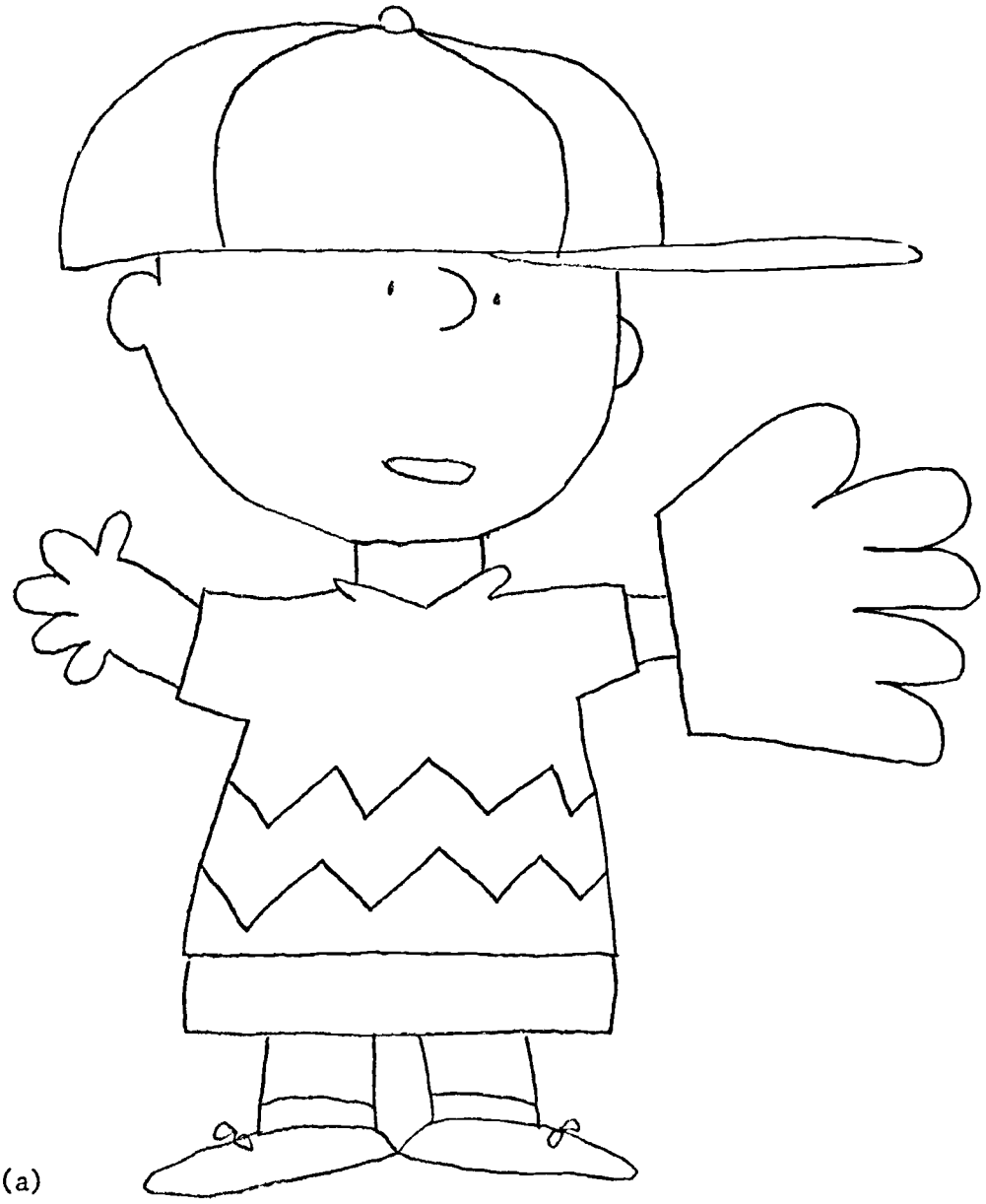
The application of the pairing heuristic can, as we have just described, be used to determine complex occlusion relationships. However, there are occasions when it is inappropriate to apply this technique, as we have already mentioned above. In our experiences with actual scene analysis, inappropriate applications have not caused serious errors. Rather they have pointed to conflicts between the model system and the occlusion processing in their interpretation of the line and region configurations. As an example, consider CHARLIE BROWN's occluded left arm in Figure 4.29a (shown in detail in Figure 4.29b). The pairing heuristic selects junctions T0259 and T0260 to show that R0220 is occluded. The stem trace for this pairing "stalls" at the arm/sleeve boundary. The secondary pairing of T0253 and T0254 allows the occlusion trace to succeed yielding the information that:

1) R0220 (the arm) is occluded by R0219 (the glove), and

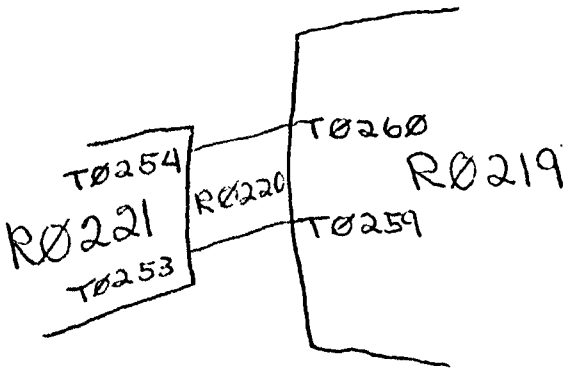
2) R0220 is occluded by R0221 (the shirt).

While the model system considers the arm to "abut" with the torso, the occlusion heuristics see the arm as occluded by the sleeve (item 2 above). The presence of item 1 (the occlusion information that was desired) prevents these conflicting interpretations from having any ill effect on the analysis.

We return now to the scene of LUCY (Figure 2.27). The tracing of the stems of the junctions T0225 and T0235 can be done in two ways, each providing a different three-dimensional interpretation of the scene. Figures 4.30a and 4.30b show exaggerated versions of the two



(a)



(b)

Figure 4.29

interpretations which hinge on the orientation selected for T-junctions T0229 and T0231. The first interpretation is the easier to obtain, the second was the result of system modification and human intervention to arrange for the avoidance of the easy solution. This was done to demonstrate that the system was capable of producing the more complicated solution (in terms of three-dimensional analysis). The simple solution offers a very two-dimensional view of the scene, region R0204 (part of LUCY's HAIR) is simply seen as an adjacent (perhaps occluded) region to the face (R0189). The model system interprets this region as part of LUCY's HAIR, but the three-dimensional knowledge that the hair regions join behind her head is not part of the system knowledge. The recognition procedures are based on the appearance of the regions from a particular point of view ("frame").

The second interpretation only succeeds after a call to the Troubleshooter (see Chapter 5) to handle a three-dimensional occlusion interpretation. This interpretation is a "Kludge" -- it works because it was arranged with this particular example in mind. Its redeeming contributions are to illustrate the power of the Troubleshooter to augment the model hierarchy's interpretation of the scene and to redirect an inappropriate application of the pairing heuristic. Refer to Figure 4.27.

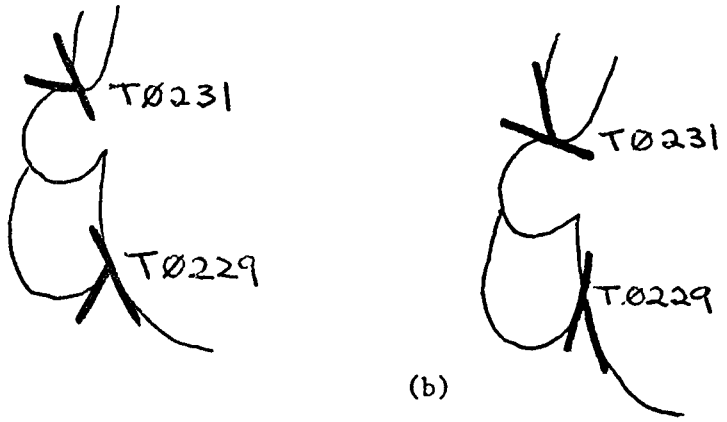


Figure 4.30

Interpretation (a):

Junction T0229 (a Y-junction) is interpreted to allow the stem trace from T0225 to continue rather than "stall". Junction T0230 has only one possible interpretation so the trace continues to T0231. The interpretation chosen here causes the trace to "stall", i.e. the ear is occluded by the hair (R0203). Junction T0231 is paired with T0232, the trace is easily completed. This tracing yields the following information:

(1) The super-region S0300 formed from the union of R0205 and R0200 occludes the R0189.

(2) The hair region R0203 occludes R0189.

This second piece of information is a correct interpretation although the HEAD Component model already takes this into account as the usual case. It is not the cause of the recognition problem. This unnecessary information does not harm the analysis in this case. It illustrates the fact that the model hierarchy might be made less sensitive to cases of expected occlusion such as this, and depend on the occlusion heuristics to discover it. We have not taken this approach for two reasons:

(1) The limitations of the applicability of the pairing heuristic;

(2) The structuring of the model hierarchy allows encoding of such three-dimensional information in a more efficient manner.

Interpretation (b):

In this case junction T0229 is interpreted in the opposite way. The trace stalls at that junction and there is no pair for T0229 to be found. The solution (performed by the Troubleshooter) is based on the interpretation of R0204 as a very curved object that both occludes region R0189 at junction T0229 and is occluded at junction T0230. This complicated solution takes place in five steps:

(1) R0204 is tentatively assumed to be part of LUCY's hair based on the fact that it must be very curved and its adjacency to the assumed face region.

(2) As part of LUCY's hair, it must somehow be joined with region R0203, i.e. it has been cut off by occlusion. A special pairing of T0230 and T0231 is proposed. (This requires the alternative orientation of T0231). No trace is required.

(3) Regions R0203 and R0204 are joined as a super-region.

(4) The stalled trace at T0229 is allowed to continue as junctions T0229 and T0232 are paired.

(5) The trace is complete.

The occlusion information obtained from this kludged performance is essentially the same as before. Region R0189 is occluded both by S0300 (R0200 and R0205) and the hair, S0301 (R0203 and R0204). In this interpretation though, the hair is disappearing behind the ear and curls forward again to obscure a small part of the face.

Such a three-dimensional analysis goes far beyond the capabilities of the pairing heuristic on its own. The sophisticated information that is required by the Troubleshooter is knowledge pertaining to how the body parts contained in the hierarchy are related in a true three-dimensional sense, rather than their appearance in a two-dimensional scene. By augmenting the capabilities of the model hierarchy and the occlusion heuristic by the Troubleshooter's knowledge of specific special cases we can improve the performance of the system. This is a compromise we have taken as a consequence of the difficulties we encountered in our attempts to find a suitable model for the curved shapes of this universe.

As we continue the analysis of the scene, there are two model parts that are completely occluded. The missing neck and collar regions are handled under model control -- not through the use of the T-junction heuristic. However, the heuristic is applied to establish that the blouse region R0199 is occluded. Once again, the tracing of the stems of the T-junctions yields more information than expected. The pairing T0224 and T0236 is proposed. The trace from T0236 stalls at junction T0237. To find an appropriate pair for this junction, the multi-junction K0247, is intepreted as a T-junction of the appropriate orientation, and the trace succeeds. There are two items of occlusion data added to the data base:

(1) The expected occlusion of R0199 by S0300 (R0200 and R0205);

(2) The occlusion of R0199 by R0198.

This second instance of occlusion is another example of the different interpretation of the scene by models and the low-level line tracing procedure. The models treat this as a case of abuttal; the pairing heuristics, as occlusion. Once again, there are no disastrous consequences for the analysis. The extra information is not needed so it is ignored.

In both cases of occlusion analysis that we have shown, the direction of the occluding region was proposed before the junctions were selected for pairing. In the first case this was done on the basis of general knowledge (a table) of facial occlusion; in the second, on local information -- the COLLAR had been totally occluded, so the BLOUSE might also be occluded from above. By choosing junctions for pairing to extract specific occlusion information, we ensure that the results will yield the evidence we require and possibly more. If such directional information is not provided, there is a danger that the occlusion information may not be valid, e.g. that the face is occluded only by the hair region (in a hypothetical case). The danger of such a situation is that the region that was not originally accepted by the FACE Description model may be subsequently allowed on the basis that it is occluded. Since the model already takes the standard occlusion of the face by the hair into consideration, the results of the pairing heuristic should not allow the match. Instead, the model should probably be rejected.

An alternative solution to this problem might be to include a check on each occlusion result with respect to the identities of the regions involved. We have rejected this approach because at the time of the occlusion processing, the region identities have rarely been established.

To conclude this discussion, we wish to emphasise that in almost all cases of occlusion that require the application of the T-junction pairing technique, there is only one candidate pair of junction (without regard to direction), so this is not a serious problem.

Our final concern in this first scene is the occlusion of the shoe. In the previous Chapter, we discussed the analysis of such leg configurations under model control. The earliest version of the system depended on the pairing heuristic to uncover the evidence of occlusion. The success of this analysis depended on rather arbitrary decisions on when to interpret multi-junctions as T-joints and when to disregard them. In this particular example the success of the analysis depends on the realisation that region R0211, the middle portion of the shoe, is occluded. The crudeness of the shape descriptors we employ, and the variety of shapes used in shoe drawings meant that we had to adjust the parameters carefully to insure that the Description model for MID-SHOE initially rejected R0211 and called in the occlusion analysis. We present the details of this analysis to illustrate the awkwardness of this approach. The

desire to avoid such complications motivated our decision to place the burden of the analysis of such a typical configuration on the model.

For the purpose of this discussion, refer to Figure 4.31 for the regions involved in this scene. The near leg has already been analysed and the regions have been merged. We begin our explanation at the point where R0211 has just been rejected by the MID-SHOE model and the occlusion routines have been called in. There are three junctions on R0211's border, only one of them is a T-junction. To form a pair J0018 is intepreted as a T-junction while the remaining multi-junction, J0009, is not. This allows the pairing trace to succeed yielding the result that R0211 is occluded by S0310, the other leg. Region R0212 is recognised without resorting to occlusion processing since the occlusion is too slight to be noticed at the model level. The missing heel region is excused because the adjacent region R0211 was occluded. Without the prior occlusion information, the missing region would have to be accounted for by the Troubleshooter using three-dimensional knowledge about legs. By ensuring that the occlusion of R0211 was detected, the subsequent total occlusion problem is simplified. We felt that such common circumstance should not require the complicated Troubleshooter solution, (nor the contrived Composition model), so the LEGS model was modified to reflect the two-dimensional appearance of this three-dimensional configuration of legs.

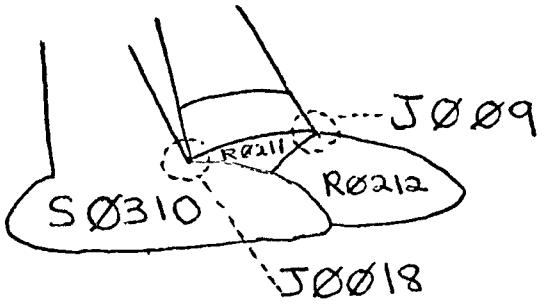


Figure 4.31

For our discussion of Figure 4.32, we concentrate on the occlusion problems related to the two pieces of hand-held baseball equipment. The baseball occludes the hand portion of the arm-hand region breaking it up into six smaller regions: one for the arm and one for each of the five fingers. Region R0304 fails to meet the Description model's shape requirements, so the occlusion routines are called in. The suspected area of occlusion is at the left-most portion of the region -- away from the SLEEVE portion (for details see Chapter 5). Junctions T0359 and K0378 are the only candidates and their orientations are chosen to permit the pairing to proceed. (The details of the orientation selection of each junction are presented in Chapter 5). As before, this pairing trace involves the further pairing of T0357 and T0358, to yield

(1) R0304 is occluded by R0302 (T0359 & K0378)

(2) R0304 is occluded by R0305 (T0357 & T0358)

In this rather complicated example, such confirmatory evidence is only the first step in the occlusion analysis. There are many difficulties involved in the interpretation of scenes related to the problems of hand-held objects:

(1) The interpretation of a grasping hand requires a true three-dimensional model. The model hierarchy depends on region relationships to determine depth information. A single surface (region) which exists both behind and in-front-of another object requires the application of knowledge from an additional, external source.

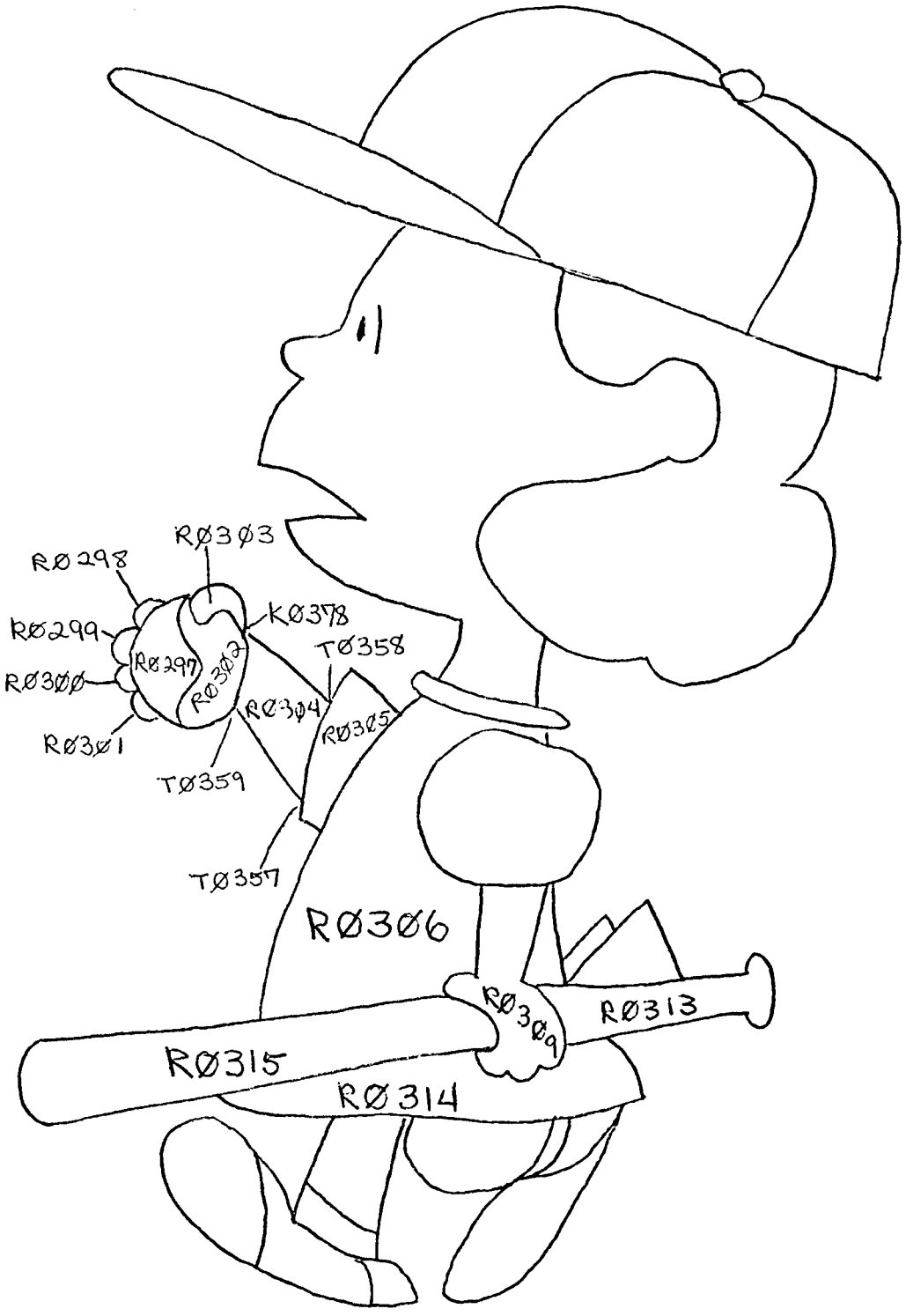

RØ298
RØ3Ø3
RØ299
KØ378
TØ358
RØ297
RØ3Ø2
RØ3ØØ
RØ3Ø4
RØ3Ø5
RØ3Ø1
TØ359
TØ357
RØ3Ø6
RØ3Ø9
RØ313
RØ315
RØ314

Figure 4.32

(2) The held object is usually occluded by the hand and may occlude other parts of the body. The occlusion of the object is more serious than the occlusion by the object. While the partial results of the PERSON model provide strong context information to guide the analysis through this occlusion of a sub-model, the appropriate model for this object, must be selected on the basis of the occluded instance.

(3) The object (whatever it is) may interfere with the analysis of the PERSON model. The strong dependence on relational information may lead to the acceptance of the object as a body part simply because it is approximately the right shape and conforms to the simple region relationship criteria contained in the model hierarchy. (See the baseball bat example below).

To cope with the first of these problems, a special set of procedures has been incorporated into the system to direct the analysis of the hand when it is gripping an object (signalled by the detected occlusion of the hand). The need for this auxiliary knowledge points to one weakness of the hierarchical model system's region analysis; it is essentially a two-dimensional analysis. In most cases this is sufficient, but hands require more sophisticated knowledge. In the current system, this hand information is associated with the Troubleshooter (Chapter 5). (The organisation of the system would be improved if this special knowledge was originally associated with the PERSON model and added to the Troubleshooter's repertoire of special cases when the PERSON model was invoked. This certainly could be

arranged; however, the scenes that have been analysed have all been PERSON-oriented and PERSON-related knowledge has not been isolated from the other specialised knowledge embodied in the Troubleshooter code.)

Once the baseball (R0297 and R0302) has been recognised (see Chapter 5), the five fingers remain unidentified. Based on the knowledge in the data-base of partial results, these regions are grouped with the arm region R0304. Essentially, they are recognised as fingers (although they are not labelled as such since arm, hand, and fingers are usually represented as one region recognised by the ARM model.) The specific data-base entries needed are:

(1) (OCCLUDED-BY R0304 R0302)

(2) (COMPONENT-MODEL ARM R0304 HORIZONTAL LEFT TORSO 50)

(3) (STRUCTURE-MODEL BALL S0412* BASEBALL STANDARD TOP-LEVEL 100)

*Where super-region S0412 is composed of R0297 and R0302

(4) Plus the region relationships:

(RIGHT R0297 R0298) (ABOVE R0298 R0297)

(RIGHT R0290 R0299) (ABOVE R0297 R0301)

(RIGHT R0297 R0300) (ABOVE R0303 R0302)

(RIGHT R0297 R0301)

The reasoning behind the analysis is based on these arguments.

(1) If the ARM is occluded by some object

```
(PRESENT (AND (DESCRIPTION-MODEL ARM !>arm-region ...)
              (OCCLUDED-BY !,arm-region !>objx)))
```

(2) and there are small unidentified regions surrounding the object

```
(AND (OR (PRESENT (NEIGHBOUR !,objx1* !>finger))
         (PRESENT (NEIGHBOUR !,objx2* !>finger))
         (PRESENT (NEIGHBOUR !>finger !,objx1))
         (PRESENT (NEIGHBOUR !>finger !,objx2)))
     (UNIDENTIFIED !,finger)
     (LESSQ (AREA !,finger) 5))
```

*(objx1 and objx2 represent the component regions of objx -- S0412)

(3) then label the regions (!,finger) as part of the ARM

```
(ADD (DESCRIPTION-MODEL AUX-ARM !,finger HORIZONTAL LEFT
TROUBLESHOOTER 70))
```

(4) and finally gather them all up as a super-region with the original arm

```
(ADD (COMPONENT-MODEL ARM S0413 HORIZONTAL LEFT TORSO 67))
```

Naturally, further refinements to this simple scheme could be made. This has proved sufficient for this particular scene as well as other scenes that have been studied by the author (but not analysed by the system).

An alternative method of solving this type of problem, would be to include special Description and Composition models in the PERSON model hierarchy to deal with such cases, rather than delegating the responsibility to the Troubleshooter. This method was rejected on the following grounds:

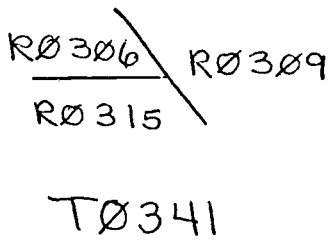
(1) Such an occlusion problem involves a very specialised region analysis; not the general shape/relational technique that typifies the model hierarchy.

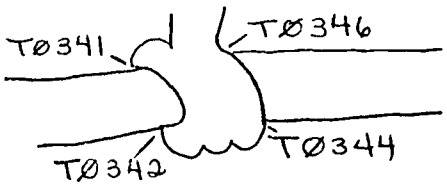
(2) The analysis involves an interaction between two distinct objects, while the model system is designed for intra-model analysis.

(3) The analysis is essentially a clean-up operation, applied when all other regions have been identified. (once again because of limited confidence in the shape models). By handling this occlusion problem before the other components have been identified, there is a high risk that some of the wrong regions would be gathered up. It is easier to postpone the decision, than back-tracking to reconsider.

We move on now to the analysis of the baseball bat (R0315 and R0313 in Figure 4.33). This time the arm is easily recognised -- the three-dimensional occlusion problems are only discovered during the analysis of the bat. The failure of the region to match the bat model invokes the occlusion routines. In this case, the object is broken into two regions which must be re-united; the shortened length provides the clue to the need to search for the second region. See Chapter 6 for a fuller discussion of the problems involved in such cases of occlusion.

For the present discussion, we join the occlusion analysis in its


RØ3Ø6
RØ3Ø9
RØ315
TØ341


TØ341
TØ346
TØ342
TØ344

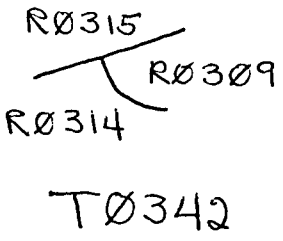

RØ315
RØ3Ø9
RØ314
TØ342

Figure 4.33

attempt to discover if region R0315 is occluded. All four neighbouring T-junctions have only one possible (non-ambiguous) orientation and no combination of the four is appropriate for the pairing heuristic. In most ordinary cases, this would signal the failure of the occlusion analysis and therefore imply that the model-to-region match must be rejected. However, this is a special case.

Region R0309 has been identified as the ARM so the special three-dimensional analysis may be applied. Figure 4.33 shows the relevant portion of Figure 4.32 in detail.

Taken individually, junction T0341 indicates that R0315 (and/or R0306) is occluded by R0309; junction T0342 shows that R0315 occludes R0309 (and/or R0314). This interpretation, that regions R0309 and R0315 occlude each other coupled with the knowledge that region R0309 has been identified as the ARM allows the occlusion analysis to succeed.

The occlusion analysis of region R0313 is straightforward. Both juncions T0344 and T0346 have only one possible orientation. They can be paired and traced without resorting to the special case analysis. This illustrates the design philosophy of the system -- always try the standard solution first before resorting to special routines which may make inappropriate assumptions concerning the scene and thereby cause problems later in the analysis.

CHAPTER 5

CONTROL

In this chapter we discuss two aspects of the control mechanisms that govern the execution behaviour of the system:

(1) The general behaviour described by the system when following the data from the initial cartoon scene through the final analysis.

(2) The intricate control mechanisms involved in model invocation, the occlusion routines, and the interactions between them.

5.1 Preliminary Data Processing

5.1.1 Extracting the Data from the Scene

The first step in the analysis is to extract the cartesian coordinates for a closely spaced series of points along the lines that define the scene. We use a Ferranti Digitiser for this purpose. As the picture is traced by hand, the digitiser registers the X-Y coordinates for a selection of points and produces a paper tape output. The rate of sampling can be set automatically to register points whenever the traced path differs from the last recorded point by a set distance. This automatic sampling may be supplemented by recording of specific points at the discretion of the user. In this manner we obtain an ordered set of X-Y coordinates for each line in

the scene. Junctions between lines are taken as the end-points for each ordered trace.

The resulting data captures the information in the scene in the same manner as "connect-the-dots" pictures in children's drawing books. The original size of the input picture has varied from scene to scene. The sampling distance was selected to capture a reasonable amount of detail for the program and the display of the re-composed picture. There was no critical setting involved, the shape parameters we employed were not sensitive enough to make this an issue. The equally spaced points were only supplemented by selected points when fine details characterised by closely packed curves were necessary, (e.g. fingers) or when a vital discontinuity fell too far from the designated sample point. Typical sampling distances for a figure eleven inches in height were slightly less than one-tenth of an inch. For the scene in Figure 5.1 there were 95 lines with between 2 and 48 points per line (average of 9). Figure 5.2 illustrates a case that calls for supplementary points for both reasons mentioned above. By using a much smaller sampling distance these supplementary points would not be required. This compromise approach was chosen to reduce the amount of data needed to encode the scene.

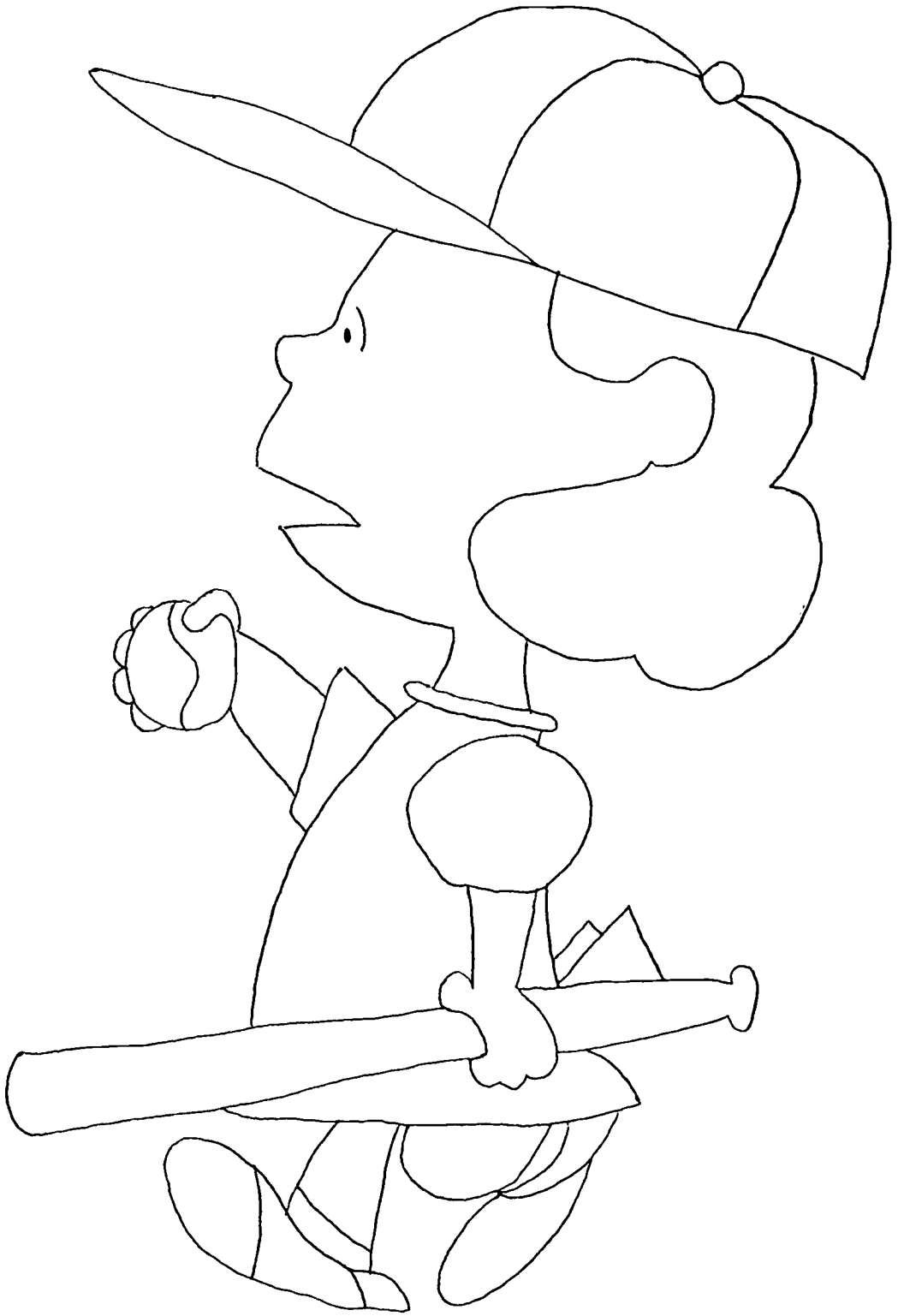


Figure 5.1

• automatically sampled

× selected for detail

⊗ selected to capture discontinuity

Figure 5.2

5.1.2 Collating the Data

In the next stage of the process, the paper tape containing the image data is fed into the computer. The data is transformed into LISP compatible format, and the lines are given unique labels, as are their end-points or "junctions". The subsequent step of this process is to determine the boundaries of closed regions in terms of their component line segments by studying the configuration of lines at the junctions. First, the number of junctions are reduced by merging those with coordinates within a pre-determined distance of each other, then the lines associated with each now unique junction point are ordered according to their angle with respect to the junction. The region boundaries are established (following Roberts [1965]) by tracing each line from one junction to the next, and selecting the next line in a counter-clockwise direction until the starting junction is reached. One such boundary will be the "outer closure" of the connected group of regions; it is isolated from the regions by examining the direction of the traced boundary. All true regions will be traced in a clockwise direction; the outer closure, counter-clockwise. Finally, there may be some lines that are not included in any closed region boundary. These are called "surface-markings" and are associated with the region which surrounds them. Facial features such as eyes, nose, and mouth are typical examples of surface markings in these cartoon scenes. See Figure 5.3 for a sample of the data representations.

Data Representations

```
B  8.732  7.115        Digitiser output
   8.694  6.994
   8.707  6.895
   8.733  6.814
   8.767  6.749
   8.800  6.723
```

After Pre-processing

```
Line        L0091  (POINTS ((8732 . 7115) (8694 . 6994)
                           (8707 . 6895) (8733 . 6814)
                           (8767 . 6749) (8800 . 6723)))

                   (JUNC+ J0092) (JUNC- J0093)


                   JUNC+ indicates the starting point
                         of the line trace
                   JUNC- indicates the final point


Junction    J0092  (ENDP (L0091 L0047 L0097))
                   (COORDS (8732 . 7115))

Region      R0200  (BOUNDS ((L0015 . +) (L0067 . -)
                            (L0109 . -) (L0112 . +)))


                   The "+" indicates the line is used in  its
                          original traced direction
                   The "-" indicates that the line is used in
                          the reverse direction


Surface-Marking  M0192  (Line L0166)
```

Figure 5.3.

5.1.3. Preliminary Processing for Scanning Program

Once the data has been coded in the proper format, some further pre-processing is performed. The CONNIVER system [McDermott and Sussman 1972] we employ is a large and unwieldy system on its own. As we add our procedural models to the data-base and our own control mechanisms to it, the system grows to enormous size and consequently swamps the operating system on our PDP-10 computer. To ease the burden placed on the system, we perform all the necessary shape tests required by the models and place the results of these shape tests on the property lists of the regions. This preliminary processing offers us both a saving in space (the code to perform the shape tests and the low-level data itself need not be added to the CONNIVER program), and time (the execution time for the analysis program represents the time for the higher-level analysis -- most low-level processing has already been performed). We stress that this decision is one of practicality only; it does not affect the generality of the solution. With a more efficient system, our function application of (AREA R0200) which simply finds the AREA property of R0200 could actually derive the area from the low-level data.

There are several other examples of such practical pre-processing. The angles of lines around junctions are used to select good T-junction orientation possibilities. In most cases, these results are not conclusive. The possibilities are ordered in order of closeness to a true "T", the proper orientation will be selected

during the application of the T-joint pairing heuristic in the actual scene analysis program. The directional relationships between neighbouring regions is also determined. These relationships are based on only four directions: up, down, left and right. Some region pairs may exhibit more than one of these four. Although these directional attributes are crude, they have proved sufficient for our applications. In fact, such coarse directional information proved to be a benefit in this cartoon environment, by allowing the region relation models a wider scope of application. These results are coded to be passed on to the data-base for analysis of the scene based on all this pre-processed data.

5.2 Identification

With the completion of the preliminary processing, the actual scene analysis can begin. Figure 5.4 illustrates the block structure of the overall system. The process is initiated with a call to the procedure labelled SCAN in the figure.

5.2.1 Selection of the Structure Model

In Chapter 3 we discussed the operation of the model hierarchy. The first step in this process is the selection of the proper Structure model for the scene. For our simplé scenes of isolated characters, the outer closure provides the essential clues for this process. The shape and size of this outline may be used as the basis for making a hypothesis about what the object might be. The ratio of height to

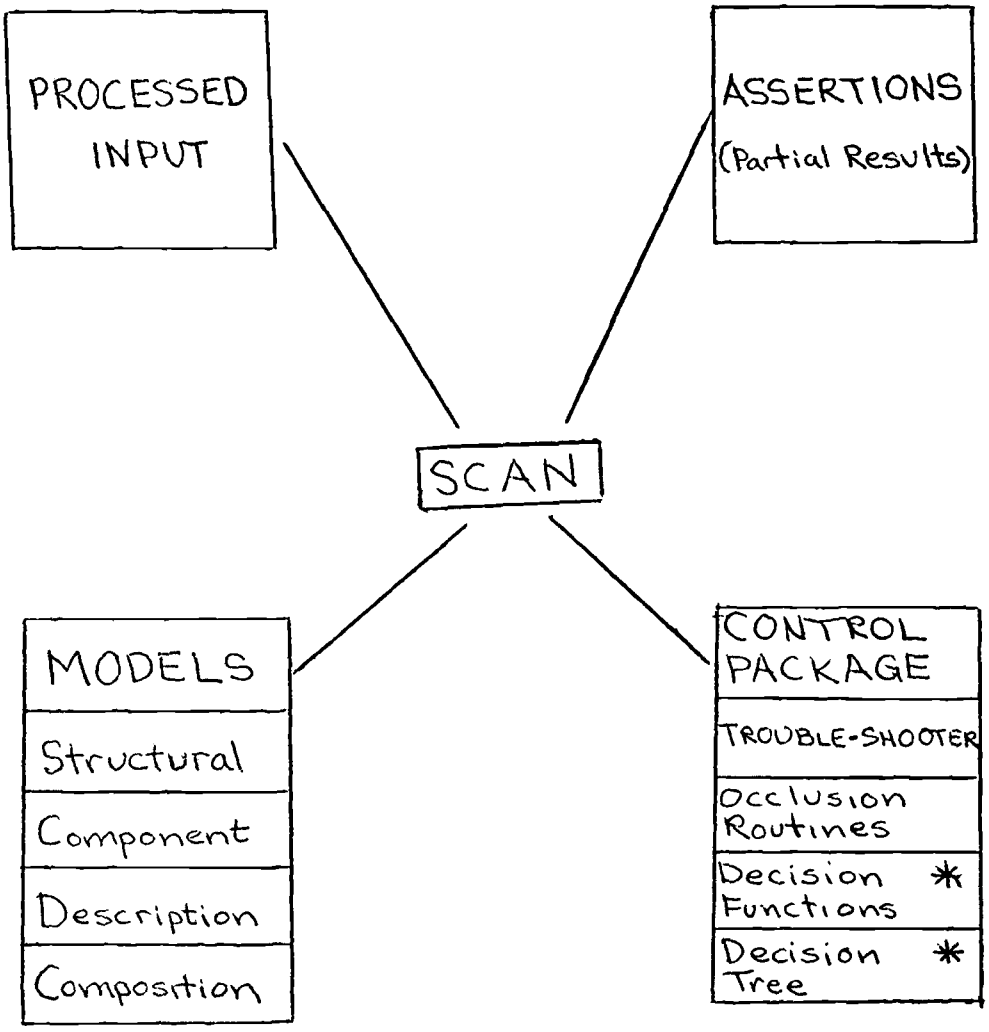

PROCESSED INPUT
ASSERTIONS
(Partial Results)
SCAN
MODELS
Structural
Component
Description
Composition
CONTROL PACKAGE
TROUBLE-SHOOTER
Occlusion Routines
Decision Functions *
Decision Tree *

Figure 5.4

width as well as the area and orientation for Figure 5.5a conforms to the PERSON model. Only ratios can be used until the model match sets the proper scaling factor. If the figure is easily isolated, then it is quite simple to distinguish a PERSON from a KENNEL or a BASEBALL. Any mistakes concerning model selection will generate an error at some stage of the process, and another possibiity may be tried.

Isolating the figure is a real difficulty at this stage. If different objects are contiguous, then the outer closure method fails since it will encompass the boundaries of both objects. If one of the objects is small compared to the other, it will not affect the gross shape comparisons. However, if both are the same size, the composite shape gives no clues to the appropriate model. Luckily PEANUTS figures are usually isolated, but we have developed some ideas (not implemented) for extracting the separate objects from such conglomerations. One idea is primarily an extension of the outer closure method to include standard cartoon conjunctions, e.g. SNOOPY on his KENNEL, or LUCY, SHROEDER and PIANO in their standard configuration (see Figures 5.5b and 5.5c). Another method is based on the isolation of heads in the scene. The head provides more detailed information about the character than any other body part, and provides the best starting point for the analysis of the whole person (as we discussed in Chapter 3). The application of one model to the scene will allow its sub-parts to be identified. By finding a PERSON's head we have an excellent starting point for the model; its constituent parts will be recognised leaving only those regions that

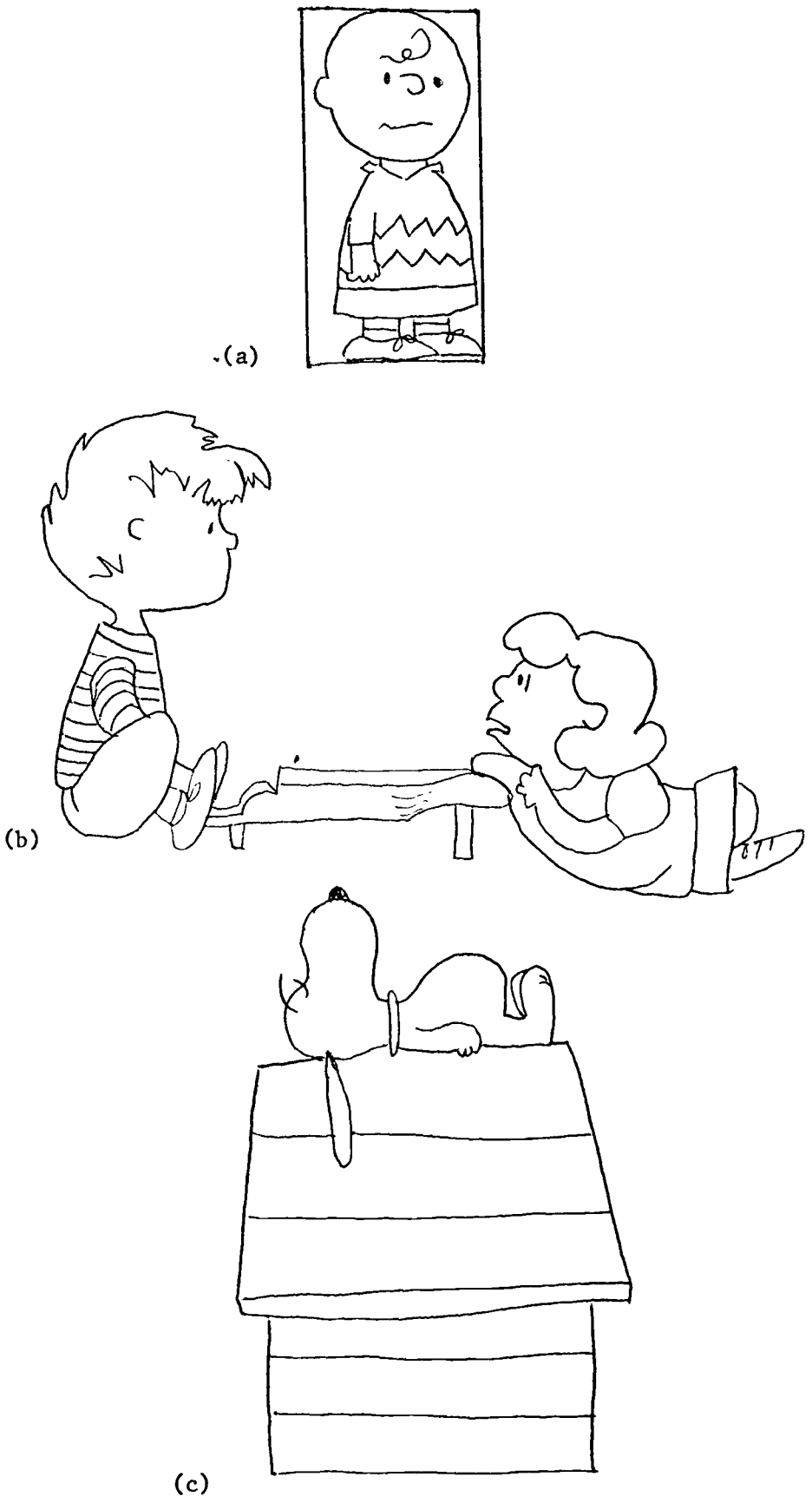


Figure 5.5

belong to other objects unidentified. In almost all cases, the heads are raised above the rest of the picture, making them easy to find by locating the highest region on some local level. This seems to be a promising approach, although such isolation techniques have not been incorporated into the existing system.

A more satisfying approach would be to apply some low-level techniques to recognise crucial features and thereby determine the correct model to guide further analysis. Examples of such easily recognised features might be the surface-markings which represent facial features to find FACE's or Charlie Brown's zig-zag stripe to find his SHIRT. While such special-case examples come easily to mind, other objects are more difficult to characterise on the basis of special features. The use of data-driven techniques to invoke models has been applied in restricted domains such as the blocks world (e.g. Grape [1973]) and Sloman's "dotty pictures" [1977]. In our domain the lack of a uniform system for the description of curvature limits the possibilities for low-level guidance of the analysis. We readily admit that the shape classification techniques we use are primitive, and it is this weakness that has led to the crude technique of outer closure analysis to select a trial Structure model. If the shape descriptions alone could identify a region, then such low-level techniques could play a much more important role in the analysis. Instead the Structure model is selected on the basis of:

(1) Orientation (angle of longest axis)

(2) Ratio of height to width (based on orientation)

(3) Area (occupancy of enclosing rectangle).

We regard this process as an approximation to a more global view of the scene. The outer closure acts as a "blurred" view of the scene, and the overall shape invokes a model which can verify its validity by a closer examination of the regions in the scene.

The selection of a Structure model on such a crude basis, is perhaps least appropriate for the complicated PERSON model. People can alter their outlines by lying down or sitting, etc. Simpler models for static objects such as balls, bats and kennels seem to be more suitable candidates for such an approach based on gross characteristics since they are unlikely to change in appearance (barring occlusion).

In some cases it is possible to use context information as an aid in the selection of Structure models once the analysis has been partially completed. This approach may be employed at two levels:

(1) Local context level -- at this level the recognition of a particular character "triggers an expectation" for (==>) usual accompanying objects, e.g. LINUS ==> BLANKET, SCHROEDER ==> PIANO

(2) Global context level -- at this level the recognition of one object in a particular set of objects increases the expectation for the remaining objects, e.g. BASEBALL CAP ==> BASEBALL BAT, BASEBALL, BASEBALL GLOVE, ...

These "expectations" are represented by a re-ordering of eligible Structure model possibilities (essentially un-ordered initially). In combination with the context information, (and by size scaling information by this time) the Structure model selection has proved adequate in our test cases. We recognise the need for a more general initiating procedure based on low-level analysis; however, without more sensitive shape descriptors, any solution we selected would be seen as a special case routine and not a general technique.

5.2.2 The Model Hierarchy

Once a Structure model has been selected, it controls the region-by-region analysis by passing control to subordinate models as described in Chapter 3. If the correct model has been selected and there are no occlusion problems, the analysis is straightforward. As each region is recognised, an item is added to the data-base of partial results. Each item has details pertaining to the particular region and model for that match, as well as a score representing the degree of success of the match. This success scoring method was rather arbitrarily chosen as a means of collecting results of various shape tests and giving particular tests a more important role in

establishing the identity of the region. The score also plays a part in the analysis of some special model selection problems discussed in Section 5.4. The score gives an indication of the strength of the evidence for the acceptance of a model match. In case of a failure at a later stage in the analysis, a low score entry in the partial results item can indicate the point where the analysis went awry.

In Chapter 3 we described the functioning of the model hierarchy. In this chapter, we describe some of the control details that allow the model interaction. Originally, the Structure model was little more than a pointer to the Component models that form the next layer of the hierarchy. As the system was developed to handle more complicated scenes, the Structure model became more sophisticated. Later sections of this chapter will deal with this top layer of the hierarchy in more detail. At this point we describe the other three layers. The Component model, the Description model, and the Composition model are implemented as CONNIVER IF-NEEDED METHODS. This CONNIVER feature allows a procedure to be executed if its pattern matches the description of the desired item. This technique allows the program to consider several models at the appropriate moment by supplying them with similar patterns. The model hierarchy takes advantage of CONNIVER's pattern-matching facilities as well as the data-base for holding both model METHODS and assertions gained through the scene analysis.

Thus CONNIVER provides the basic control primitives to allow these models to interact quite easily. When METHODs (models) are called by their patterns, CONNIVER finds all possible matches and constructs a list of all possible METHODs to apply. This list is called a POSSIBILITY-LIST. The METHODs are tried one at a time until the appropriate model is found. A later failure due to a wrong decision causes control to jump back to the same POSSIBILITY-LIST and try the next possibility. Each model forms POSSIBILITY-LISTs of the next lower model type in the hierarchy. The pattern reduces the many possible models to a small group of eligible ones. The high-level categories shown in the Structure model are thus decomposed into their sub-parts which can be handled by a model of the appropriate type. Again take the Stucture model for PERSON as an example. HEAD is one of its parts and has Component models that describe and recognise various HEADs. We may FETCH, i.e. retrieve from the data-base, all HEAD models that match our description so far. Initially, this will be all HEAD Component models. These are invoked one by one until a successful match is made. The appropriate information is recorded and the Structure model chooses the next step.

The Component models call Description models to do the testing of their sub-parts. There are Description models for various HAIR and FACE possibilities. These have a structure similar to models of the Component class. Composition models may be called instead to build up contiguous regions into a more recognisable form. Refer to

Section 3.3.4. It will have emerged from the discussion that the deeper one goes in the hierarchy, the greater the number of CONNIVER METHODs that are required at that level.

Within this analysis framework, occlusion is a major source of difficulties. The control capabilities of the model hierarchy are insufficient to deal with occlusion related problems. To handle these problems, special control procedures have been incorporated into the system. (See system diagram in Figure 5.4).

5.3 The Troubleshooter

The Troubleshooter is the name given to the section of code that is called in to handle irregularities in the normal processing routine caused by occlusion. It has more intricate control capabilities than the models, because it must examine past decisions to find where a mistake has been made, and re-start the processing at that point. The basic mechanism is based on Fahlman's [1973] approach to a similar problem. The central notion involved here is that understanding a problem includes knowledge about what to do when the analysis goes wrong. The technique involves a class of decision making functions with special properties called Decision-makers. These functions make all the choices throughout the analysis that may later cause trouble. As a side effect, they build a list of such decision points called a Decision-trace which is available to the Troubleshooter. Along with a tag to the actual decision node, the

entries include the reason for each decision. The Troubleshooter sends error messages back to the Decision-makers. In this manner, the function that actually makes a wrong decision can correct its own mistake by choosing an alternative possibility. This method illustrates the great power available in CONNIVER that its precursors such as PLANNER lacked [Sussman and McDermott, 1972]. For example, in this situation PLANNER backs up one node at a time, exhausting all the possibilities there before retreating one more node. The Decision-trace approach records the purpose of each decision and so makes jumping back to the correct point relatively easy, i.e. it skips many of the irrelevant nodes automatically. (See the example in Section 5.4).

We use three special Decision-making functions to deal with establishing an occluding region:

(1) Decide-occluding-direction

This function suggests where the occluding region is, based on context knowledge of some common causes of occlusion. For example, HAIR is often occluded by HATs from above, FACEs occluded by ARMs from below.

(2) Decide-T-pair

Suggests T-junction pairs based on local clues. These are checked in detail by the occlusion heuristic.

(3) Decide-T-orientation

Suggests an orientation for a junction from the calculations obtained by the pre-processor.

In addition to handling problems directly related to occlusion at the T-junction level, the Troubleshooter also handles occlusion-related problems at the model level by communicating with the appropriate Structure models. There are three such problems which are handled in this manner:

(1) Dealing with three-dimensional occlusion. As we described in the chapter on occlusion (Chapter 4), the T-pairing heuristics based on two-dimensional topology may break down in some situations. In such cases true three-dimensional occlusion processing is required. A hand holding an object may occlude and be occluded at the same time. Turner [1974] had a similar problem. In his case, line labellings were inconsistent because there was no junction at the point where the line reversed direction. In our case, this inconsistency (detected by the pairing heuristic) will be allowed by the Troubleshooter. In effect, very curved objects such as hands and folds of hair are treated as special cases by the occlusion routines. (In the implementation, these are the only two cases we require. The addition of new objects to the data-base may require more.)

(2) Model-suggesting. This means finding an appropriate model for an unexpected picture region which interferes with the normal

model processing of the scene. This role of the Troubleshooter corresponds to the initial task of SCAN -- to detect a suitable Structure model for the scene. The difficulty, as always, is that the region on its own may not be easy to identify -- it may be nothing more than a small sub-part of a larger object. The solution we adopt is, once again, to depend on context information and whatever shape information is available. In the analysed scenes, this technique was sufficient, but we recognise that other scenes may be more difficult. (See discussion in Chapter 6.) For the cartoon world, this is not as much of a problem as it may seem. Except for the very complicated structure of the PERSON model, most objects are formed of only a few simple regions. Furthermore, most problems that must be dealt with by this routine are caused either by hand-held objects or much larger structures such as a table or a wall which is situated in front of the person. Local clues are used in conjunction with the global context (e.g. baseball scene) to make model suggestions.

(3) Dealing with missing model parts. Sometimes an entire sub-structure of a model is completely occluded, occluded to the extent that it is unrecognisable, or not easily located. There are several options available, depending on the situation.

The system makes every effort to find the missing region before attempting to find an "excuse" for its absence. This entails looking at the regions surrounding an occluding region for some

partial appearance of the missing part. If no appropriate region is found, the region is assumed to be completely occluded. The Troubleshooter returns control to the model analysis signalling that the missing part should be ignored, and the analysis should continue with the next model part. The Structure model may be used to indicate the next component for analysis.

This section of code has been modified to deal with the distorted scenes we have examined. A region which has been distorted beyond recognition is handled as if the missing part has been completely occluded by an unknown object. This allows the analysis to continue as outlined above, gathering as much information from the scene as possible.

To solve these occlusion problems at the model level requires information about the model system structure itself. That is, when the normal structure of region adjacency expected by the model is interrupted, we need more information to solve the problem than simply the name of the missing model part and its expected location. What is required is higher-level knowledge at the proper point in the model structure. For simple cases of occlusion, the most common solution is to restart the analysis at an adjacent region with the proper model part at the appropriate level in the model hierarchy. However, for more difficult cases and in particular for the distorted

scene analysis, higher level knowledge or meta-knowledge about the models was required. To recover from an error in the model-driven analysis, we require information about how the models fit together, and what sort of problems may arise in the analysis that may be attributed to model matching errors. Our approach has been to tackle this problem at the Component model level. There are several reason for this decision:

(1) The Component models themselves were carefully chosen to represent functional groups of regions which convey details of the scene. By restarting the analysis at this level and selecting the appropriate model, we can gain more high-level information about the scene.

(2) The Description and Composition models function best when there is strong context information to signal their invocation. The occlusion or distortion that caused the current failure has destroyed the relational information which guides the analysis. The Component model provides this necessary relational information, so it seems appropriate to use this level as the new starting point.

(3) On purely practical grounds: there are dozens of shape models at the Description and Composition level, while only a few Component classes (for PERSON). By restarting at the Component model level, we reduce the amount of information that must be coded. By tentatively accepting the previous interrupted Component model (if there is sufficient cause -- i.e. successful Description model applications) the high level

information may be unaffected. For example, by recognising the BLOUSE part of the TORSO model, before missing the SKIRT, we extract all the necessary information, i.e. the direction of the TORSO and confirmation of the female identity of the person. The missing SKIRT supplies very little information on its own. If instead it was the BLOUSE that was missing, then this information would be lost. The SKIRT would be unidentified since the TORSO model was aborted, but as we have stated, there is little to be gained from that Description model in any case.

The meta-knowledge we have considered for these scenes is:

(1) The relationship between the suggestor (i.e. the successful Component model) and the adjacent region-to-Component model match.

(2) The application of knowledge about model similarities to preserve the effects of a partially successful model application and suggest a more appropriate model.

Again, to give a more detailed explanation of the way the control mechanisms function we present some examples.

5.4 Examples

The control examples we present illustrate the complex control issues mentioned above rather than the simpler issues (such as model selection by pattern invocation). The discussion concentrates on the

system behaviour when it is confronted with an unexpected problem -- a contradiction or evidence of an incorrect decision. Details of system performance on deliberately distorted scenes is also presented.

5.4.1 Selection of T-junction Orientation

To begin this section of examples we present an example of T-junction orientation by the Troubleshooter. In addition to clarifying some of the previous occlusion examples, this should illustrate the power of the Troubleshooter in controlling the analysis. In Figure 5.6a the orientation of both junctions T0227 and T0228 is ambiguous. This is rare -- usually one of the pair will have a unique interpretation thereby forcing a complementary interpretation on the other. Figures 5.6b and 5.6c show the two possible orientations for each of the junctions.

The occlusion analysis of this scene was discussed in Chapter 4. In this section we describe the control procedures involved. Refer to that chapter for the details. Briefly, the first decision that must be made concerns the selection of a pair of T-junctions to show that region R0189 (the face) is occluded. The Troubleshooter is called upon to solve the problem. The first step is to decide on the direction of the occlusion, since there are so many junctions surrounding R0189. Based on the current model's suspicion that R0189 is a FACE, the Troubleshooter (in its decide-occlusion-direction

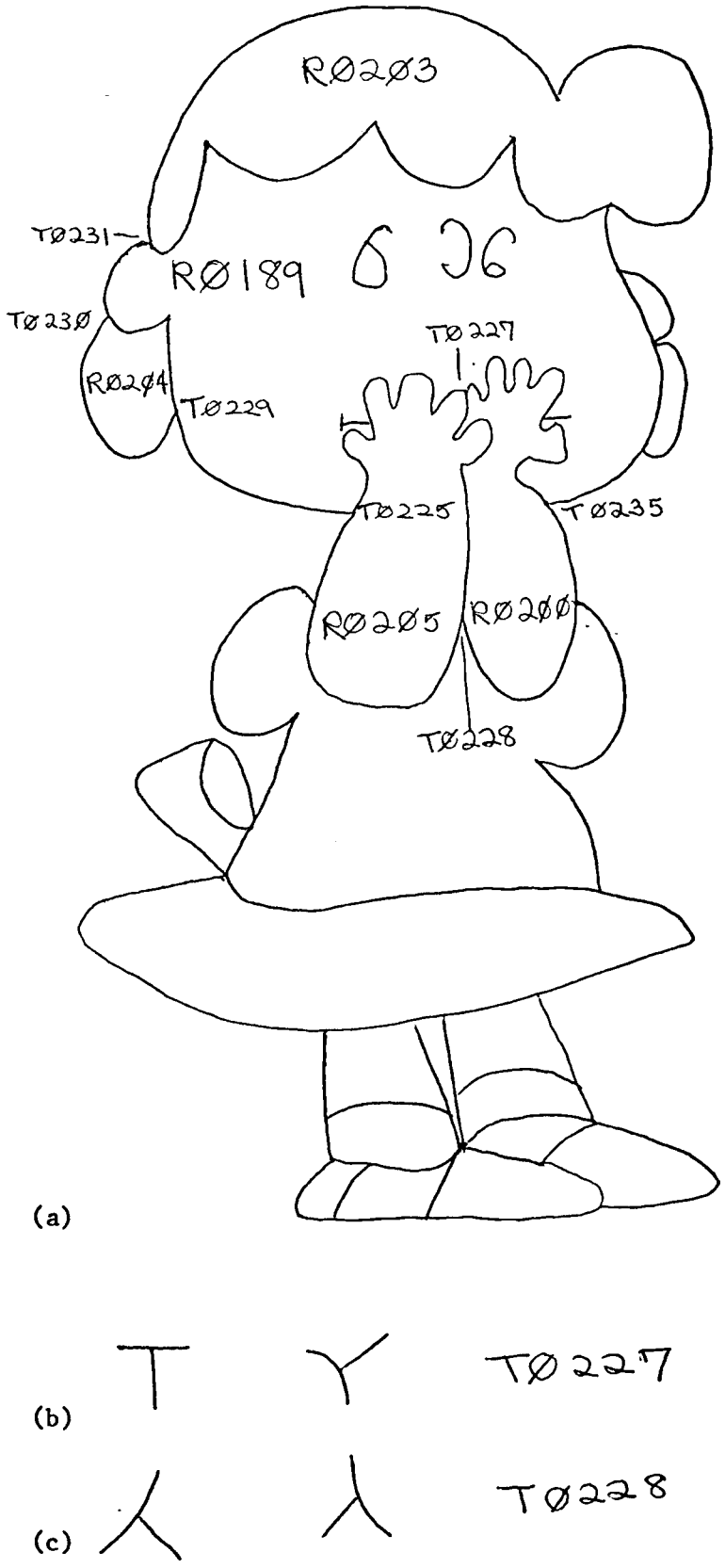

RØ2Ø3
TØ231
RØ189
TØ23Ø
TØ227
RØ2Ø4
TØ229
TØ225
TØ235
RØ2Ø5
RØ2ØØ
TØ228
(a)
(b)
TØ227
(c)
TØ228

Figure 5.6

mode) refers to a table of helpful "common-sense knowledge" and returns BELOW(*). This decision is added to the Decision-trace list. The entry incorporates a tag back to the decision function so it can later reconsider this choice if necessary. Figure 5.7 contains a diagram of the decision trace for this partial analysis of the scene. Using this directional information, the set of candidate junctions for pairing is cut in half. The Troubleshooter in its decide-T-pair mode must select an appropriate pair from this set according to the following criteria:

(1) both junctions must have the same region "above" their bars;

(2) the junctions must have a common region "between" them. (R0189 in this case.)

The orientation of ambiguous junctions is chosen to meet these criteria. As we have already mentioned, each ambiguous junction has an ordered set of possible orientations associated with it. The Decide-orientation function originally selects the first of these possibilities, and on subsequent failures returns successive possibilities one at a time. (Refer to Figure 5.8 for an explanation of junction orientations.) In this case, the original selection of

---

(*) This table contains some standard hints about usual instances of occlusion. If there is no appropriate entry or the advice fails, the system uses local junction information.

(1) Decide-occlusion-direction for R0189 (FACE)
Returns: BELOW

(2) Decide-orientation for T0227
Orientation 1 -- rejected
Orientation 2
(Note: Orientation 1 is disallowed since R0189 the region we are trying to prove occluded is in the occluding region slot for this orientation)

(3) Decide-junction-pairing for R0189 from BELOW
Returns: T0225 and T0227
(Note: After an orientation has been chosen, the junction no longer appears ambiguous to the system)

(4) Decide-orientation for T0235
Returns: Orientation 2

(5) Decide-orientation for T0228
Orientation 3 -- rejected (as in Node 2)
Returns: Orientation 1

(6) Decide-junction-pairing for R0200
Returns: T0227 and T0228

(7) Decide orientation for T0236
Returns: Orientation 1

(8) Decide-junction-pairing for R0189 from BELOW
Returns: T0225 and T0235

(9) Decide-orientation for T0229
Returns: Orientation 3

(10) Decide-orientation for T0231
Returns: Orientation 3

(11) Decide-junction-pairing for T0231
Returns: T0231 and T0232

Alternative Analysis

(9) Decide-orientation for T0229
Returns: Orientation 1

*(10) Decide-orientation for T0231
Returns: Orientation 2

*(11) Decide-junction-pairing-3D for R0204
Returns T0230 and T0231

*(12) Decide-junction-pairing T0229
Returns: T0229 and T0232

Figure 5.7

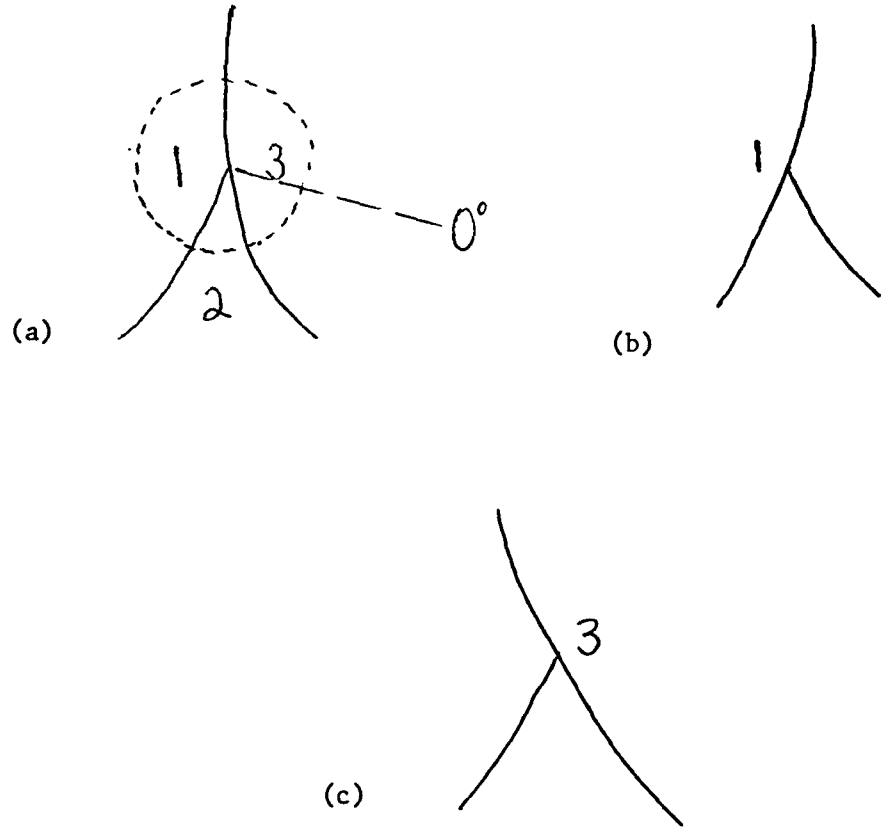


Figure 5.8 Orientation numbering: The regions surrounding a junction are numbered in a counter-clockwise direction starting at the first line leaving the junction with an angle greater than 0 degrees. The orientation numberings refer to the number of the region above the T-bars. For the case illustrated, the possibilities would be: 1 and 3.

orientation 1 for junction T0227 is rejected by the junction pairing function. The error message:

(WRONG-ORIENT T0227 ...)

is presented to the Troubleshooter; the correct decision node is found; and the decision is reconsidered. The alternative orientation for this junction allows the pairing of T0225 and T0227. These decisions (pairing and orientation) contribute two more entries to the Decision-trace. (Refer to Figures 5.6 and 5.7). In this case, by choosing orientation 2 for junction T0227 (the less likely of the possibilities on local angle information), the pairing of T0225 and T0227 is allowed. These decisions contribute two more entries to the Decision-trace. As the tracing proceeds along the stem of T0227, it reaches junction T0235, an ambiguous junction. There is no overriding global information to augment the local angle information, so the most likely local orientation is chosen and the trace fails. (Notice that pairing a junction with another provides global information to influence the selection of the orientation, but tracing itself is the collection of local evidence and so does not normally overturn the locally derived orientation decision). Any of the following may be the underlying cause of such a trace failure:

(1) The wrong pairing has been selected (another one exists)

(2) The wrong orientation for a junction has been selected

(3) Because of multiple-occlusion the wrong pair was selected

(4) Three-dimensional analysis is needed

(5) There is no occlusion (in the specified direction)

To cope with this problem the Troubleshooter passes the failure information:

(TRACE-FAIL T0225 T0227 T0235)

back up the Decision-trace path expecting each Decision-function to determine if it caused the error, and if so, to try to correct the mistake. The first two junctions refer to the proposed pairing, the third to the trace failure point. In this case, no other pairing (Node 3 in Figure 5.7) is possible and no other orientation for T0227 (Node 2) is appropriate. The occlusion direction decision is not a direct cause of this failure so it is not reconsidered. The nature of the failure is consistant with an instance of multi-layered occlusion. This sets up a new pairing goal. Region R0200 is suspected of being occluded since it is the common region to junctions T0227 and T0235 (in addition to R0189). No occlusion direction information is sought since the identity of R0200 is not known. The only pairing possible is T0227 and T0228 forcing the orientation of T0228 to be orientation 1. The trace succeeds (choosing the first orientation of T0236 in passing). Region R0205 occludes R0200.

The next step in the multi-occlusion analysis is to reconsider the original occlusion problem with the new super-region structure S0300 formed of R0205 and R0200. Now junctions T0225 and T0235 may be paired. This node is added to the trace like the others; it does not replace node 3. The trace of this pairing causes junctions T0229 and

T0231 to be oriented. The trace "stalls" at this junction. A pair for T0231 must be found. This is easy since there is only one other junction on the border of R0203. The trace can now be completed. Region R0189 is occluded by R0203 and S0300. The model analysis may now continue.

As we mentioned in Chapter 4, if the other interpretation of junction T0229 had been selected, the occlusion analysis would return a very different result. The difference comes at node 9. The trace stalls at this junction instead of passing around it. There is no possible pairing for this junction, so once again the Troubleshooter is called in to solve the problem. As before, it works its way through the possible reasons for failure by passing the error message back to the relevant procedures. This time multiple-occlusion must be ruled out due to the region configuration. The last resort before rejecting occlusion altogether is to try the three-dimensional occlusion solution (see Chapter 4). This uses a special section of code to deal with this particular problem. The nodes marked with an asterisk in Figure 5.7 mark decisions involved in this three-dimensional solution. The first step is to establish that region R0204 is sufficiently curved to justify the application of these techniques. Only the HAIR and HAND-ARM regions qualify for this distinction. Region R0204 is unidentified, so the auxiliary HAIR models (added by the HAIR Description model) and ARM model are applied. HAIR succeeds so the three-dimensional techniques may be used. The alternative orientation of T0231 is used to allow special three-dimensional

pairing configuration.

The HAIR regions R0203 and R0204 are now both interpreted as being occluded by R0189 (at the EAR). A super-region S0301 is formed to represent the HAIR. Finally the stalled trace at junction T0229 can continue. T0232 is paired with T0229 (in the normal way) and the trace is completed. This special technique illustrates the power of the Troubleshooter to alter the control flow of the system to handle special cases as well as the more routine problems that occur in the scene analysis. As stated in Chapter 4 this second interpretation of the scene has been forced -- we altered the junction orientation data to force this result and illustrate the immense scope of the Troubleshooter.

It should be clear that an occlusion trace is not as straightforward as it may have appeared in Chapter 4. Many decisions must be made at the junction level. The selection of the proper pair of junctions usually involves orientation decisions as well. Due to the complications involved, occlusion processing is kept to a minimum by incorporating knowledge of the expected cases into the model system.

5.4.2 Preservation of Knowledge/Model Selection

The second example is taken from Figure 5.1 reproduced here as Figure 5.9. We join the analysis as the Description model for SKIRT is applied to R0315 (the larger portion of the baseball bat). The match

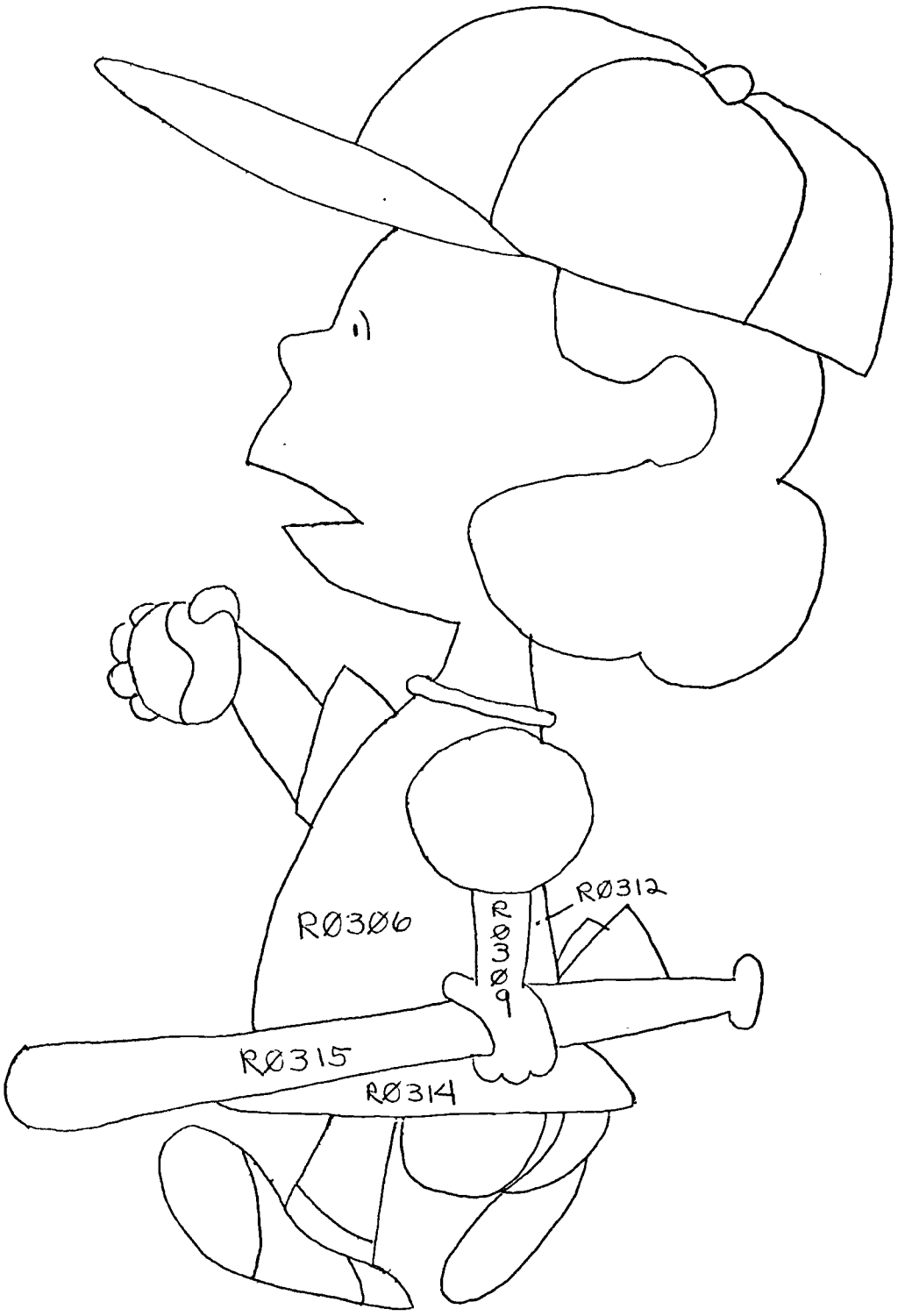


Figure 5.9

succeeds although the region is smaller than expected. The TORSO model is completed. (Note: as described in Chapter 3, ARM region R0309 was identified during the search for the small part of the blouse, region R0312.)

The next step involves the analysis of the ARM region R0304. The mistaken analysis of region R0315 is not detected until the LEGS Component model is invoked to find the legs below the TORSO, which erroneously includes region R0315. The real SKIRT region R0314 is eventually matched to the Description models for LEG and fails. Once again the Troubleshooter is called in.

We could backtrack through all the (correct) decisions until we reach the error point, correct the mistake, and then make all the same decisions over again. Obviously, we would prefer a more efficient method -- one which preserves all the correct decisions, changing only the incorrect ones. For example, the analysis of the HAND-ARM occluded by the baseball is completely independent from the BAT/SKIRT error. There should be no need to interfere with those decisions.

The two underlying considerations are:

(1) The need to discover the error

(2) The need to correct the error in isolation.

The former is the much more difficult problem. With the correct

solution in mind, the necessary deductions are obvious:

(1) If R0314 is not part of the LEGS then perhaps R0315 is not part of the TORSO. (The TORSO model is the "suggestor" for the location of the LEGS. Since LEGS must be BELOW the TORSO the lowest region is most suspect.)

(2) If R0315 is not part of the TORSO (SKIRT) then it must be something else. (The low confidence score corroborates this.)

But how far back should such logic be carried? If R0315 is not the SKIRT then perhaps R0306 is not part of the BLOUSE, etc. For this scene a third clue provides the best information.

(3) R0315 is adjacent to R0309 the ARM, so it might be a hand-held object

Most occlusion (barring self-occlusion) is caused by hand-held objects. The system treats ARMS as very special objects, and this third clue is the vital piece of evidence that guides the system to the error point.

This is the logic that governs the behaviour of the Troubleshooter, but it does not explain how it functions. To handle these sorts of problems requires coded knowledge of the relationships between the model parts. We have decided to code this knowledge at the Component model level, i.e. the Structure model contains information concerning the relationships between its member Components which it supplies to the Troubleshooter.

In addition, the Troubleshooter can access the environment of each Component model's existence through the use of function closures. In simple terms, the Troubleshooter can evaluate expressions in the environment of specially selected points in the history of the processing of the scene. By correcting the data-base in the past environment, the future processing is allowed to continue as if there had never been an error. The large degree of independence between the Component models allows us to use this technique. Otherwise patches to the data-base might cause contradictions at a later stage in the processing. With this background information we proceed to outline the system's performance. The Structure model informs the Troubleshooter of its dilemma:

(TROUBLESHOOT '(MISSING-MODEL-PART LEGS (BELOW TORSO)))

The first list element indicates the type of problem; the second, the missing part; and the third, its expected location. The data-base entry for TORSO is in terms of its super-region name, S0400. Its lowest subregion component, R0315 is isolated and its model entry is sought:

(DESCRIPTION-MODEL SKIRT R0315 ...

Model-method closures are not preserved (as Decision functions are) so the Description model itself cannot re-consider its decision. In any case, it is occlusion that has caused the error -- in the same environment the model would make the same error all over again.

The Troubleshooter's first step is to find another model to match region R0315. From the data-base it discovers that R0309, an

adjacent region, has been identified as an ARM. The Troubleshooter proceeds on the assumption that R0315 is some other hand-held object (see discussion of this below). The baseball context has placed baseball equipment at the top of the list of possibilities.

The subsequent steps in the correction procedure are outlined below:

1) A new CONTEXT is created (to preserve all existing results).
2) The identification of R0315 is removed from the data-base.
3) A new model is found for the region (baseball bat).
4) The TORSO Component model is called again. Most of its work has been already done. Instead of finding only model methods, the FETCH's return the existing data-base entries first. The models are ignored. The only analysis needed is for the missing SKIRT region.
5) As a final step these results are added to the old environment (in case of future problems). The old entries are not removed, the new ones added last, are the first to be found.
6) The Troubleshooter returns control to the Structure model, with the error corrected.

Since this is the only such example we have considered, we do not wish to make too strong a claim for the solution technique. There are still some issues to consider, such as how hard to try to force the success of the model match. In this particular case, there was very strong evidence that the PERSON model was appropriate. The

HEAD, ARMs, and TORSO Component models had returned successful results (despite the error in matching the SKIRT to the baseball bat). If only the HEAD had succeeded with a low confidence score then the evidence would suggest that the wrong Structure model had been applied. It is very hard to determine at what point to draw the line.

The crucial factors appear to be:

(1) The number of successful sub-models and some measure of their complexity;

(2) The strength of the confidence scores for each match.

This is an area that requires more investigation before any conclusive results can be determined.

The matching of the baseball bat model to R0315 and the related occlusion processing is another rather complicated problem. Refer to Figure 5.10. The bat is occluded causing problems in selecting the appropriate model. The solution we adopt is to use a very rough guide to select model possibilities, and let the application of the model determine whether or not it is appropriate. Only the enclosing rectangle is used as a guide to select the model. The overall width is unaltered by the occlusion, although the length is reduced by a third. The ratio between length and width is still quite large (5:1) and this is used as a basis for selection.

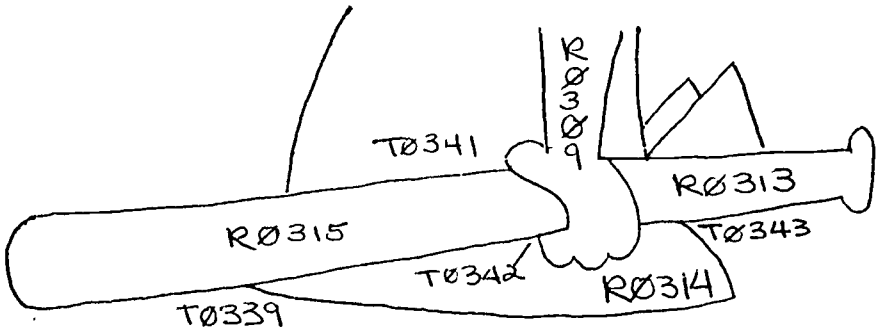


Figure 5.10

The single region R0315 does not meet the baseball bat Description model's shape criteria, so the occlusion routines are invoked. This is another three-dimensional occlusion case. There are no ambiguous T-junctions on the border of R0315. The two junctions at the boundary of R0309 and R0315 are not in the proper configuration for pairing. T0341 indicates that R0309 occludes R0315; T0342 implies the opposite. A call to the Troubleshooter is made. This occlusion example differs from the previous ones since region R0309 has been recognised as an unoccluded ARM. The important factor is that R0309 as an ARM can exhibit the three-dimensional characteristics which frustrate the occlusion heuristics. The three-dimensional occlusion information supplied by the Troubleshooter is passed back to the model (and data-base):

(OCCLUDED-BY R0309 R0315)

(OCCLUDED-BY R0315 R0309)

The remaining region, R0313, is located on the other side of the ARM using the region relation knowledge of the data-base under the control of the model:

((RIGHT R0309* R0315) (RIGHT R0313 R0309*)

(BELOW R0314 R0309*) (RIGHT R0309* R0306*)

(RIGHT R0312* R0309*))

*already identified

The only two unidentified regions which are adjacent to the ARM region R0309 are R0313 and R0314. Only R0313 maintains the proper directional relationships:

R0315 is to the right of R0309 (and occluded)

R0313 is to the left of R0309

The occlusion routines (without the three-dimensional knowledge) determine that R0309 also occludes R0313. The union of R0313 and R0315, super-region S0415, is identified as the baseball bat.

On its second incarnation the TORSO Component model discovers the baseball bat where it expected the SKIRT. This external object is interpreted as a possible occluding object by the Troubleshooter, and the adjacent unidentified region R0314 is tested by the SKIRT Description model. The occlusion routines pair junctions T0339 and T0343 to demonstrate that R0314 is occluded by the bat. The Description model succeeds and the TORSO Component model is complete for a second time. The analysis of the ARM region R0309, by the ARM Component model requires no new processing. The call:

(FETCH '(COMPONENT-MODEL ARM ...

simply returns the data item first -- the IF-NEEDED METHOD model is not applied. The Structure model finally applies the call to the LEGS Component model which this time succeeds.

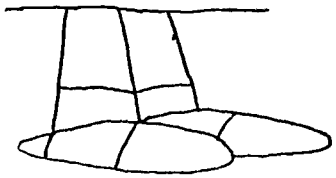
In the previous scene, occlusion separated a functional group of regions into two or more parts. The ARM broke the bat into two; and the bat separated the SKIRT from the TORSO. In both these cases, the isolated region was found through the use of adjacency information

aided by directional knowledge. Naturally this technique may be refined. For instance, the baseball bat handle could have been located by the superposition of the bounding rectangle followed by the investigation of the enclosed regions by the model. In our study such techniques were not needed.

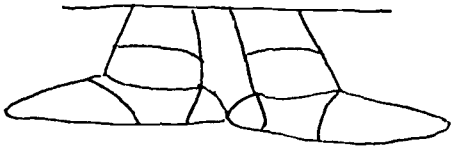
As a final example in this section, we illustrate the application of knowledge of model similarities and partial results to select the correct model and perhaps preserve part of the aborted analysis of the wrong model. For this example, we ran only a partial analysis of the previous scene (Figure 5.9) using the LEGS Component models on the regions below R0314. The only global information available was that R0314 was a SKIRT, and that who was bound to LUCY. The directional information which eliminates too many of the wrong models was omitted in order to illustrate the desired interaction between similar models. The approach taken here is similar to the restarting of the TORSO model described in the previous example, this time through the interaction of the Component model and Structure model rather than the Troubleshooter and Structure model.

For this experiment the POSSIBILITY-LIST of LEGS Component model was ordered as follows (refer to Figure 5.11):

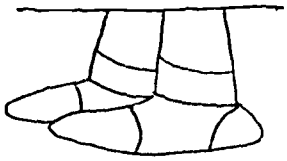
(a)



(b)



(c)



(d)

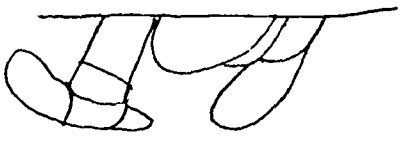


Figure 5.11

	Direction	Type
1	RIGHT	STANDING
2	FRONT	STANDING
3	LEFT	STANDING
4	RIGHT	WALKING
5	LEFT	WALKING

•

•

•

The invocation of the first model succeeds to match the LIMB Description model to R0320 and R0321. Then the SHOE Description model fails because the shoe is facing the wrong way. The failure of this right-facing shoe removes all the other right-facing possibilities from the pending POSSIBILITY-LIST. The next model for FRONT/STANDING LEGS has access to the successful parts of the first model, that is, the FETCH to get the Desciption model for LIMB retrieves the results of the aborted RIGHT/STANDING model. The SHOE analysis of the first leg succeeds, but the second LIMB fails. Using knowledge of related models (see below), the FRONT/STANDING model places the (correct) LEFT/WALKING model at the head of the POSSIBILITY-LIST. This model completes the analysis, again using the partial results of the previous model applications rather than re-applying the Description and Composition models. The clues that are used by the model are:

(1) The successful LIMB/SHOE analysis

(2) The fact that there was no occlusion

(3) The failure of the STANDING model

In this example, knowledge of similarities between the family of LEGS Component models is used to home in on the appropriate model and preserve partial results. Since the structure of the leg regions is so similar for the variety of views, the Component model contains information about the shared sub-structures. The failure of one model may provide good evidence for another. To use such techniques for less closely related models is much more difficult. See Chapter 6 for further discussion of this problem.

5.4.3 Distortion of the Input Scene

The system as it was originally designed would grind to a halt if some model part could not be found (or its absence could not be explained by an instance of occlusion). The control structure has been modified to allow the system to continue the analysis, returning as much information as possible. In this version, the Component models allow the acceptance of incomplete models and the Structure model's information concerning relationships between component parts is more loosely interpreted by the Troubleshooter to allow it to "skip over" troublesome regions.

Figure 5.12 shows a scene of LUCY with a distorted SKIRT. The

Figure 5.12

analysis proceeds without error until the SKIRT Description model is matched to region R0249. This region is too large to match the model and is rejected. The Troubleshooter is called in to solve the problem. It searches through the data-base for suitable models for this region. In the first trial there were no suitable models for R0249 (but see below). This scene differs from the last one in two ways:

(1) The model match error occurs within the TORSO Component model rather than at the TORSO/LEGS boundary.

(2) There is no low confidence score to indicate an error.

Instead of aborting the analysis, the Troubleshooter attempts to continue the PERSON analysis at another point. The region, R0249, is marked as unidentified in the data-base, and the Description model entry for SKIRT indicates that it was not located:

(DESCRIPTION-MODEL UNIDENTIFIED R0249 ...)

Using the Structure model's table of relationships it successfully directs the LEGS Component model to try BELOW region R0249. The resulting Component level description is not affected by the missing model part.

The addition of an applicable model for R0249 to the data-base does not help -- it makes things worse. R0249 is matched to a table-top. As such it should provide evidence for occlusion of the SKIRT region,

instead the T-junctions indicate that the BLOUSE occludes the TABLE-TOP. The table model must be rejected since the supports for the TABLE-TOP are missing. So once again the Troubleshooter relies on Structure model information to find the LEGS.

One may concoct alternative interpretations for this scene. Region R0249 might have a hole in it, or it might be an oddly shaped piece of wood carried by LUCY. The significance of R0249 is not really the issue. This scene demonstrates that the analysis can cope with problem regions by labelling them as unidentified and continuing the recognition process gaining as much information as possible from the scene.

CHAPTER 6

PROBLEMS/SOLUTIONS/ALTERNATIVE IDEAS

The previous chapters explained the mechanism of the system and presented details of the analysis techniques. In this chapter we discuss some reasons for the present approach and the limitations of the solution.

6.1 Models

As we have mentioned before, the major difficulty in this domain has been description of shape to allow recognition. Various methods were tried, (e.g. transforms, picture-grammars, feature-extraction) but none were satisfactory. The causes of the failure were due to the following problems:

(1) Most methods of shape desciption are too exact. Similar shapes were not recognised.

(2) In the cartoon world, exact shapes are usually not as important as relationships between parts. The cartoonist may allow the exact shapes to vary within some range, relying on the relationships to convey the appropriate interpretation. So the "precise" models mentioned above are even less appropriate in the cartoon world than in more natural domains.

(3) Three-dimensional information for complex curved objects was difficult to include in the shape model. Again, this was complicated by cartoon conventions. Different views of the same object cannot be merged to form a three-dimensional model. The

representation of the image (and the model system) relies on a limited selection of representational views (frames) to convey various poses.

(4) Occlusion altered the perceived shapes of the regions.

(5) Matching isolated regions to models, i.e. without context information, was in most cases impossible, although some regions, such as faces could be recognised. Consider the variety of interpretations for a small rectangle: CALF, SOCK, BOW, ...

Some of these problems may be attributed to the domain chosen for this study. Baumgart [1973] has designed a system which extracts three-dimensional shape information from real objects using depth information to form a polyhedral model. However, recognition models (rather than display models) are more complicated. Certainly, if depth information had been available occlusion would be easier to detect; however, the problem of recognising the partially hidden portion would still remain.

Perhaps it is our lack of depth information which adds so much complexity to the system. The only clues to depth information come from the model driving the analysis. The T-junction analysis provides a small degree of local information -- but this is not totally reliable. Contrast this with polyhedral analysis where as Waltz [1972] has shown, the local interpretations of junctions can in many cases, uniquely solve the whole scene. In real world scenes,

laser ranging techniques can segment objects into generalised cylinders. In this cartoon universe, the division into regions can act as camouflage -- disguising the depth relationships by breaking up the scene into a two-dimensional network of neighbouring shapes. Just as local information alone is not sufficient to find the faces of the block in Figure 6.1, one must use high-level knowledge to distinguish the unoccluded regions (and their associated junctions) from the occluded ones in the cartoon scenes.

The representation we finally selected based on gross shape characteristics and augmented by special feature tests is adequate for the selection of scenes we analysed. The addition of models to the data-base similar to existing models may require some tuning of the tests to differentiate between two similar model parts. (Information concerning relationships between sub-parts allows us to distinguish similar shapes such as SOCKS from CALVES. However, the introduction of a new character to the system may cause problems if the new hair style is similar to an existing hair model. Relational information cannot be applied to differentiate between possible candidates for the same model sub-part.) The shape models are probably the weakest part of the system. A more complete shape description technique would diminish the likelihood of tuning the system as new models are added. The improvement of the descriptive techniques is a problem for the future.

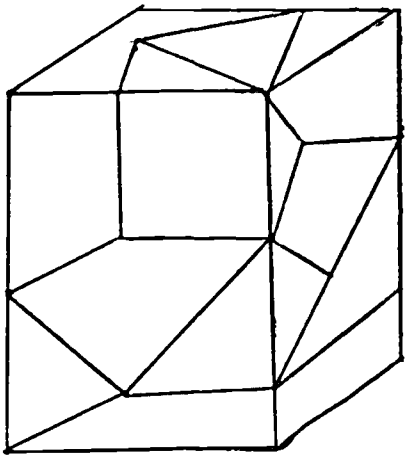


Figure 6.1 From Winston [1975]

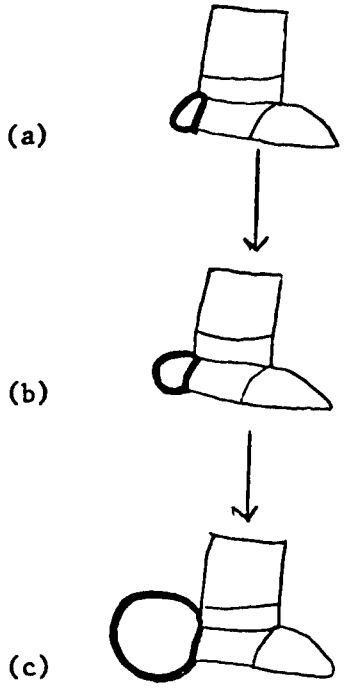


Figure 6.2 Distorted heel

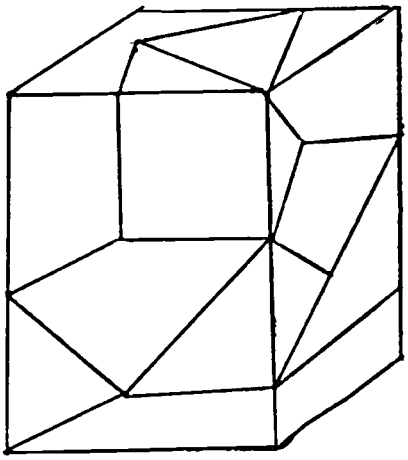


Figure 6.1 From Winston [1975]

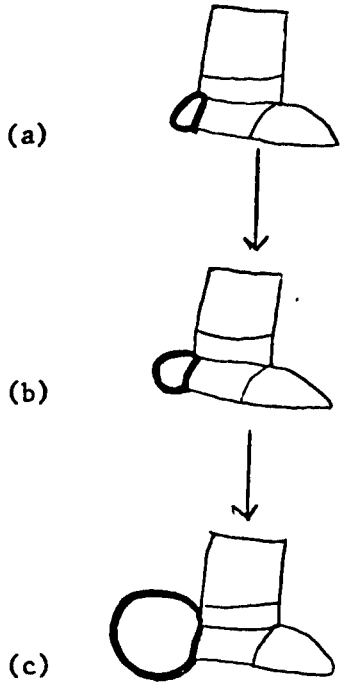


Figure 6.2 Distorted heel

We have shown that by incorporating relational information into the model system we can compensate for the lack of definite shape information. Our experience with the system on Peanuts scenes and our study of other (unanalysed) scenes shows that a fairly high degree of reliability can be achieved despite the problems mentioned above. The hierarchical structure of the system places less emphasis on the shapes of those regions corresponding to less significant parts of the model, relying instead on the inter-relationships. This technique is well-suited to this particular domain and may also have applications in other domains. To illustrate this point we include an example of systematic distortion of the HEEL region of LUCY's shoe corresponding to the lowest part of the model hierarchy, the Composition model. Figure 6.2a is a normal scene; in Figure 6.2b the HEEL region has tripled in area; and finally in Figure 6.2c it is five times its original size. The system recognised the first distortion as a legitimate HEEL, but rejected the second one. This reflects our own interpretation of the scene. The distortion will not be noticed by a cursory glance at Figure 6.2b but it stands out in Figure 6.2c.

The hierarchy achieves this result by setting a very low threshold of acceptance for the Description models. The relational information of the Composition model is the determining factor. In Figure 6.2b the match of region R0251 to the HEEL model achieves a score of 70. The threshold is set to 50, so the match is accepted. The Composition model for SHOE has a higher threshold: 70. When the sub-regions are

gathered together, their scores are averaged. The composite score is 90, well above the threshold. In Figure 6.2c the HEEL fails all tests. The occlusion routines called in to show that R0251 occludes the HEEL fail because of the junction configuration. There are no appropriate models to match the region. The SHOE Description model fails. If the analysis proceeds, the effect of the low score becomes less pronounced in the higher portions of the hierarchy.

One may interpret these scores as a reflection of the confidence in the appropriateness of the model. The success of the individual lower-level models produce a global context effect -- reinforcing the validity of the higher-level model. The significance of the failure of one small portion of the model may be overshadowed by the success of the remaining parts. This effect is very different from the Waltz filtering technique. Waltz's system depends on complete and accurate information throughout the scene. A misinterpretation of one junction (due to bad data) would spread disaster throughout the system. In our case we have sought to minimise the effects of poor shape descriptions (not bad data) by imposing a hierarchical structure on the significance of each part.

Another important feature of the hierarchy is the structure it imposes on the relationships between parts, both at the region level and at the Component model level, i.e. functional groups of regions. In this cartoon world these relationships between regions are often

more important than the shapes especially for the less significant parts of the scene. The cartoonist uses detailed shapes to emphasise parts of the scene such as faces, while using coarse shapes to complete the drawing, e.g. legs and shoes. The importance of such relationships is not limited to the interpretation of cartoons, it is a necessary part in any vision system. There may be some counterpart in the human vision system since we can often recognise people at a distance using not detailed information such as facial features, but relational attributes such as the way they walk. Unfortunately, in this system the role of the relationships between parts has had to be emphasised to compensate for problems in characterising shapes.

The structure of the hierarchy also plays an important part in dealing with the flexible nature of the cartoon bodies. The Component model level effectively isolates the rigid portions of the body. The exact inter-relationships of the limbs, head, and torso is determined by adjacency information. A minor exception from this strategy was the grouping of both legs as one unit. This was a convenient method of solving two problems:

(1) The top-level description of the scene depends on information based on both legs. The walking configuration is reflected as one bent leg and one straight one.

(2) The leg self-occlusion that is almost always evident is easier to handle at the model level than at the junction level. By coding this occlusion as standard, the analysis is much less complicated. (See Section 6.2 below).

This structuring also allowed the encoding of the three-dimensional information as alternative sub-models (frames). By selecting a consistent set of Component models, the system can represent numerous body configurations expressing a variety of characters, from different angles and in different body attitudes.

The hierarchy has been adequate to deal with the examples presented to the system. However, there is a danger that if more models are added to the system, particularly models that are very similar to existing ones, the system will be confused. The present system structure places the responsibility of knowing about similar models (which might be confused) within the model itself.

Such knowledge may be applied to eliminate similar models from a POSSIBILITY-LIST based on one failure; or to preserve part of partially successful analysis, passing on the information to a similar model. The current system uses such facilities at the Component model level. Whether or not this technique could be extended at a higher level depends to a large degree on the particular set of objects in the universe. The Structure model would have to be modified to allow this interaction to take account of the structural similarities between specific models. The main obstacle, as ever, is the difficulty in using bottom-up analysis in this domain. Taking the baseball scene as an example, we would have liked to have jumped to the baseball bat model using knowledge of model

similarities. However, a baseball bat and a skirt are usually very dissimilar in shape; the occlusion in this scene was the cause of the model mismatch. Once again, the combination of poor shape descriptors and occlusion frustrates any attempts to use low-level data to drive the analysis.

Finally, there is the problem of a very large data-base of models. One problem in using a procedural representation for the model hierarchy is that the addition of a large set of models requires a vast amount of space. One way to reduce the glut of models in the data-base would be to allow the higher level models to control the addition and removal of the model methods from the data-base. This is the frames idea being used at a higher level. The Structure models themselves could be grouped according to size and/or context and swapped out with their sub-models when they were inappropriate.

6.2 Occlusion

Occlusion effects the processing of the scene in two ways:

(1) It alters the shapes of regions;

(2) It changes the inter-relationships between regions.

In other words, occlusion destroys both types of information which are used by the models for recognition. The types of occlusion handled by the system fall into three categories, each with a different solution technique:

(1) Expected self-occlusion

(2) Occlusion solvable in two dimensions

(3) Occlusion requiring a three-dimensional explanation

Our naive assumption that sophisticated junction heuristics (category 2) could account for all the occlusion problems was destroyed in the analysis of the first scene. The junction analysis proved to be much more complicated than expected, involving orientation of ambiguous junctions (including multi-junctions), explanation of contradictory evidence due to the three-dimensional occlusion problem, interference by junctions caused by abuttal and not occlusion, and the irregularity of the lines.

These difficulties reinforced our belief that an examination of the junctions should only be used when model failure indicated that occlusion might be present; i.e. as a last resort to allow a weak match to succeed, rather than a general purpose low-level technique to discover all instances of occlusion.

The junction pairing technique, although not completely reliable is a valuable tool in the discovery of occlusion relationships. By supplementing this method by higher-level techniques to handle the troublesome cases (categories 1 and 3), the pairing technique has successfully contributed to the solution of occlusion-related problems.

The need for some three-dimensional occlusion techniques to account for gripping hands, etc. in this "two-and-a-half" dimensional representation is evident (see Chapter 4). The need to modify the model hierarchy to reflect standard instances of occlusion may be less obvious. There are two levels of occlusion information encoded in this way:

(1) Standard occlusion within a functional group of regions, e.g. a shoe occluding a sock occluding a leg. The standard appearance of a sock is in its occluded state: a rectangle. To invoke occlusion routines for every such appearance of a sock would be a rather meaningless exercise. (There is no reason not to include two models for an object: an occluded and unoccluded instance, i.e. a sock on a person and a Christmas stocking.)

(2) Occlusion caused by the interaction between functional groups of regions, e.g. one leg occluding another or an arm occluding a torso.

This second model alteration was not necessary for the analysis of the first scene. However, as other scenes were considered, it emerged as a very useful technique. Not only could it handle certain standard cases of occlusion much more efficiently than the cumbersome T-junction heuristics, but it also could be used to identify small regions that were isolated from the main portion of the object by an intervening occluding region. For example, region R0312 in Figure 6.3 is part of LUCY's blouse that has been cut off from region R0306. By incorporating this knowledge of a typical instance of a vertical arm occluding a torso in the model, identification is simplified. In

Figure 6.3

some cases, due to the weakness of the shape models, the occlusion of a region may not be noticed. Without special knowledge of such problems, the isolated region would be virtually impossible to identify. This may be viewed as an additional means of incorporating three-dimensional information into the model hierarchy.

As a final point of discussion, we return to the problems of the junction pairing heuristics. Despite its inherent faults it has proved to be useful and remarkably successful. By modifying the model system (as described above) to account for particularly awkward cases, only the relatively simple cases of occlusion (consistent with the design criteria) were referred to the pairing heuristic. Orienting curved junctions proved to be less of a problem than anticipated. Part of the success was due to context information (see Chapter 4). In the analysis of the baseball bat, the arrow-shaped junction T0346 (Figure 6.4) was unambiguously interpreted as a T-junction in the required orientation to allow the pairing. Perhaps even more amazing is the analysis of the multi-junction at the intersection of the blouse, skirt and bow K0247 (Figure 6.5) which returned an ambiguous result with the "correct" interpretation given a higher priority. These decisions were made on the basis of local information alone using a few points of each line to calculate angle differences around the junction. Most junctions received unique interpretations.

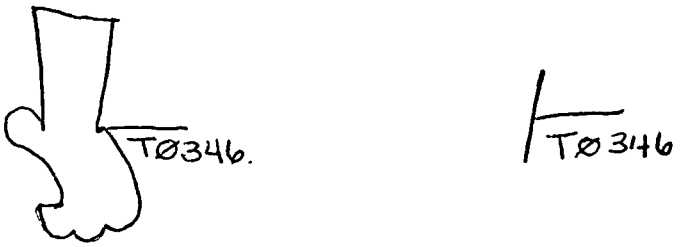


Figure 6.4

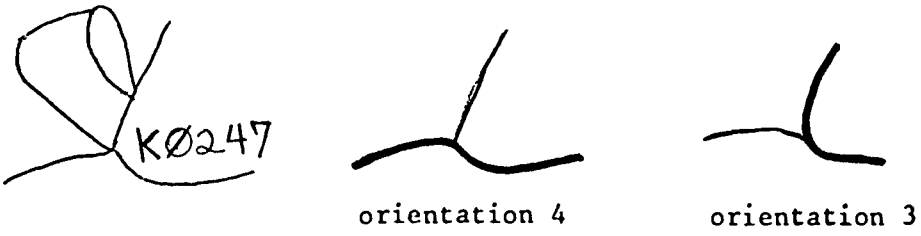


Figure 6.5 Bow from scene of Lucy with her hands in front of her face.

Figure 6.6 The orientations for T-junctions T0215 and T0216 are unambiguous. However, the pairing heuristic would interpret the background as the occluding object.

By applying more global knowledge (see Chapter 4) to restrict the possible ambiguous interpretations a correct pairing always emerged.

The success of the occlusion analyses indicates that the various occlusion techniques complement each other very well. The models prevent the inappropriate cases such as Figure 6.6 from being referred to the pairing heuristic; while the heuristic competently handles occlusion and multi-occlusion problems that frustrate the model-to-region matching.

6.3 Control

One of the least pleasing aspects of the system is its serial analysis of the scene, taking each adjacent region in turn as it is guided by the relational information contained in the model hierarchy. Efforts to introduce a type of bottom-up or data-driven capability to the analysis were unsuccessful. There are several factors that led to this failure:

(1) Poor shape information.

Without the guidance of context-information, the shapes of the isolated regions did not offer any definite information. A rectangle could be a bow, or a sock, part of a shoe, etc. Some relational information is usually required to limit the possibilities.

A few regions could be recognised in isolation, but a "sweeping up" operation to identify the remaining regions would probably be much more difficult than the present scheme. Since there would be no coordination between the partial results, any conflicts would be hard to resolve.

(2) Occlusion.

Occlusion causes problems by altering the region shapes and their inter-relationships. Some of the difficulties in limiting occlusion processing exclusively to a low-level junction analysis were described in the previous section of this Chapter. High-level information proved to be essential.

A further problem is that of object overlap within a scene. The current system isolates "foreign objects" since they are not needed to complete the current model. A bottom-up system would need another method of separating parts corresponding to different models.

(3) The variety of sub-models.

Low-level parallel techniques seem to be more suited to simple static objects or ones with a fixed set of relationships between their sub-parts. Baseball bats, balls, or blocks are trivial in comparison with the complexity of the person model with its variety of sub-models to account for changes of character identity as well as contortions of the body. Finding a consistent whole model would almost certainly involve a lot of

thrashing through various sub-model possibilities. The serial top-level approach avoids this problem.

(4) Non-uniformity of data.

The significance of the regions used as the basic primitives for analysis varies according to their role in the hierarchy. The top-down analysis by the model hierarchy uses this fact to its advantage; however, a bottom-up analysis has no means of deciding the level of importance of the individual pieces. A single region might represent a whole object or an insignificant part of an object formed of over twenty regions. It is hard to imagine a "uniform" low-level process to deal with such non-uniform data.

We recognise the need for an interaction between data-driven and goal-driven control strategies. We feel that such techniques are best developed in domains with an abundance of low-level data as well as a rich set of high-level models. We found the relatively data-deprived cartoon domain to be unsuitable for such a task (although this is partially due to the lack of good shape descriptors). The current system is not as dependent on the top-to-bottom order of analysis as it may seem from the examples that have been presented. We have run a partial analysis of a scene starting from the lower portion of the screen. The analysis was moderately successful although the analysis of the LEG portion of the body involved a lot of thrashing between Composition models trying to

establish the proper relationships between the ten regions which formed the legs. This thrashing was caused by the similarity of the regions which form LUCY's shoes and the occlusion of one foot by another in a side view. (There was neither an established "view" for the scene nor a previously recognised LIMB to help distinguish the pieces.) The ordering of models in the data-base placed the female leg models at the top of the possibility list. If the male leg models had been tried there would have been even more problems since the occlusion routines would have been invoked to account for the smaller size of the three regions that compose LUCY's shoes, rather than the single region expected by the model. (See Figure 6.6 and the discussion above.)

The indications are that by making minor modifications to the model hierarchy a variety of starting points may be accomodated. Since the analysis is a serial one, we have maintained the top-to-bottom processing direction. To achieve a greater measure of independence from such a strict sequential ordering, modifications have been made to the Structure model to incorporate more general relational information which is used to recover when an intervening object blocks the standard analysis sequence. (See Section 5.4.3.)

Another area where a more general solution would be preferable to the limited technique employed by the system is the initial selection of Structure model, made on the basis of the outer closure of the

regions. This silhouette offers a global view of the scene yielding a rough indication of the height to width ratio and the ratio of area with respect to the bounding rectangle. This data is used to select an appropriate model and set the scale of the scene. This is a satisfactory technique if the objects are isolated, or if small objects intersect with large ones leaving the gross characteristics unaltered. For cases where objects of approximately the same size overlap, something more sophisticated is needed. Perhaps some low-level techniques could be used to trigger a model. Since the scale of the scene is not known, these would probably search for special shapes to trigger a specific model. Since most PEANUTS scenes include at least one person, special procedures might be used to locate the heads which are usually occluded and analyse the scene using that region group as a starting point. Another method might be to exploit a local context mechanism. (See Section 5.2.1.) People are often located with familiar objects, e.g. sitting at a desk, or in a car. Models of such typical unions could be used to simplify the analysis. If one views the PEANUTS cartoon scenes as examples of processed scenes of the real world with the vast selection of possible objects, this selection problem is very difficult. In a sense, one has to know what one is looking at before one can recognise it. The use of the outer closure is an attempt to capture the flavour of this argument.

CHAPTER 7

CONCLUSION

We divide this final Chapter into three sections:

(1) Retrospective reconsideration of the problem domain and solution: Many of our original ideas and opinions have been altered by the experience of designing the system. We re-examine some of our decisions.

(2) Unanticipated problems: The experience of writing and debugging the system led to the discovery of new problem areas and a re-appraisal of the difficulty involved in solving known problems.

(3) Achievements/Future development: We conclude with a discussion of our achievements and the future possibilities of research in this domain as well as related areas of vision research.

7.1 Retrospective Reconsideration

7.1.1 Choice of Domain

In Chapter 1 we presented our reasons for choosing this cartoon world. Here we analyse the influence of the domain on the nature of the solution.

When we originally selected this cartoon domain, we did not anticipate the extent of the difficulty we would have in

characterising the irregular shapes. Had we chosen a more restricted world (i.e. one in which the shapes were less irregular -- more suitably described) the systems dependence on model-driven analysis would have been reduced. While we can only speculate on the results that might have been achieved within a different universe, the influence of the PEANUTS universe on this system is quite evident. Although we believe some of our techniques will have more general applictions, others must be seen either as solutions to problems that are limited to the chosen domain or as limited solutions to general problems. As an example of the former we cite the camouflaging of surfaces by certain region configurations usually corresponding to colour changes. In the real world depth information and texture would reduce this effect. The characterisation of a person on the basis of hair style alone, and the limitation of the number of allowable views are examples of the latter type of solution: suitable for this domain, but requiring more robust counterparts for real-world applications.

By restricting the domain, we may also restrict the range of possible solution techniques. Perhaps the best example of this is the unnecessary effort that was spent eliminating the shadows in TV images of the blocks world which could have provided valuable information to aid in the analysis of the scene [Waltz 1973].

The introduction of additional features such as colour to our sparse environment would probably simplify some of the problems we faced. It is difficult to find the proper balance between the toy worlds and the real world. Techniques which effectively solve problems in restricted worlds may not generalise, yet the proper application of the wealth of features available in real world scenes is beyond the current state of the art. Simple domains serve as a valuable device for the initial investigation of a problem, but care must be taken to ensure that the domain is not overly restricted to the extent that the solution is not representative of more general domains.

We view our research as an initial investigation of the problems in recognising flexible non-geometric objects subject to self-occlusion as well as occlusion by other objects. Based on our experience, we have reservations concerning the value of further research of this nature in the cartoon domain. The role of context information may be over-emphasised due to the lack of feature clues. Our model recognition procedures indicate that even rough shape features are valuable in distinguishing similar objects; presumably by enriching the domain with more features, (increasing redundancy of information) one may achieve better recognition results and lessen the effects of occlusion. In the cartoon domain certain common types of self-occlusion were most efficiently handled by incorporating the expected instances into the model. It would be interesting to determine if this technique was also appropriate for richer domains.

7.1.2 Implementation Language

We chose CONNIVER as our programming language because it had all the features we thought we needed: pattern-directed invocation, data-base monitoring, sophisticated control mechanisms, etc. It turned out to be a very large, unwieldy, and inefficient system to use on our machine. In retrospect it would have been wiser to use the locally supported language, POP-2, and build into the system only the features that proved necessary.

There were various programming errors that were either caused by system bugs or our misinterpretation of the manual. We were able to use alternative programming strategies to overcome these problems, but this type of annoying problem is a consequence of being the sole user of CONNIVER in a computer user community. The enormous size of the system was another major drawback which led to the segmentation of the system into a pre-processing program and recognition phase instead of allowing a complete analysis from digitised data through to the final scene description.

While CONNIVER does have a number of useful features, it is a dreadful system to use, growing larger (and slower) all the time. Closures of methods and procedures caused great increases in the program size. New contexts were easy to generate as one proceeded but much more difficult to splice into the existing context-tree structure. It was a constant battle to get the system to do o bidding, an experience to avoid if possible.

7.1.3 System Design Decisions

The final shape of the system was influenced by key decisions which were made rather early in the design stages. The two choices which have had the most effect are:

(1) Using only size and feature information instead of precise shape information related to the curves, and

(2) Using models based on the expected region appearance of the scenes rather than the deeper underlying structure of the objects that were represented.

Within this cartoon domain both these decisions seem reasonable and we would probably take the same choices if we started again.

We spent several months trying various schemes for characterising the irregular shapes that are represented in the cartoons. The variation of the shapes (representing the same object part) and the disfiguring effect of occlusion ruled out various transforms as well as curve-fitting techniques. Much more than in the real world, context seems to play a major role in the way 'shapes' or, in our case, region boundaries are interpreted. It was very much a matter of being able to recognise a region as an instance of a particular shape if there was a priori knowledge of what it should be. In a universe where shapes are more easily described or perhaps a more complex universe which offered an abundant range of features to better characterise objects, the importance of the context effect would be reduced, and low-level processing would play a more important role in

the recognition process. By limiting the desciptive aspect of the program to size and feature extraction techniques (for specific sub-model parts) we increased our dependence on model-based top-down analysis.

The model hierarchy we employ is used to match the regions in the scene to parts of objects. Although the hierarchy can handle changes in the position and orientation of the objects, the three-dimensional knowledge is coded to reflect the appearance of the object (in terms of regions). There is no explicit knowledge of the underlying three-dimensional structure of the object (although the components of the Structure model do reflect the independence of the structural parts of the model). In a more complex domain, especially if there was access to depth information, such an explicit model would probably be more useful. In the two-dimensional cartoon world, the depth information is limited to clues obtained from some line junctions. (The models help distinguish junctions indicating occlusion from those associated with "abutting" regions.)

The hierarchy also proved to be useful for modelling standard cases of self-occlusion. The dependence on the T-junction heuristics was reduced by incorporating new models to represent the occluded appearance of the object (in terms of the resulting region configuration) in the hierarchy.

Our success with this task re-affirms our belief that this model system is well-matched to the cartoon domain. If depth information is available (e.g. the laser ranging generalised cylinder system of Agin [1972]) then a model system capable of dealing with that very different type of information would be required. In such a system structure not appearance is emphasised; in ours, the opposite is true.

7.2 Unpredicted Problems

There were several problem areas which we only discovered during the process of planning, writing and debugging the performance program. We have already discussed our difficulties in characterising the irregular shapes in this domain. Here we discuss the problems of occlusion processing and control strategies.

7.2.1. Occlusion Problems

Most of the occlusion problems stem from the difficulty we had in interpreting junctions based on the available local evidence. We had rather naively assumed that the orientation of the junctions would be fairly obvious. Examination of the data proved otherwise. The T-joint pairing heuristic depends on local junction orientation information, so the unexpected ambiguity of the data led to a re-examination of the heuristic and its value in determining occlusion information. This motivated several changes to the system:

(1) The control structure was altered to allow reconsideration of choices regarding junction orientation.

(2) The pairing heuristic was used to provide a degree of global information to govern the orientation decisions. The examination of pairs of junctions allowed for consistency checking.

(3) Standard cases of occlusion were incorporated into the model hierarchy system to reduce the amount of necessary occlusion processing (which had become less reliable due to junction ambiguity and more cumbersome due to the added control procedures).

The resulting program not only works in the presence of the ambiguous junctions, but the incorporation of standard self-occlusion instances into the model seems to be a more reasonable approach than our original exclusive dependence on the occlusion heuristic.

Another occlusion problem that should have been anticipated but only emerged during an attempted scene analysis is that of mutual occlusion, i.e. when a curved object is both in-front-of another object and behind it, so the two objects occlude each other. This type of occlusion cannot be handled by the pairing heuristic, (which is based on two-dimensional principles) and must be solved with special case knowledge. Although this solution is far less satisfying than the one above, we accept it as one consequence of choosing this peculiar two-and-a-half dimensional curved domain.

7.2.2 Control Strategy Problems

One of the positive points that emerged from the scene analysis is the very strong power of context information. Perhaps the inadequate shape description capabilities emphasised its importance, but context information clearly provided the best guidance for the recognition system. Although we did attempt to apply some low-level techniques the results were not very successful. We believe there are several reasons for this:

(1) The main cause of the problem is the very high degree of ambiguity at the region level. Without context information a region may have dozens of possible corresonding model parts. The number of constraints in this domain were too small to effectively apply a Waltz filtering type of solution.

(2) The high degree of similarity between different sub-models caused further complications. While knowledge of model similarities was used to advantage in our top-down approach, it was far more difficult to achieve good results from a bottom-up analysis. Choosing between conflicting suggestions was rather difficult. The filtering of model patterns (using the results of successful model matches to eliminate inconsistent model applications) is much less effective with uncertain results. There is insufficient (context) evidence to support the chain of deductions that eliminate the model applications.

(3) Occlusion is another source of problems. Since occlusion alters the perceived shapes of the regions it can add to the confusion by altering regions so they no longer meet their corresponding model's criteria but instead meet the criteria of another model.

(4) The pairing heuristic (as it was designed) depends on high-level knowledge to invoke it under the proper circumstances since not every T-junction indicates occlusion. The heuristic loses its value if it is invoked in the wrong circumstances generating incorrect hypotheses which can clog the system.

Despite these problems, we do not wish to claim that bottom-up processing has no role to play in a vision system. We felt that a top-down approach in this domain offered a far more efficient solution. There was a sufficient number of other problems without complicating the system with a multi-processing low-level approach. We feel that such an approach is better suited for richer domains where there is more scope for carefully specified interaction between different types of feature detectors and scene descriptors.

7.3 Achievements and Future Development

7.3.1 Achievements

In such an open-ended domain (PEANUTS cartoons) it is difficult to

determine if the techniques which have been developed and tested on a number of scenes are sufficient to handle additional scenes. We have attempted to keep our procedures as general as possible, keeping in mind typical cartoon scenes in the domain. The implementation of the model hierarchy structure allows for the simple extension of the set of recognisable objects. Additional models may be added which share sub-model parts with the existing hierarchy. New models which are sufficiently similar to the existing models to cause confusion (i.e. undetectable mis-match of models to regions) will require the alteration of the existing models. The acceptance tests must be refined to distinguish the similar shapes. Knowledge of model similarities may be incorporated into the model hierarchy to shorten the search and/or preserve partial results. We do not regard the need for such alterations as a flaw in the system design. The refinement of the model system allowing it to distinguish between objects which closely resemble each other reflects an increase in knowledge. Just as a human expert can rely on minute differences to distinguish between objects which appear identical to a naive observer, the system can become more "expert" by re-balancing the recognition tests to take account of additional features.

The general reliability of the occlusion routines is harder to predict. The pairing heuristic is not guaranteed to handle all occlusion problems in this domain. However, we feel that it will be sufficient to handle almost all cases when used in conjunction with models which account for the standard cases of occlusion, and some

special-case routines to handle the three-dimensional problems (e.g. gripping hands) which are beyond the scope of the two-dimensional routine. The task is simplified by the artist who draws the cartoons. Cases of occlusion are usually drawn in a manner which minimises any possible confusion in the recognition process.

The hierarchical structure of the model system proved to be a very useful means of encoding the alternative region configurations representing the various views of the body in multiple poses. The simplicity of the cartoon world characters was well-suited to this representation of flexible bodies. The combination of the limited number of standard poses used by the artist and the inexact shape and relational information of the model system allowed us to restrict the necessary number of sub-models. Despite the limited number of sub-models used, the most essential information in the scene (e.g. character identity, position, and to some extent action) was extracted.

The success of this experimental system is related to two basic facts:

(1) Exact or precise information is not usually necessary for recognition tasks.

The shape descriptions we use are very crude. Some specific shape features such as lobes or inflections points are used but their <u>exact</u> location and size is not an important

factor. The directional relationships between neighbouring parts in the system are limited to: above, below, left and right. Small variations in position will not be detected by the system just as human observers would be unlikely to notice such changes.

(2) Context information is a very essential component in the recognition process.

The system uses both global and local context information to select models and recognise objects in the scene. Knowledge of the global context (e.g. baseball scene) alters the expectation of finding an object in the scene. Even more important is the local context information provided by the model system that is used to properly interpret ambiguous shapes. The overall effect (or context) overshadows the local shape characteristics.

To some extent these principles are determined by our chosen domain, and may not hold in other situations. (For example, robot assembly tasks require very precise knowledge of location and orientation to allow manipulation of the objects.) However, our research leads us to believe that these are important considerations for all vision programs.

One of our original goals was to develop shape descriptors for irregular curves which would be suitable for both cartoon world and real world scenes. Despite our failure to derive any general shape descriptors, we were able to analyse the cartoon scenes. This may be an indication that precise shape descriptors are not essential to real world vision either. AI vision research in geometric toy worlds relied on such information because it was easy to obtain. In the real world which has so many different features to offer, perhaps shape is not as important as size, colour and texture, and local context relationships.

7.3.2 Future Development

The obvious next step for future development within the PEANUTS domain would be to analyse a sequence of cartoon scenes using knowledge of the previous scene to interpret the following one. The analysis of the first scene would provide strong context evidence for the second one. Knowledge of character identity and pose, as well as hand-held objects, could be applied to severely limit the model possiblities before analysing the data. The processing time for subsequent scenes should be reduced. At a deeper level the interpretation of earlier scenes might be used to "predict" the final scene. This is a much more complicated psychological task involving the interpretation of scenes in terms of character's motives and possible actions.

Our experiences of vision research in this artificial cartoon world have made us question the value of work in such limited domains. Certain underlying assumptions which may be exploited in a restricted world may not hold in more general domains. Therefore, the results may not be extensible to other domains (where the assumptions are not valid.) The most interesting work for the future would be to try to extend the basic model system and occlusion techniques to handle real world scenes. We believe the hierarchical model system would prove to be a valuable tool in such scenes to provide an understanding of the basic structure of flexible objects. The availability of more features shold improve the overall performance of the recognition phase of the models. We suspect that the occlusion techniques we developed (founded on two-dimensional principles) will prove to be less appropriate in the real world than in the "two-and-a-half" dimensional world of curved line drawings. However, if three-dimensional cues are available alternative solutions may be found to replace them.

BIBLIOGRAPHY

Adler, M. (1975) Understanding Peanuts Cartoons. In Progress in Perception. Department of Artificial Intelligence Research Report No. 13, University of Edinburgh.

Adler, M.R. (1976) Recognition of Peanuts Cartoons. In AISB Conference Proceedings, July 1976, pp. 1-13. University of Edinburgh.

Agin, G.A. (1972) Representation and Description of Curved Objects. Ph.D. Thesis. Stanford AIM-173. Stanford University.

Barrow, H.G. and Popplestone, R.J. (1971) Relational Descriptions in Picture Processing. In Machine Intelligence 6, eds. B. Meltzer and D. Michie, pp. 377-396. Edinburgh: University Press.

Barrow, H.G. and Tenenbaum, J.M. (1975) Representation and Use of Knowledge in Vision. SIGART Newsletter, No. 52, pp. 2-9. Edinburgh: University Press.

Barrow, H.G. and Tenenbaum, J.M. (1976) MSYS: A System for Reasoning about Scenes. Stanford Research Institute A. I. Center Technical Note 121.

Baumgart, B.G. (1973) Geometric Modelling for Computer Vision. Stanford AIM-249. Stanford University.

Blum, H. (1964) A Transformation for Extracting New Descriptions of Shape. In Proceedings of the Symposium on Models for Perception of Speech and Visual Form, Boston, November 1964, pp. 362-280.

Bobrow, D.G. and Wegbreit, B. (1973) A Mode and Stack Implementation of Multiple Environments. CACM, Vol. 16, No. 10, pp. 591-603.

Bornat, R. and Brady, J.M. (1976) Using Knowledge in the Computer Interpretation of Handwriten FORTRAN Coding Sheets. Int. J. Man Machine Studies, Vol. 8, pp. 13-27.

Clowes, M.B. (1971) On Seeing Things. Artificial Intelligence, Vol. 2, No. 1, pp. 79-116.

Clowes, M.B. (1972) Computer Interpretation of Pictorial Data. Laboratory of Experimental Psychology, University of Sussex.

Clowes, M.B. (1973) Lectures given at AISB Summer School, Oxford.

Deregowski, J.B. (1972) Pictorial Perception and Culture. Scientific American, Vol. 227, pp. 82-88.

Draper, S. (1975) Grape -- Some Notes. School of Cognitive Studies, University of Sussex.

Duda, R.O. and Hart, P.E. (1973) Pattern Classification and Scene Analysis. New York: John Wiley and Sons.

Fahlman, S.E. (1973) A Planning System for Robot Construction Tasks. Artificial Intelligence Technical Report AI-TR-283. Artificial Intelligence Laboratory, M.I.T.

Falk, G. (1972) Interpretation of Imperfect Line Data as a Three-Dimensional Scene. Artificial Intelligence, Vol. 3, 101-144.

Fischler, M.A. and Elschlager, R.A. (1973) The Representation and Matching of Pictorial Structures. IEEE Trans. on Computers, Vol. C-22, No. 1, pp. 67-92.

Freuder, E.C. (1976) A Computer System for Visual Recognition using Active Knowledge. Artificial Intelligence Technical Report AI-TR-345. Artificial Intelligence Laboratory, M.I.T.

Gibson, E.J. (1969) Principles of Perceptual Learning and Development. New York: Appleton-Century-Crofts.

Grape, G.R. (1973) Model Based (Intermediate-Level) Computer Vision. Ph.D. Thesis. Stanford AIM-201. Stanford University.

Gregory, R.L. (1974) Concepts and Mechanisms of Perception. London: Duckworth.

Guzman, A. (1968) Decomposition of a Visual Scene into Three-Dimensional Bodies. AFIPS, Vol. 33. Washington, D.C.: Thomspon Book Co.

Guzman, A. (1971) Analysis of Curved Line Drawings using Curved Context and Global Information. In Machine Intelligence 6, eds. B. Meltzer and D. Michie, pp. 325-376. Edinburgh: University Press.

Haber, R.N. and Hirshenson, M. (1973) The Psychology of Visual Perception. London: Holt, Rinehart and Winston.

Hewitt, C. (1970) PLANNER: A Language for Manipulating Models and Proving Theorems in a Robot. Artificial Intelligence Memorandum 158, M.I.T.

Hewitt, C., Bishop, P. and Steiger, R. (1973) A Universal ACTOR Formalism for Artificial Intelligence. In IJCAI-3, pp. 235-245.

Hinton, G. (1976) Using Relaxation to Find a Puppet. In AISB Conference Proceedings, July 1976, pp. 148-157. University of Edinburgh.

Hollerbach, J.M. (1975) Hierarchical Shape Descrption by Selection and Modification of Prototypes. Artificial Intelligence Laboratory TR-346.Master's Thesis. M.I.T.

Hubel, D.H. and Wiesel, T.N. (1965) Receptive Fields and Functional Architecture in Two Non-striate Visual Areas (18 and 19) of the Cat. J. Neurophysiology, Vol. 28, pp. 229-289.

Huffman, D.A. (1971) Impossible Objects as Nonsense Sentences. Machine Intelligence 6, eds. B. Meltzer and D. Michie, pp. 295-323. Edinburgh: University Press.

Kelly, M.D. (1970) Visual Identification of People by Computer. Ph.D. Thesis. Stanford University.

Kuipers, B.J. (1975) A Frame for Frames: Representing Knowledge for Recognition. A.I. Memo 322. In Representation and Understanding, eds. D.G. Bobrow and A.M. Collins. New York: Academic Press.

McDermott, D. and Sussman, G. (1974) The CONNIVER Reference Manual. Artificial Intelligence Laboratory Memo 259A. M.I.T.

McLennan, M. (1975) Understanding Simple Plant Pictures. In Progress in Perception, Department of Artificial Research Report No. 13, pp. 74-98. University of Edinburgh.

Mackworth, A.K. (1974) On the Interpretation of Drawings as Three-Dimensional Scenes. Ph.D. Thesis. University of Sussex.

Marr, D. (1974) On the Purpose of Low Level Vision. Artificial Intelligence Laboratory Memo 324. M.I.T.

Marr, D. (1975) Early Processing of Visual Information. Artificial Intelligence Laboratory Memo 340. M.I.T.

Marr, D. and Nishihara, H.K. (1975) Spacial Disposition of Axes in a Generalized Cylinder Representation of Objects that do not Encompass the Viewer Artificial Intelligence Laboratory Memo 341. M.I.T.

Marr, D. and Nishihara, H.K. (1976) Representation and Recognition of the Spatial Organization of Three Dimensional Shape. Artificial Intelligence Laboratory Memo 377. M.I.T.

Milner, D. (1970) A Hierarchical Picture Interpretation System using Contextual Information. Diploma Project Report, School of Artificial Intelligence, University of Edinburgh.

Minsky, M. and Papert, S. (1972) Artificial Intelligence Progress Report. Artificial Intelligence Memo 252. M.I.T.

Minsky, M. (1975) A Framework for Representing Knowledge. In Psychology of Computer Vision, ed. P.H. Winston. New York: McGraw-Hill.

Nevatia, R. and Binford, T.O. (1977) Description and Recognition of Curved Objects. Artificial Intelligence, Vol. 8, pp. 77-98.

Nilsson, N. (1971) Problem Solving Methods in Artificial Intelligence. New York: McGraw-Hill.

Ohlander, R. (1975) Analysis of Natural Scenes. Ph.D. Thesis. Carnegie-Mellon University.

Paul, J.L. (1976) Seeing Puppets Quickly. In AISB Conference Proceedings, July 1976, pp. 221-233. University of Edinburgh.

Paul, J.L. (1977) An Image Interpretation System. Forthcoming Ph.D. Thesis. University of Sussex.

Roberts, L.G. (1965) Machine Perception of Three-Dimensional Solids. In Optical and Electro-Optical Information Processing, eds. J.T. Tippett, et al., pp. 159-197. Cambridge, Mass.: M.I.T. Press.

Schulz, C.M. (1976) Peanuts Cartoons. Harmondsworth, Middlesex: Penguin Books.

Shepard, R.N. and Metzler, J. (1971) Mental Rotation of Three-Dimensional Objects. Science, Vol. 171, pp. 701-703.

Shirai, Y. (1975) Analyzing Intensity Arrays using Knowledge about Scenes. In Psychology of Computer Vision, ed. P.H. Winston. New York: McGraw Hill.

Sloman, A., Hinton, G., O'Gorman, F. and Owen, D. (1977) Popeye's Progress through a Picture. Cognitive Studies Programme, School of Social Sciences, University of Sussex.

Sussman, G. and McDermott, D. (1972) Why Conniving is Better than Planning. Artificial Intelligence Laboratory Memo 255A. M.I.T.

Tenebaum, J.M. (1973) On Locating Objects by their Distinguishing Features in Multisensory Images. A.I. Center Technical Note 84. SRI Projects 1187 and 1530.

Turner, K.J. (1974) Computer Perception of Curved Objects using a Television Camera. Ph.D. Thesis. University of Edinburgh.

Waltz, D.G. (1972) Generating Semantic Descriptions from Drawings of Scenes with Shadows. Artificial Intelligence Technical Report AI-TR-271. Artificial Intelligence Laboratory, M.I.T.

Warnock, J.E. (1969) A Hidden-Surface Algorithm for Computer Generated Halftone Pictures. Technical Report 14-15. Department of Computer Science. Salt Lake City, University of Utah.

Winston, P.H. (1972) The M.I.T. Robot. In Machine Intelligence 7, eds. B. Meltzer and D. Michie, pp. 431-463. Edinburgh: University Press.

Winston, P.H. (1973) Progress in Vision and Perception. Artificial Intelligence Laboratory, M.I.T.

Winston, P.H. ed. (1975) Leaning Structural Descriptions from Examples. In The Psychology of Computer Vision, pp. 157-200. New York:McGraw-Hill.

Yakimovsky, Y. (1973) Scene Analysis using a Semantic Base for Region Growing. Ph.D. Thesis. Stanford AIM-209. Stanford University.