

VISUAL PERCEPTION IN A CHANGING ENVIRONMENT:

A PARADIGM FOR REAL TIME SCENE ANALYSIS

by
David Crossland Hogg

5/

Department of Computer Science

Submitted in partial fulfillment
of the requirements for the degree of
Master of Science

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
October, 1976

© David Crossland Hogg 1976.

THE UNIVERSITY OF WESTERN ONTARIO-FACULTY OF GRADUATE
STUDIES

C E R T I F I C A T E O F E X A M I N A T I O N

Chief Advisor

A. H. Dixon

Advisory Committee

Examining Board

John Givon

G. W. Wood

C. Kenneth Bouchey

The thesis by
David Crossland Hogg

entitled
Visual Perception in a Changing Environment:
A Paradigm for Real Time Scene Analysis

is accepted in partial fulfillment of the
requirements of the degree of
Master of Science

Date Dec. 15, 1976

Arthur L. Givon
Chairman of Examining
Board

ABSTRACT

A paradigm is proposed for generating descriptions, in real time, of simple visual situations involving one or two moving objects.

Particular attention is paid to the integration of low level feature extraction routines with high level processes concerned with mapping such features into an internal representation of an interpretation.

Various problems usually encountered within a low level system are resolved by providing high level guidance based on certain presuppositions about the scene domain.

An implementation of the system serves as an illustration throughout the thesis.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Tony Dixon, for introducing me to the study of motion analysis and for his guidance throughout the preparation of this thesis.

I am very grateful to my internal readers, Prof. Ted Elcock and Prof. Kee Dewdney, for their criticisms of earlier versions of this dissertation and to my external reader, Prof. J. Girvin.

In addition, I would like to take this opportunity to express my gratitude to all those people who have made my stay in Canada so rewarding.

Finally, I want to thank my parents for being so understanding during the past year.

TABLE OF CONTENTS

	page
CERTIFICATE OF EXAMINATION	ii
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
CHAPTER 1 - INTRODUCTION	1
1.1 Objectives	2
1.1.1 Verification of the Methodology	3
1.2 Discussion	5
1.2.1 Hardware	5
1.2.2 The Recognition of Elementary Motions	6
1.2.3 Higher Level Motion Concepts	7
1.2.4 Representation	8
CHAPTER 2 - THE SYSTEM	13
2.1 Development of a Representational Scheme	13
2.2 A Methodology for Procedures	19
2.3 The Low Level System	24
2.3.1 FRAME	25
2.3.1.1 A Low Level Motion Recognition Scheme	36
2.3.1.2 A Manipulative Theory of Forms	38

2.3.2 GROUP	40
2.3.2.1 Object Representation	40
2.3.2.2 Pair Representation	40
2.3.2.3 Object Maintenance - Tracking	41
2.3.2.3.1 Criteria of "best fit"	43
2.3.2.4 Pair Maintenance	45
2.3.3 CREATEGROUP	45
2.4 The High Level System	47
2.4.1 Locative Procedures	48
2.4.2 Recognition of Motion Concepts	49
2.4.3 Object Identification	51
2.4.4 Object Interaction	54
2.5 Natural Language Output	56
2.6 The Implementation	56
CHAPTER 3 - CONCLUSIONS	59
BIBLIOGRAPHY	63
VITA	65

LIST OF FIGURES

Figure	Description	Page
1	Illustrating the relation "refers to the assertions of" between the procedures appearing in the implementation	18
2	Partition over a conceptual network	23
3	Illustrating the representation of a form	30
4	Illustrating the representation introduced by FRAME	31
5	Recognition of elementary motions	37
6	Configuration for implementation	57

CHAPTER 1

Introduction

The field of automatic visual perception has received considerable attention during past years. Most of the research in this area has, however, been devoted to the analysis of static environments. Apart from a small number of recent contributions, a study of the automatic visual perception of changing environments has been almost neglected. This domain has many direct applications in areas such as robotics, industrial automation and security. As a more immediate application, it has been demonstrated that information about the motion within a scene can provide assistance in those areas which have traditionally been associated with static environments. At the lowest level, segmentation of a scene into its constituent objects can be effected using motion information (Potter[12]), whilst at a higher level, it has been suggested that motion provides strong cues for the identification of objects (Michotte[10]), a problem that has always been central to

visual analysis in static environments.

1.1 Objectives

The purpose of this thesis is to propose a methodology for generating reasonable descriptions, in real time, of simple visual situations involving one or two moving objects. Scenes are perceived as a two dimensional picture by a T.V. camera system which generates digitized brightness information suitable for further processing by a computer.

Particular attention is paid to the integration of (i), low level feature extraction routines with (ii), higher level processes concerned with mapping such features into an abstract representation in a computer. Previous contributions in related areas, many of which will be discussed, have tended to concentrate in one of these areas and assumed the other, thereby avoiding many of the problems and, at the same time, being unable to realise the benefits of a joint consideration.

Various problems normally encountered within a low level system are resolved by providing informed and, therefore, more reliable high level guidance. The occlusion of one object by another and the severe deterioration of basic visual features will be recognised and the problems they present overcome in this way. Similarly, object recognition for the purposes of tracking, is performed with

high level guidance to reduce relocation time and increase reliability.

An important consequence of the proposed methodology is the support it lends to the supposition that certain significant concepts involving motion can be recognised in a scene without the need for a complicated analysis of each of a sequence of instances of the scene.

1.1.1 Verification of the Methodology

An implementation, based upon the ideas contained in the system proposed by this thesis, was constructed by the author to demonstrate their feasibility. A flavour of the system's capabilities and intentions is given by the following description produced by the implementation from an actual scenario set in the laboratory. The scenario will first be described by the author.

The scenario begins in an empty room, containing only a table to the right of the scene as perceived by a T.V. camera. A person enters the scene from the left and crosses to walk behind the table, thus being partially occluded. He places a package on the table and leaves the scene to the right. A second person enters at the right of the scene and walks in front of the table, thus totally occluding the package, crossing to the left of the scene where he lies down on the floor and immediately stands up again. This same person then crosses back to the table, removes the

package and leaves the scene to the right.

Real Time Description Produced by the Implementation:

An object has appeared at left of scene, call it object A.

Object A has begun moving.

Object A looks like a person, call it Fred.

An object has appeared in middle of scene, call it object B.

Object B came from Fred.

Fred placed object B in the scene.

Object B looks like an inanimate object.

Fred has disappeared at right of scene.

An object has appeared at right of scene, call it object C.

Object C has begun moving.

Object C looks like a person, call it Mary.

Mary has completely crossed the scene.

Mary has stopped moving.

Mary has laid down.

Mary has stood up.

Mary has begun moving.

Mary has completely crossed the scene.

Object B has disappeared in middle of scene.

Object B was engulfed by Mary.

Mary picked up object B.

Mary has stopped moving.

Mary has disappeared at right of scene.

1.2 Discussion

1.2.1 Hardware

For visual analysis in a static environment, the computer eye is usually a conventional T.V. camera with special hardware attached. The T.V. picture is transformed into a computer useable image by measuring the light intensities at the points of a grid covering the entire picture. Various concepts, including edges, lines and eventually complete objects and the spatial relations between them, can then be deduced from these intensity values. However, significant hardware problems are encountered when we attempt to extract low level pictorial information in a changing environment using such equipment (Jones[7]). A moving object can become totally unrecognisable amongst the data extracted from the picture as a result of hardware distortions. In static environments, it is enough to sample each point once. However, in environments which are changing, we may want to sample a point many times. The sequence in which points are sampled is at our discretion, subject to certain timing constraints. The most obvious approach involves sampling sequences of entire pictures. With current hardware, we can only approximate this situation since the acquisition of each point involved in a complete image cannot be made at exactly the same instant. For a changing scene, therefore, each image will be distorted. However, recent developments

have enabled very fast complete picture acquisition, so minimising the inaccuracy introduced.

1.2.2 The Recognition of Elementary Motions

Having collected sufficient data, we must locate and describe low level changes in the scene. In particular, how do we recognise elementary motions? The basic approach involves initially locating simple features in the internal image. At a later time, these features are relocated on the basis of their type and position, assuming what has come to be known as the "consistency supposition" for changing environments. In this way, velocity and acceleration attributes can be efficiently recognised. The consistency supposition simply states that changes in the environment are of a continuous nature. Features found in any position at a particular instant remain arbitrarily close to that position during a sufficiently small temporal neighbourhood of the instant. For practical purposes, we also require that the rate at which such changes occur be bounded. Potter[12] proposes relocating templates matched to regions in an image whilst Jones[7] describes a system capable of tracking five basic feature types, located in windows extracted from the image. In general, the consistency supposition is insufficient in itself to relocate features since occlusion and orientation changes can cause them to completely disappear. High level guidance is essential for relocation in anything but the simplest of applications.

1.2.3 Higher Level Motion Concepts

Having discussed the recognition of elementary motions, we now consider several higher level motion concepts which might be recognised and suitably represented. In particular, what are the easily recognised motion concepts? Constant linear motion of an object is perhaps the simplest such concept and its recognition is necessarily fundamental to any system which attempts to analyze a changing environment. Zero velocity is usually explicitly identified, for it has special significance for the recognition of higher level concepts. Linear and angular acceleration (rotation) are also extremely useful simple motion concepts. Again, commencing motion and coming to rest can be explicitly represented. The system described in this thesis places special significance on the appearance and disappearance of objects; such events are particularly simple concepts of change.

At a higher level, we are interested in the interactions between objects. Weir[15] has constructed a program which attempts to recognise and classify several simple interactions between objects in the same way as did human subjects under experiments performed by Michotte[10]. Although the objects were simple geometric shapes, "pushing", "carrying away" and "frightening" are typical of the impressions received by the human subjects.

Perhaps the most significant collection of abstract motion concepts are the motion verbs used in natural language. By defining a hierarchy of motion concepts, Badler[1] has proposed a methodology for recognising the motions corresponding to a reasonable subset of the English motion verbs. Tsostos[14] describes a similar methodology which utilises a hierarchy of primitive motions that were rigorously defined by Miller[11], and clarifies several of the ideas developed by Badler.

Rather than working directly from the image generated by a television camera, the above three systems process a sequence of representations of instances of a scene. Each individual representation involving simply a list of object locations. By the absence, in each case, of a clearly defined methodology for generating this basic representational level, these systems can not be said to verify the supposition that certain significant motion concepts can be "easily recognised".

1.2.4 Representation

Suppose we are interested in a scene during a finite interval of time, referred to here as a "period". Given such a period, how should it be represented, both for ease of construction and simple interpretation by application programs?

The simplest approach would be to treat the period as a sequence of instances of the scene or "frames", similar to a movie film. Each frame can then be represented by the methods developed for representing static environments. This scheme is unsatisfactory, mainly because it offers no explicit motion concepts. In addition, most of the information contained in the representation for each frame will be repeated for many other frames, thus wasting time and space. This repetitive situation could be relieved by introducing a CONNIVER type data base[13], allowing only those parts of the representation which change between frames to be associated with the later frame, the remainder of the representation being global to that frame.

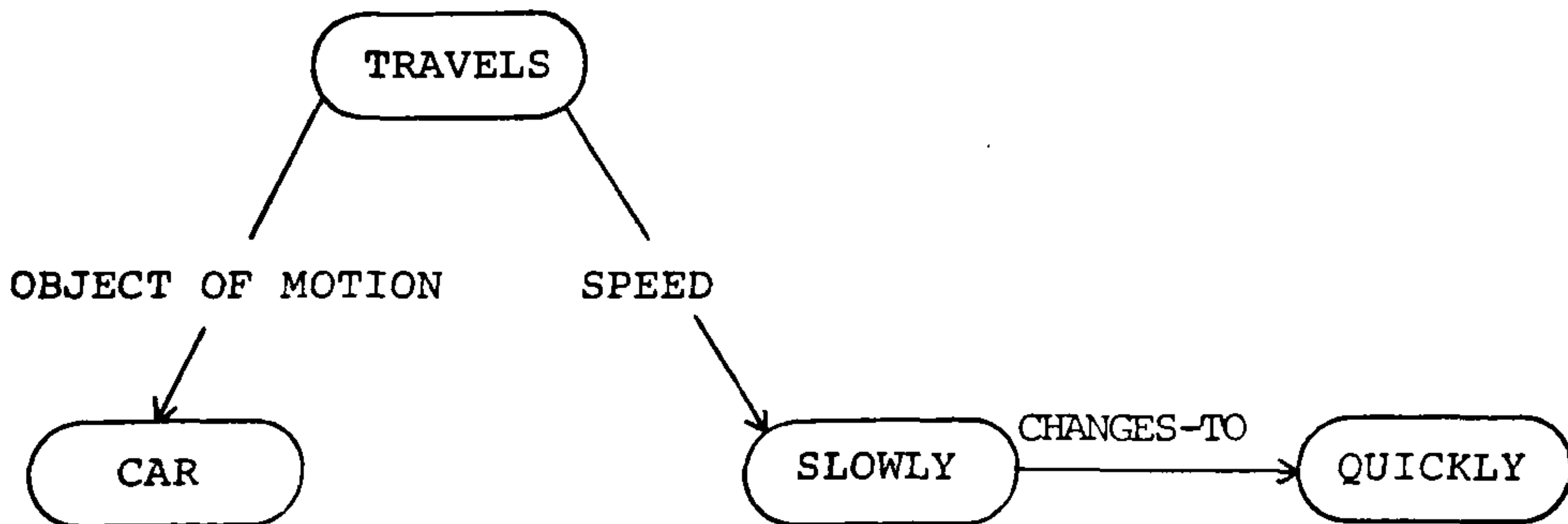
A more satisfactory representation is proposed by Badler[1], based upon the introduction of the "primitive event". Primitive events are simple motion concepts involving physical objects, and are in some sense the most elementary and yet meaningful changes that can occur in a natural scene. The basic motion concepts described in the previous section would each be primitive events. Each primitive event is represented by a list of properties containing such information as the subject of the event and the start time and end time of the event. By chaining primitive events together in chronological order, the fate of a particular object, during a period, can be easily represented. An entire period can, therefore, be represented as a chronologically directed graph of primitive

events. Such a representation could be said to "span" the period uniformly and without the dislocations inherent in a representation involving a sequence of frame representations. Given a detailed description of a single instance of the scene during a period, the information contained in the chained sequences of primitive events would be sufficient to extrapolate that description to any other instance of the scene during that period. The net loss of relevant information resulting through not representing explicitly many instances of the scene during the period would, therefore, be small. This construction is analogous to the solution of a differential equation; the representation and time being the dependent and independent variables respectively, whilst the chained sequences of primitive events are the systems of equations and the detailed descriptions of a single instance of the scene are the boundary conditions. Badler describes a methodology for constructing chains of primitive events from a basic representational level involving temporal sequences of object locations. Each motion concept is recognised by a particular demon, similar to those described by Charniak[3]. By recognising patterns between successive primitive events, demons are also able to assign English motion verbs to particular changes in the scene. The spatial concepts encountered as the subjects of primitive events and the relations between them are represented by a simple graphical network. For example, "the car is on the road" would appear

as:

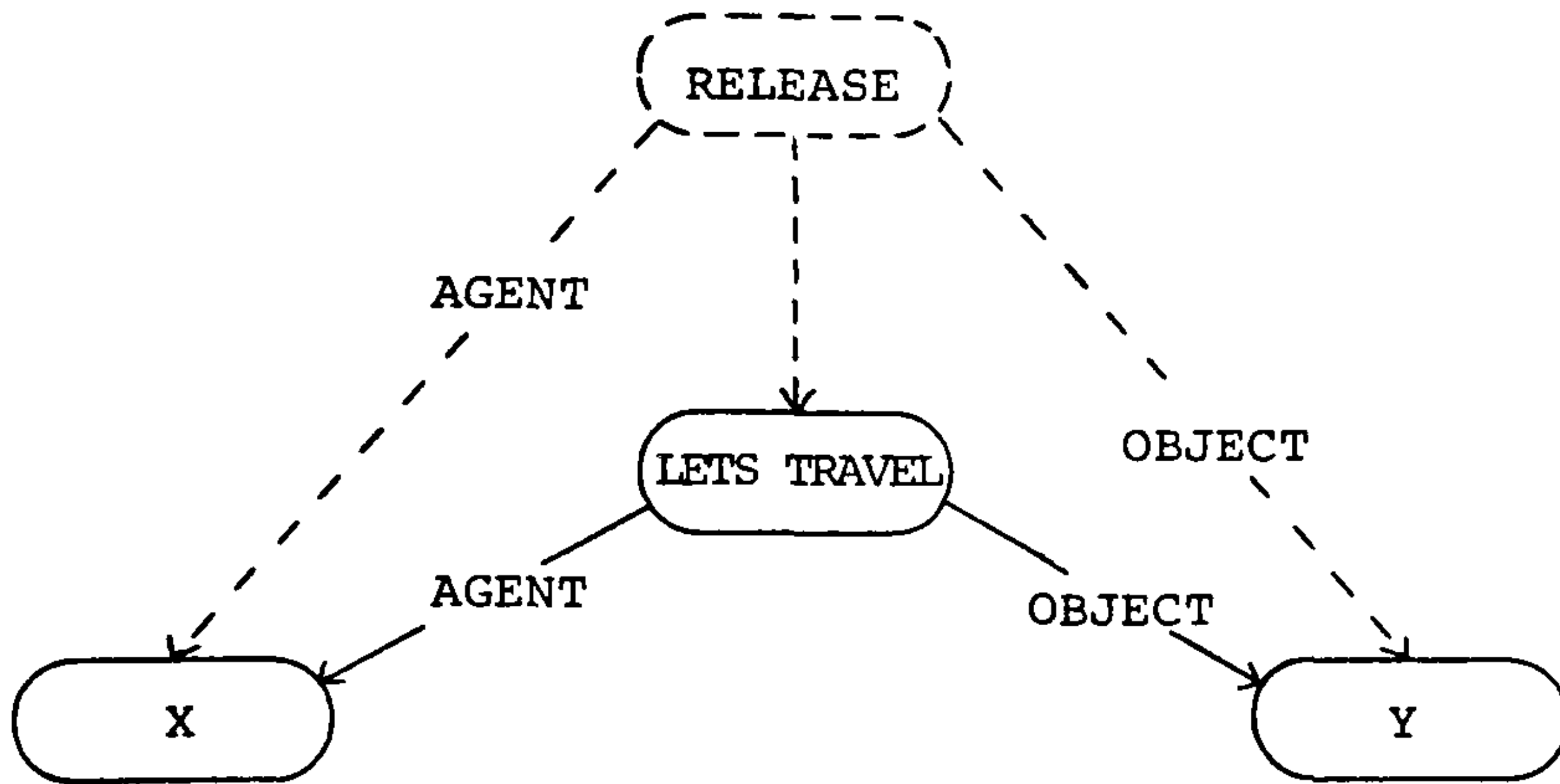


Tsostos[14] describes a system based upon that proposed by Badler. He greatly expands the graphical representational scheme employed by Badler to represent spatial concepts, so that it is also capable of representing primitive motions, as defined by Miller[11], which are essentially equivalent to primitive events. All concepts recognised during a period are, therefore, represented uniformly. For example, "the car travels slowly, then quickly" would appear as:



Corresponding to each motion concept recognised by his system, there is a model graph. Part of this model graph defines a pattern which is compared against the representation under construction. Should a successful match be found, then another part of the graph defines additions to be made to the representation, the details of

which may depend upon the conditions of the match. For example, the following model graph corresponds to the concept "release".



If the solid part of the graph is matched in the representation then the dotted part of the graph is added to the representation.

CHAPTER 2

The System

2.1 Development of a Representational Scheme

Most interesting applications involving visual analysis in a changing environment require immediate high level interpretations and, therefore, the existence of an evolving representation which is available throughout the period; an instance of this representation being particularly concerned with the state of the world in a temporal neighbourhood of the corresponding instant in time. Such real time representations introduce several additional aspects over the non-real time representations discussed in chapter 1.

The total real time representation for a period can be said to be composed of a continuum of instantaneous representations. Unfortunately, this observation provides little insight into the nature of the instantaneous representations. The simplest approach is suggested by considering a period to be composed of a continuum of

instances of a scene. Inspired by this observation, we maintain a representation throughout a period such that an instance of this representation describes, as accurately as possible, only the corresponding instance of the scene. This solution is more appropriate for a real time representation than was the similar solution for a non-real time representation for no attempt is made to retain more than the current instantaneous representation, much less the representation for the entire period. Again, however, this approach is unsatisfactory since it offers no explicit motion concepts, although several robotic systems requiring real time visual analysis have employed such a representational scheme (Winston[16]).

Given a system which generates a non-real time representation for a period, the following modification might produce a useful approximation of a real time system. The given non-real time system is altered such that, at any time during a period, the partially completed representation becomes available as an instance of a real time representation for that period. The system described by Badler[1] could be simply applied in this way, as was probably intended. Tsostos[14] assumes that his system be applied in this way, for he suggests using his representation to guide the selection of image points. Procedural information associated with each concept defining model graph can be used to initiate an active search for subconcepts in the scene.

In general, however, real time systems constructed in this way, fail to represent the essential involvement of the system, captured in the period. In particular, they cannot actively guide the acquisition of raw data, nor consider any active role the system could be taking in the environment. Nor do they necessarily provide detailed information about the current state of the scene. In addition, instances of the representation are prone to represent vast quantities of useless information concerning previous events as a result of the intended use of their main component. A simple improvement could be made to any real time system constructed in the above fashion by destroying those parts of the representation which are no longer "current" and which are not required for the recognition of other concepts.

The representational scheme employed by the system described in this thesis contains many of the ideas discussed during the previous pages. Periods are mapped into a real time representation, instances of which do not retain knowledge of long past events but provide a detailed representation for a temporal neighbourhood of the current believed state of the world.

The methodology can be divided conceptually into two parts, called the high and low level systems. The high level system concerns itself only with the recognition of those concepts involving complete physical objects (i.e. high level concepts), whilst the low level system is responsible for recognising such objects from lines and edges, etc. (i.e. low level concepts) deduced directly through the camera image.

Central to the paradigm is a set of distinct and recognisable concepts. In common with the primitive events of Badler[1], each instance of a particular concept is represented by a set of assertions (implemented by a property list, such as defined by LISP[9]), henceforth known as a "packet". At any time, the set of all packets constitutes the complete instantaneous representation.

Corresponding to each concept, there is a procedure whose purpose is to recognise instances of that concept during a period. In fact, such a procedure can be said to define the concept. Procedures are, therefore, responsible for introducing packets into the representation, where they are then available to other procedures. In addition, they are responsible for removing packets from the representation once they are no longer required. Control cycles continuously through the procedure set throughout the period. The time during which each procedure is being executed will be known as its "execution phase". Figure 1

contains each procedure appearing in the implementation of the system and illustrates the partial ordering imposed by the relation "refers to the assertions of".

Relative to the duration of certain conceptual changes in the environment, the execution of a terminating procedure may be considered to occur instantaneously, assuming temporary suspension to be forbidden. As a result, procedures are effectively unable to refer to the representation or to the camera image at more than a single instant. To enable the recognition of concepts involving motion, procedures are, therefore, capable of introducing into the representation packets corresponding to partially observed concepts, which can then be used by future executions of that procedure. Such concepts could also be recognised by enabling procedures to temporarily suspend themselves during execution. However, the equivalence of these two schemes is made clear if we consider those assertions corresponding to partial recognitions of a concept to be just the internal states of suspended procedures.

Representing a period as a sequence of frame representations is perhaps the simplest approach. Although the beginning of this chapter argued that such a scheme was unsatisfactory for a total real time representation, it is nevertheless appropriate for a constituent representational level. Periods are, therefore, represented at the lowest

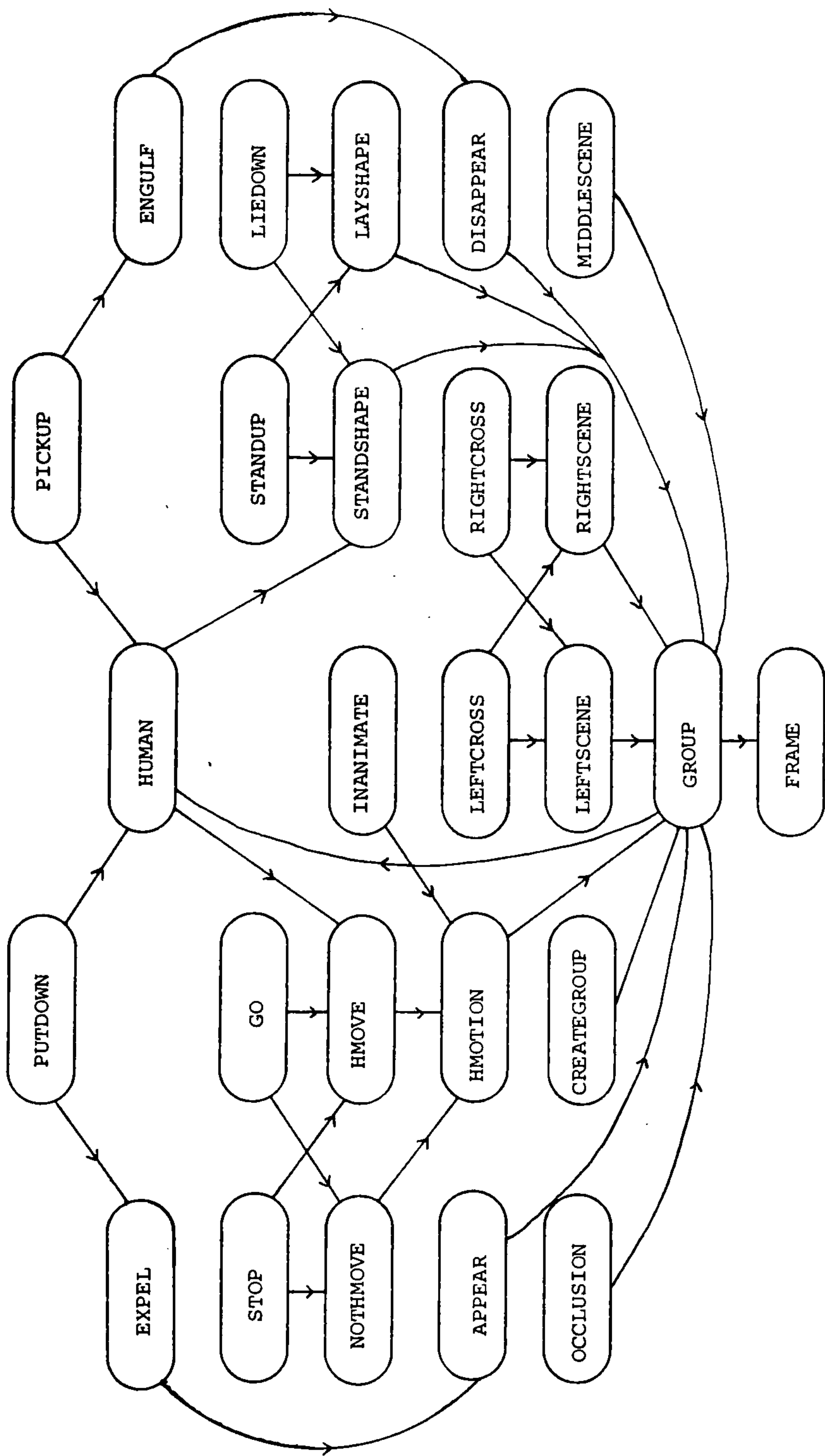


Figure 1. Illustrating the relation "refers to the assertions of" between the procedures appearing in the implementation

level by a sequence of frame representations. Instances of a scene are the most elementary low level concepts known to the system and are "recognised" by the "low level" procedure FRAME. Each execution of FRAME causes another frame to be extracted and a concise representation introduced into the main representation as a packet.

Many of the concepts recognised by the high level system are sufficiently significant to be of interest to the observer. Notice of their successful recognition is, therefore, transmitted to the observer and constitutes the desired real time description.

2.2 A Methodology for Procedures

Having introduced the function of a procedure, we now require a methodology by which such procedures may use the evolving representation to determine instances of the concepts to which they correspond. For the set of procedures which recognise high level concepts (i.e. high level procedures), such a methodology is made possible by introducing the concept of a group.

Definition:

A group is a set of complete physical objects.

A pair of oranges would, therefore, constitute a group. However, the texture or colour of their skins would not.

With reference to the linguistic analysis performed by Miller[11], it is obvious but nevertheless significant that most motion verbs or primitive motions involve either a single object or a pair of objects, ignoring subparts of objects and any reference objects. For example, the primitive motion "Y travels" requires only a single object whilst "Y applies force to make X travel" requires just two objects. In general, consider the following working hypothesis:

"Instances of a high level concept can be associated with a group such that.

- (i) The instance does not involve any object external to the group.
- (ii) The group is the simplest group that satisfies (i).
- (iii) At most one instance of each concept will be associated with any particular group."

At any time, the system embodies a known set of groups. By the hypothesis above, high level procedures are able to exhaust all possible instantiations of their corresponding concept by searching for an instance in association with each group known to the system. Rather than repeating this search machinery within each procedure, each high level procedure is executed many times during its execution phase; once in association with each group known to the system.

A high level procedure will normally be concerned with groups of a particular type. For example, object identification procedures should only be successful with single objects whilst a procedure that recognises the relationship "nearby" between two objects will only be successful with pairs of objects. Certain procedures, however, might be capable of functioning with many types of group. A procedure that recognises horizontal velocity might observe this motion in a collection of objects, just as it does in a single object.

Efficiency is important in any real time system. In particular, pieces of knowledge about the world, encapsulated in individual assertions, should be easily accessible, both by those processes responsible for maintaining the representation and by application programs which actually use the representation. Different representational structures provide different possibilities for optimising the availability of this declarative knowledge.

A possible solution to this accessibility problem involves clustering or partitioning the space of assertions in such a way as to enable particular assertions to be located by following easily computable pathways between clusters and applying a simple search scheme within the target cluster.

Consider the partition imposed upon the set of all possible packets characterising high level concept instances (i.e. the set of high level packets) by the equivalence relation "associated with the same group as". Consider now the set of high level packets required by a high level procedure executing in association with a particular group. These packets will be contained within the partitions associated with that group and any groups that it properly contains. If this were not the case, then any concept instance recognised during a successful execution would be involved with a physical object that was external to the associated group, thereby contradicting the hypothesis above. The search space for a particular assertion contained in a high level packet is, therefore, reduced to several partitions. In fact, by defining each group in terms of component groups, the search space can be reduced to a single partition. Individual packets within a partition can be identified by simply labelling each packet with a name assigned to its corresponding concept. This labelling is unique within a partition by virtue of part (iii) of the hypothesis above.

The above partition also simplifies the question of reference within individual partitions. Most assertions assert something directly about a group. Within each partition, therefore, there can be no ambiguity introduced by omitting explicit referents from these assertions.

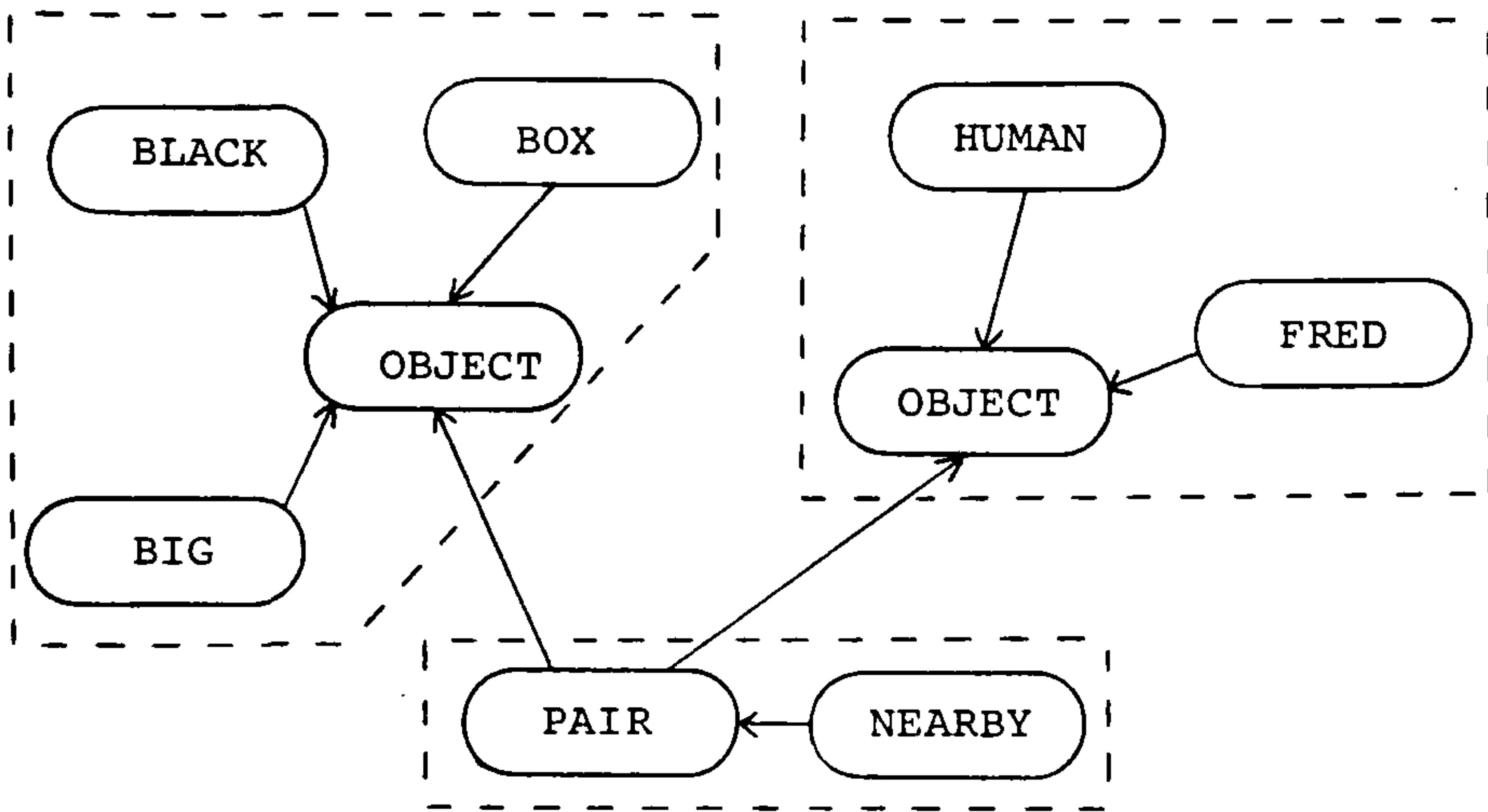


Figure 2. Partition over a conceptual network

Consider the set of assertions represented by the conceptual network in figure 2. Each node represents a particular instance of a concept with any referents being defined by the arcs. The network implies three groups; two consisting of objects and one consisting of a pair of objects. The group with which each concept instance is associated is illustrated by imposing the above partition on the nodes of the network. The two objects and the pair are associated with those groups which are composed of themselves. The arcs passing between the partitions become re-interpreted as "pathways" between partitions which define the group composed of a pair of objects in terms of the groups composed of individual objects.

Each group known to the system has an internal representation which is maintained by the low level procedure GROUP and introduced into the appropriate partitions as a packet. GROUP can best be described as a tracking procedure which interfaces the high level system with the low level system, for each cycle relocating all known groups through the representation introduced by FRAME.

The addition of new known groups is undertaken by the third and final low level procedure, CREATEGROUP. This procedure hypothesizes groups in the scene and, provided sufficient evidence is subsequently found to support their existence, incorporates them into the set of known groups.

Unlike the high level procedures, each low level procedure is executed only once during its execution phase and retains a single packet external to the partition imposed upon the set of high level packets, on the understanding that only one instance of its corresponding concept will exist at any time.

2.3 The Low Level System

The low level system consists of three procedures; FRAME, GROUP and CREATEGROUP, each concerned with subobject concepts. With an implementation in mind, the low level system assumes a hardware vision system similar to that available to the author, although application to a different configuration should present no difficulty.

The vision system can return a gray level between 0 and 255 corresponding to the light intensities from dark to light, according to a logarithmic or linear relationship. Such a value can be obtained from any point in a 512*480 square grid covering the entire field of vision as determined by the settings of coordinate registers in a controller.

The internal structure for each low level procedure was developed by experimenting with actual implementations of several preliminary ideas.

A detailed understanding of the low level system is best achieved by considering each procedure individually.

2.3.1 FRAME

FRAME is the only procedure having direct access to the vision hardware. The representation introduced by FRAME is, therefore, the lowest level packet available to the rest of the system.

FRAME employs a straightforward picture differencing algorithm to locate moving and, therefore, interesting objects in a scene. By comparing the images of successive frames with the image of an initial instance of the scene at the pixel level, any changes in the scene since that instance was memorised can be detected. This algorithm depends indirectly upon the existence of motion within a

scene to segment particular objects from the rest of the scene, thereby, verifying the conclusion of Potter[12], noted in chapter 1. The ease with which a moving object is located suggests that this is one of the most elementary features extractable from a sequence of images.

Simple though the picture differencing algorithm appears, several problems had to be solved before it could be incorporated into our system. Firstly, an implementation of the algorithm must be capable of real time performance.

By carefully choosing the order in which points are selected from the camera, it is possible, with the camera hardware described previously, to address every pixel in seventeen seconds. It is immediately evident that the desire to operate in real time prevents the use of complete images, seventeen seconds being too long for applications in a natural environment. In addition, the time required to extract useful information from 250,000 intensity values would be prohibitive. The solution to this information problem is very much dependent upon the type of elementary features to be extracted from each frame. If a detailed local analysis is required, for the detection of features whose positions are approximately known, then small windows can be extracted, requiring only a fraction of the time. Alternatively, the time to obtain a complete picture could be reduced by considering a representative grid, spread uniformly across the picture. The picture differencing

technique does not rely heavily on local detail and the system, therefore, employs the latter scheme.

A 64*64 representative square grid, covering the entire picture, reduces the number of pixels to 4096, an easily managed number of points. By normal scanning algorithms, this reduced grid would require two seconds to digitize completely. However, by applying a series of local sheers of the reduced grid over the original grid, a scheme suggested by Dixon[5], the entire picture can be digitized in 0.3 seconds.

This scheme, in common with most, involves selecting points from a narrow column which moves from the left to the right of the picture. The selection time interval between points near to one another is, therefore, very small, an essential property when considering a changing environment. Nevertheless, a distortion will be introduced into the images of all moving objects. The consistency supposition limits the speed of a moving object and, therefore, ensures that such distortions will remain "small enough" to be ignored.

The system is initialised by storing the gray level at each point from an initial instance of the scene, "the background". The period can then begin. As individual points are selected, the gray levels obtained are compared with those levels from the same positions in the background. If the absolute difference exceeds a given threshold value (system parameter) then that position is flagged. The significance of a flagged point being that something has changed at the corresponding point in the scene. In particular, this change could be caused by the presence of a moving object in the scene. Once the scan is complete, the flagged points are combined into regions on the basis of eight way adjacency in the grid structure. In this way, local evidence of change is consolidated to produce "change" regions.

In the simplest situation, each change region would correspond bijectively with certain objects in the scene and would appear as silhouettes of their images. However, in general, an object will give rise to two or more change regions. This implies the existence of "invisible" change regions, which cover that part of the object's image not included in the generated change regions. These invisible change regions arise because the gray values of their constituent points do not differ from those of the corresponding background points by the required amount or they arise from an occluding background object. A human observer would have no difficulty in deducing these "in

between" regions when they are required as part of an object, instantiated through the cues provided by the visible change regions. Association of the visible change regions with the same object, therefore, must be made by higher level procedures.

When two foreign objects come into contact with or partially occlude one another, individual change regions may be partly caused by both objects. Again, without guidance from higher levels it would seem impossible to isolate these objects.

The problems of association described above prompted separation of FRAME from the remainder of the system after generation of the change regions. The procedural structure of the total system could then be used at this basic level. Rather than listing their interior points, the change regions are described to the rest of the system by a concise and, therefore, manageable representation. Each region is represented by the four coordinate boundaries of the minimum rectangle which can completely contain that region. Such a set of numbers will henceforth be known as a form and the coordinate boundaries as LEFT, RIGHT, TOP and BOTTOM (figure 3).

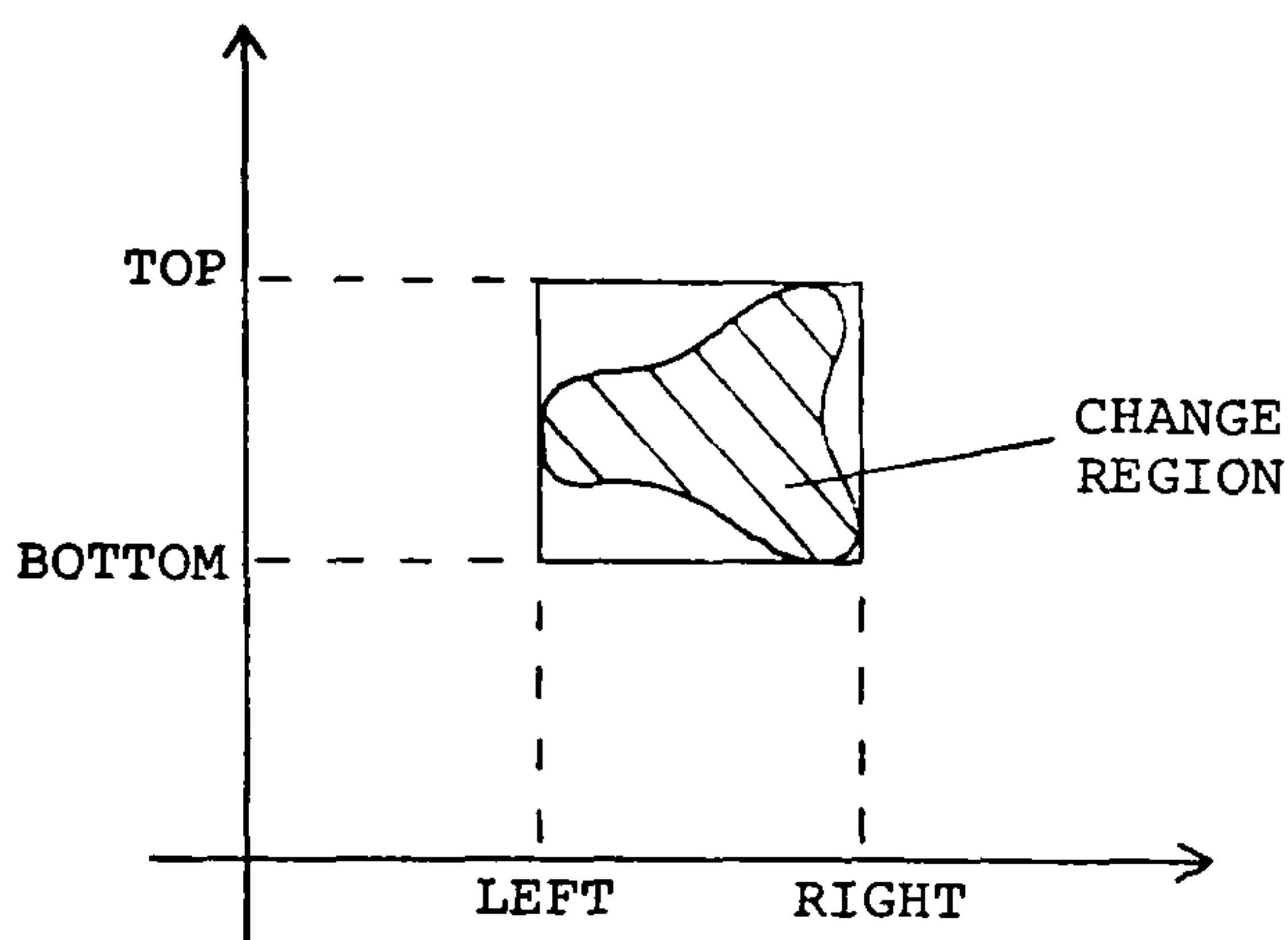


Figure 3. Illustrating the representation of a form

To guard against spurious gray levels caused by noise in the camera hardware, regions containing less than a given number of points are not communicated.

Figure 4 illustrates the basic representations introduced by FRAME resulting from the implementation and period described in chapter 1. Each frame depicts the forms generated during a single system cycle.

This example demonstrates several of the problems to be resolved by the rest of the system. The large forms, evident in frames 5,6,7 and 8 were caused by the person, who can clearly be seen crossing the scene. The person was partially occluded by the table during frames 14,15,16 and 17, the package, which was left behind, beginning to cause an identifiable sequence of forms at the same time. A person can again be seen to have entered the

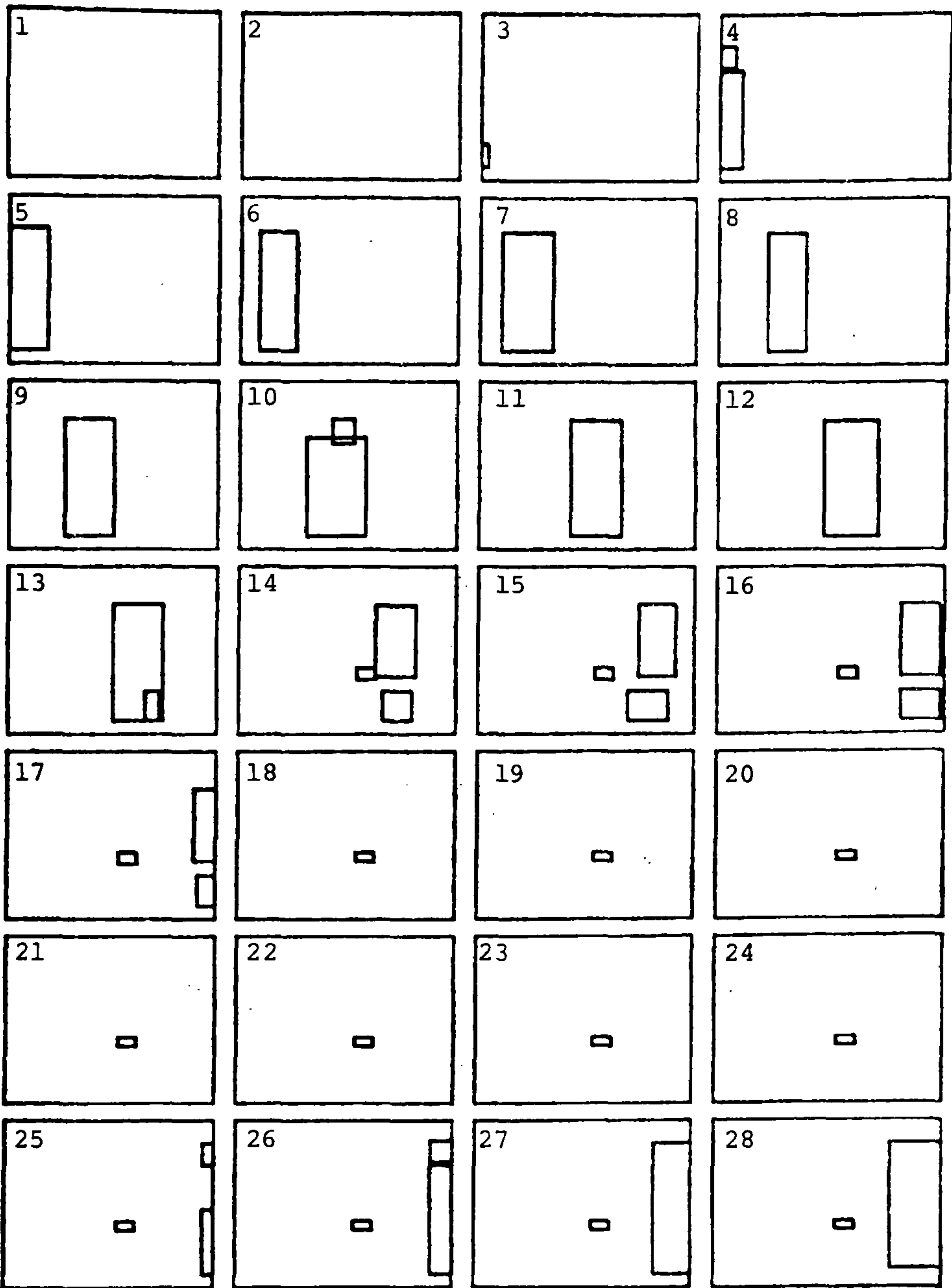


Figure 4i. Illustrating the representation introduced by FRAME

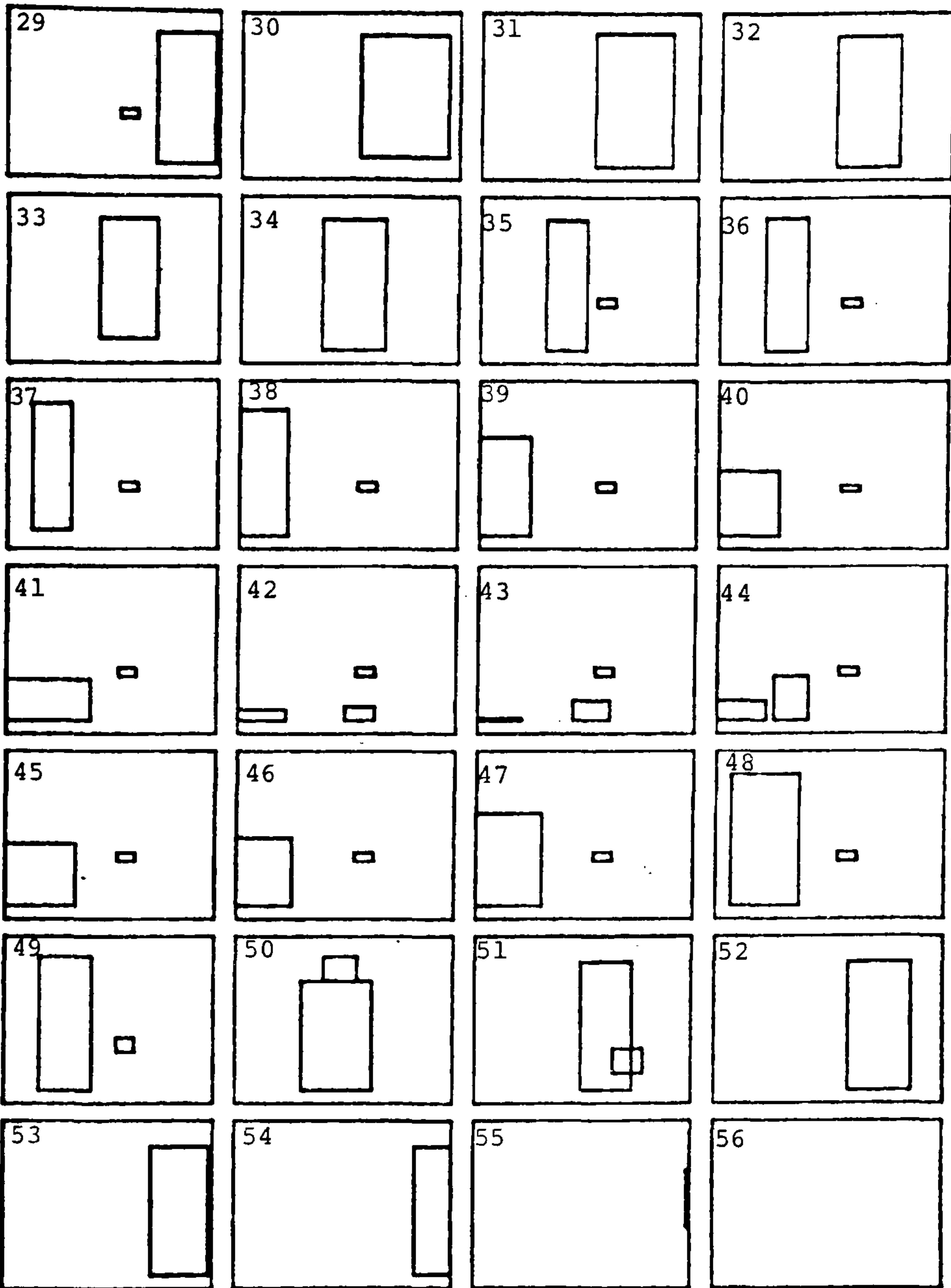


Figure 4ii. Illustrating the representation introduced by FRAME

scene in frame 25, crossing to occlude the package during frames 30,31,32,33 and 34. Deterioration of the resulting form set occurred as the person lay down and stood up during frames 39 to 47 and elsewhere in frames 4,10,13,25,26,50 and 51.

As demonstrated by Jones[7], the above picture differencing strategy has a serious drawback. In scenes where the light sources are varying in intensity, such as open air scenes on a cloudy day, the entire image will rapidly become a single change region. Alternatively, we could say that the background, very quickly, becomes out of date. Consider also the problems introduced by allowing motion of the camera or an object in the scene when the background was made, coming into motion, so leaving a hole in the image. Small, unimportant changes occurring in the environment continually create change regions while the study of detail within a moving object is impossible.

By making a small modification to the picture differencing strategy described above, these problems can be partially resolved but only at the expense of introducing other, equally severe, problems. As stated above, the difficulty lies with the background becoming out of date. However, we can ensure that the background is always up to date, by simply updating it, after each cycle. The background is always taken from the most recent frame. Such a modification was made temporarily to an implementation of

FRAME, with interesting results. Firstly, the system behaved very well when subjected to the awkward conditions above. Provided the undesirable environmental changes took place over a short period (1-2 scan periods), the change regions generated were few and could easily have been absorbed with little effect into the present high level system. Slow lighting changes and continuous camera movements were still a problem for they keep the picture in a constant state of change.

However, refreshing the background after each scan completely changes the nature of the change regions resulting from objects. Objects entering the scene are by assumption "merged" into the background after each scan. An object only produces change regions when it is in motion, once it becomes stationary then it effectively disappears. This has the advantage of further focusing attention upon those objects which are actually in motion and alone merits inclusion of this strategy into a more sophisticated low level system. For the purpose of tracking objects, however, it has a severe disadvantage. The intensity function over the image of an object remains almost constant within a small number of clearly defined regions. When an object is in motion, such regions become only slightly displaced between each scan, and many points contained within larger regions will, therefore, remain in the same region during several scans. Consequently, the gray levels from such points remain almost constant and do not become available

for the growth of change regions. In addition, change regions result not only from an object in its current position but also from its disappearance from a position in the previous scan. In extreme cases, this effect could produce a "double" image of an object. For the above reasons, objects in motion generate many small change regions distributed over the current and previous image of that object, in particular, around the boundaries of constant intensity regions. Associating these change regions with their generating object proved to be a formidable task. Partial success was achieved using a simple growth algorithm, although the resulting associations gave an imprecise spatial description of the object and combined distinct objects within a small distance of each other. The growth algorithm begins by selecting the largest change region as the kernel of an object. This kernel grows by engulfing change regions within a calculated distance, dependant upon its current spatial size. When growth can continue no further, the resultant kernel is interpreted as being the total result of an object in the scene. This process is repeated, each time beginning with the largest unassociated change region, until no change regions remain.

The fundamental problem appears to be knowing which points to update and which to leave alone. For each point, this decision is based upon the significance of the change region to which it belongs. To this end, we might consider separating objects in the scene from mere lighting changes

across arbitrary surfaces, by looking for edge features along the boundaries of change regions. However, this and other partial solutions require more operator analysis of the frame than was desired. The system, therefore, assumes a non-refreshed background, with the intention of refreshing chosen change regions by command of higher level procedures, where reliable significance can be attached.

2.3.1.1 A Low Level Motion Recognition Scheme

Based upon several ideas described by Lamontagne[8], a scheme was devised for recognising elementary motions of change regions within the low level system and, therefore, before associations between objects and change regions are made.

Given the change regions after a particular scan, each can be divided into two parts, either of which could be empty; those points which were also contained in a change region on the previous scan (E-type), and those which were not (N-type). Consider the change region in figure 5 resulting from a moving ball.

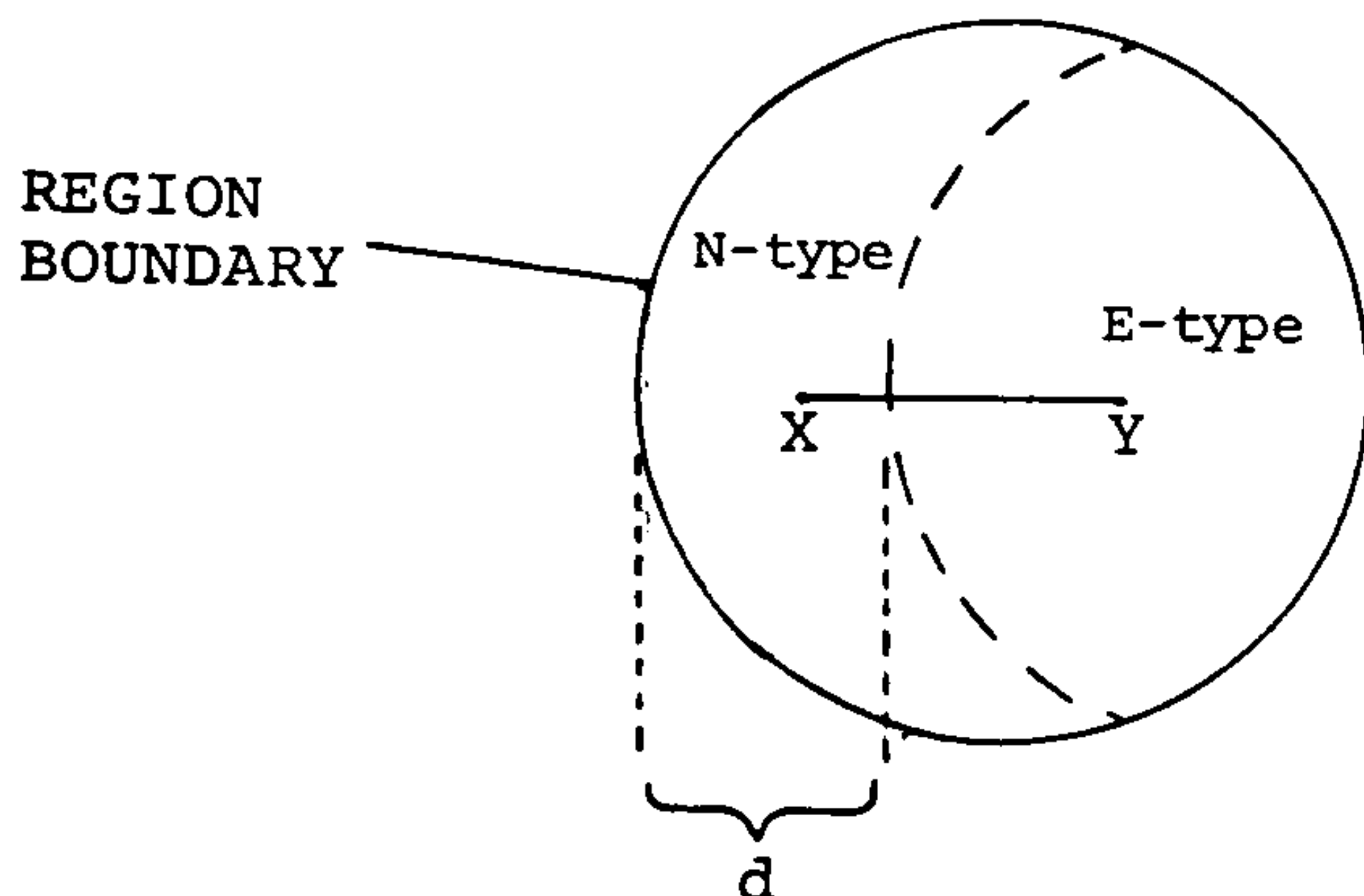


Figure 5. Recognition of elementary motions

X and Y are the centres of area for each subregion. By a simple argument, we may deduce that the direction YX and the distance d would be approximations for the direction of motion and distance travelled since the previous scan respectively. In general, however, these assumptions will not always be correct, especially if there is no obvious relationship between change regions from successive frames. If the subregion of E-type points is empty then the appearance of an object might be hypothesized, whilst if the subregion of N-type points is empty then a completely stationary object is suggested.

Attractive and simple to implement though this scheme is, equivalent information, deduced by higher level procedures from a more global standpoint promised to be more reliable and this scheme is, therefore, not employed in the present system.

2.3.1.2 A Manipulative Theory of Forms

Before continuing to describe how objects are tracked and velocities extracted from the stream of forms provided by FRAME, a manipulative theory of forms is proposed. First, we define a metric:

Definition:

Let A and B be forms, then

DISTANCE(A,B) :=

$$\max[(\text{if sign(DL)=sign(DR) then min[|DL|, |DR|] else } \emptyset), \\ (\text{if sign(DT)=sign(DB) then min[|DT|, |DB|] else } \emptyset)]$$

DL=LEFT(A) - LEFT(B)

DR=RIGHT(A) - RIGHT(B)

DT=TOP(A) - TOP(B)

DB=BOTTOM(A) - BOTTOM(B)

This metric was chosen for its extreme stability under changes in shape and size of a form, particularly important when the euclidean distances between objects are comparable with object dimensions.

Some means for comparing the size and shape of two forms is required. SIZEDIFF compares both properties simultaneously, paying due attention to each.

Definition:

Let A and B be forms, then

$$\text{SIZEDIFF}(A, B) := |(DXA - DXB) / (DXA + DXB)| + |(DYA - DYB) / (DYA + DYB)|$$

$$DXA = \text{RIGHT}(A) - \text{LEFT}(A)$$

$$DXB = \text{RIGHT}(B) - \text{LEFT}(B)$$

$$DYA = \text{BOTTOM}(A) - \text{TOP}(A)$$

$$DYB = \text{BOTTOM}(B) - \text{TOP}(B)$$

A form corresponding to the combination of two forms is needed, especially for the purpose of defining an object's representation.

Definition:

Let A and B be forms, then

$$\text{COMBINE}(A, B) := C$$

where C satisfies:

$$\text{LEFT}(C) = \min[\text{LEFT}(A), \text{LEFT}(B)]$$

$$\text{RIGHT}(C) = \max[\text{RIGHT}(A), \text{RIGHT}(B)]$$

$$\text{TOP}(C) = \min[\text{TOP}(A), \text{TOP}(B)]$$

$$\text{BOTTOM}(C) = \max[\text{BOTTOM}(A), \text{BOTTOM}(B)]$$

2.3.2 GROUP

This thesis will only consider groups consisting of single objects and pairs of objects, although extension to larger groups should be possible. GROUP maintains a distinct representation for objects and pairs of objects in packets placed into each partition.

2.3.2.1 Object Representation

Objects are represented by a single form, characterising the object's current spatial position in the scene. Both the current and the previous representations are recorded for the benefit of those procedures which recognise simple motion.

2.3.2.2 Pair Representation

Pairs are represented by pointers to the representations of the objects which constitute the pair and, in addition, the result of combining the representation forms of those objects.

For each group known to the system, the essential task of GROUP involves deciding which of those forms provided by FRAME were totally or partially caused by that group, "the resultant subset". Once determined, the only reasonable choice for a new representation form is the combination of this subset by repeated applications of COMBINE.

2.3.2.3 Object Maintenance - Tracking

In a simple situation, an object would cause a single form with truly representative spatial dimensions. In general, however, partial occlusion of the object or severe changes in the pronouncement of the object against its background will result in radical changes in the resulting form or fragmentation into several separate forms.

By the consistency supposition, we can be sure that all forms caused by an object in each new cycle, will be within a small distance of the current representation form, provided this form is approximately representative. Call the set of nearby forms F .

$$F := [X : X \in G \text{ and } \text{DISTANCE}(A, X) < d]$$

where G is the set of forms from FRAME and

A is the current representation form.

If F is empty, the object is assumed to have disappeared from view and an assertion to this effect is made in the associated packet. Should this assertion still remain on the following cycle, then the object and all pairs containing it, are forgotten by the system and their associated partitions deleted. However, this assertion may be removed by subsequent procedures if it is determined that the object is only temporarily lost from view, in particular, if it is occluded by another object.

Given that F is non-empty, GROUP looks for the object's resultant subset by generating all subsets of F and the corresponding combination forms, then selecting the most appealing. In fact, we need only consider combinations up to the fourth degree, four forms being all that are necessary to define the four spatial extremities of the representation form. Each combination is assigned a numerical value which reflects various "goodness of fit" criteria. However, before a best fit is selected, this same process is repeated for all other objects known to the system. All objects must be considered before any one object can be updated, to resolve problems of competition for the same component forms. From the resulting set of "best fits" the most convincing is selected and the corresponding object representation updated. The old current representation form becomes the previous representation form and the new "best fit" form becomes the new current representation form. The set of forms that

combined to form the best "best fit" form is subtracted from the available set provided by FRAME and the entire process is then repeated for the remaining objects, with the reduced set of available forms.

GROUP contains a fundamental fault which surfaces when two objects are in competition for the same form. In this situation the winner takes all, thus threatening the successful relocation of an object which may be quite justified in forwarding an, albeit weaker, claim. However, should such an object be lost by GROUP then it will be artificially relocated by the high level procedure OCCLUSION as if it had been totally occluded by the object with which it was in competition.

2.3.2.3.1 Criteria of "best fit"

In addition to limiting the distance travelled by an object between frames, the consistency supposition also places bounds on the rates of change of its shape and size characteristics. If the current object representation form is truly representative of the object, then consistency of size and shape must be maintained by the new representation form and its maximisation could indeed be the criterion for selecting such a form. However, it may happen that the current representation form be significantly unrepresentative of the object. In this situation, high level knowledge of the object in question could be used to

make decisions that may not maintain consistency between successive representation forms, but would, hopefully, better relocate the object.

The effect of maintaining an object representation based only on maintaining consistency of shape and size can best be illustrated by referring to figure 4, produced by the implementation and the period described in chapter 1. Consider the transitions between frames 13 and 17. In each case, consistency of shape and size is best achieved by associating all three change regions with the human representation form, therefore neglecting to realise that he has left an object behind and is rapidly leaving the scene. Since GROUP has no way of knowing whether or not it is maintaining a truly representative representation form or whether other knowledge the system may have is correct, a compromise between the two approaches must be drawn.

GROUP begins by applying previously acquired knowledge to the relocation process by comparing the size and shape of each combination form with that of a PREFERRED form, which may be associated with the object by any high level procedure. Should a combination compare favourably with the PREFERRED form, then that form is selected as the best fit and assigned the best possible numerical value to ensure success. In the case where no such form exists or when none of the combination forms compares favourably, GROUP selects the combination form which optimises shape and size

consistency. To this end, a numerical judgement is assigned to each combination form according to the amount by which its size and shape deviates from the current representation form, as determined by SIZEDIFF.

The high level procedure for recognising a human being is able to provide a PREFERRED form to GROUP based on its knowledge of the human "form" and a particular instantiation. Returning again to figure 4, GROUP succeeded in correctly associating only the right two forms in each frame with the human being, leaving a form trailing behind, to be subsequently recognised as a distinct object.

2.3.2.4 Pair Maintenance

Pairs of objects are considered after GROUP has serviced all objects. GROUP simply checks that each component object has been successfully relocated and updates the pair representation form accordingly. Should either of the objects have disappeared then an appropriate assertion to this effect is made in the associated packet.

2.3.3 CREATEGROUP

When GROUP completes its execution, there may still be one or more available forms from FRAME. In particular, these forms may have been caused by a new object appearing in the scene. The system must be capable of recognising this situation and adding the new object and any pairs of

objects its appearance creates to the set of known groups.

Recognition of a new object and creation of the appropriate partitions is performed by CREATEGROUP, executed immediately after GROUP. For each remaining, available form, CREATEGROUP hypothesises an object, represented by that form. If the hypothesised object appears in the following cycle as a single remaining, available form satisfying the consistency supposition by being within a given distance of the original representation form, then sufficient evidence is assumed for the addition of a new known object. In addition, pair representations are created for all pairs constructed by combining the new object with each existing known object.

For simple scenes involving a small number of significant objects, it is possible to represent all pairs explicitly. However, as the number of significant objects in the scene increases, so the number of pairs increases as the square, rendering this blanket approach impractical. As a possible alternative, the association between two objects could be retained but only as part of a group composed of more complex objects. For example, the walls of a room might be explicitly represented as individual objects but only appear elsewhere as part of a group corresponding to the entire room and not in an exhaustive set of pairs of objects and triples of objects, etc..

2.4 The High Level System

The most common type of high level procedure for recognising concepts involving change will be known as the demon. Demons recognise the successful recognition of two specific concepts in temporal succession, just as "left of" recognises a horizontal spatial order between two objects. To draw a comparison with the demons of Charniak[3], recognition of the first concept (precondition 1) should be seen as verifying the context in which the second concept (precondition 2) must be recognised. Successful recognition of the total concept causes an appropriate packet to be introduced into the representation (postcondition).

Properties within a packet serve only to distinguish between assertions and are otherwise uninteresting. For the sake of clarity no further reference will be made to this internal structure. As a single exception, the property STATE deserves mentioning. In general, if a procedure succeeds in recognising a concept, the value of property STATE becomes ACTIVE, otherwise it contains a state value denoting a partially recognised concept. For simple procedures, in particular for demons, STATE may be the only property in the packet.

As before, each procedure will be considered individually.

2.4.1 Locative procedures

Interesting locations in the scene may be associated with a locative type procedure. Such procedures recognise when an object comes within a predetermined distance of their particular location, represented by a form. The location can be relative to the T.V. picture such as "leftside of picture" or particular to the current scene as in "by the table". An alternative approach involves introducing objects into the scene as reference locations and relying on procedures recognising a concept such as "closeby" for the equivalent information. Although conceptually more satisfactory, this scheme is expensive if all we require is mere locative information.

Three elementary locative procedures, recognising proximity to locations relative to the T.V picture, are included in the implementation. LEFTSCENE, RIGHTSCENE and MIDDLESCENE recognise when an object is within a given distance of the left edge, right edge and middle of the picture respectively.

Transfer of spatial position between two locations known to the system is easily recognised by demon type procedures. In the implementation, RIGHTCROSS and LEFTCROSS recognise when an object horizontally crosses the picture in either direction.

RIGHTCROSS

Demon: precondition 1 - Object located at left of picture.
precondition 2 - Object located at right of picture.

LEFTCROSS

Demon: precondition 1 - Object located at right of picture.
precondition 2 - Object located at left of picture.

2.4.2 Recognition of Motion Concepts

Motion is represented in a very simple fashion, ignoring absolute values of speed or refined directional information. The reasons for this loose approach are twofold. Firstly, the calculation of absolute speed values requires accurate timing capabilities, possibly requiring buffering of information from the low level system and certainly increasing the complexity of the motion extraction algorithms. Secondly, the descriptive capabilities of the system do not require absolute values for their construction, but rather such vague assertions as "in motion" or "moving to the left". We shall only consider

motion in the horizontal direction although vertical motion could be treated in an identical fashion.

HMOTION

HMOTION determines the horizontal motion characteristics of each group. Groups are described as "moving to the left", "moving to the right", "remaining completely still" or if none of these is applicable, then as simply "stationary". These assertions depend upon the LEFT and RIGHT values of successive group representation forms as follows:

For the i th system cycle, an intermediate assertion (P_i) and a reliable assertion (Q_i) are made.

Let $DL = LEFT(A_i) - LEFT(A_{i-1})$

$DR = RIGHT(A_i) - RIGHT(A_{i-1})$

where A_i is group representation form for the i th cycle.

$$P_i := \begin{cases} \text{"moving to the left"} & : DL < 0, DR < 0 \\ \text{"moving to the right"} & : DL > 0, DR > 0 \\ \text{"remaining completely still"} & : DL = DR = 0 \\ \text{"stationary"} & : \text{if none of the above} \end{cases}$$

$Q_i := (\text{if } P_i = P_{i-1} = \dots = P_{i-m} \text{ then } P_i \text{ else "stationary"})$

where m is a system parameter.

HMOVE and NOTHMOVE

HMOVE recognises whether horizontal motion is present by considering the assertion of HMOTION whilst NOTHMOVE recognises the absence of horizontal motion.

The following demon type procedures recognise a group stopping or starting motion by considering the assertions of HMOVE and NOTHMOVE.

STOP

Demon: precondition 1 - group in motion.
precondition 2 - group not in motion.

GO

Demon: precondition 1 - group not in motion.
precondition 2 - group in motion.

2.4.3 Object Identification

Attempting to identify an object given only that information provided by FRAME from a single frame would be totally impractical and very prone to error. However, by utilising certain knowledge involving motion as suggested in chapter 1, the system identifies the most important object type present in any general scene, the human being. The primary step on the road to the identification of an object,

namely isolation of the object from the rest of the scene, is performed by the low level system.

STANDSHAPE

STANDSHAPE recognises the human "form" in an upright position by evaluating the ratio of the length against the width of the group representation form. Recognition occurs if this ratio lies within a given interval.

LAYSHAPE

LAYSHAPE recognises the human "form" in the lying position, again, by determining whether the ratio of the length against the width of the group representation form lies within a given interval.

HUMAN

HUMAN recognises that a group is a human being if it has the upright human "form", as recognised by STANDSHAPE, and is in motion, as recognised by HMOVE. That an object is in self-propelled motion is a strong clue towards the identity of that object, especially within a restricted environment. Although insufficient in themselves, the simultaneous recognition of the above two concepts provides substantial evidence for the presence of a human being.

Recognition by HUMAN of what GROUP believes is an object is consistent. However, it is possible for HUMAN to recognise what GROUP believes is a pair of objects or, more generally, any group which is not a single object. This would be inconsistent, for within the same instantaneous representation, HUMAN implies that the group is an object, whilst GROUP believes that the group is more complex. Identification of a group as corresponding to a particular object provides more substantial evidence for the existence of that object than does the technique of hypothesis employed by CREATEGROUP and the subsequent maintenance by GROUP. For this reason, the set of known groups should be modified accordingly. In this case, the pair representation packet is replaced by an object representation packet and the representations and partitions corresponding to the pair's component objects are destroyed.

After a human is successfully recognised, whenever the corresponding representation form is recognised by STANDSHAPE, this form becomes the PREFERRED form, for use by GROUP.

The following demon type procedures recognise an object going through the motions of "standing up" and "lying down".

STANDUP

Demon: precondition 1 - "LAYSHAPE" recognised
precondition 2 - "STANDSHAPE" recognised

LIEDOWN

Demon: precondition 1 - "STANDSHAPE" recognised.
precondition 2 - "LAYSHAPE" recognised.

INANIMATE

INANIMATE recognises an inanimate object by considering the assertions of HMOTION.

2.4.4 Object Interaction

OCCCLUSION

If an object cannot be relocated by GROUP then either the object has disappeared from the scene or behind a background structure in the scene or the object has been occluded by a known object. It is desirable to recognise the latter case and to relocate artificially the object so that correct associations will be made automatically should the object reappear.

This task is handled by OCCLUSION during the cycle of grace before GROUP has an opportunity to delete all knowledge of an object. OCCLUSION is interested in pairs of objects which have not been relocated by GROUP through one and only one of their component objects having disappeared. If, in addition, the successfully relocated object is spatially above the last known position of the lost object, then OCCLUSION asserts that the relocated object has occluded the lost object and removes the negative assertion made by GROUP.

ENGULF

Should an object disappear from the scene nearby another object, then ENGULF, executing in association with the pair consisting of the two objects concerned, interprets this as the latter object engulfing the former.

EXPEL

Should an object appear in the scene nearby another object, then EXPEL, executing in association with the pair consisting of the two objects concerned, interprets this as the latter object expelling the former.

PICKUP

If the engulfing object, as determined by ENGULF, is a human, as determined by HUMAN, then PICKUP asserts that the human has picked up the object.

PUTDOWN

If the expelling object, as determined by EXPEL, is a human, as determined by HUMAN, then PUTDOWN asserts that human has placed the object into the scene.

2.5 Natural Language Output

For the benefit of a human observer, the system produces a descriptive output whenever a significant concept is recognised. For such concepts, the corresponding procedures contain natural language output sentences with a number of key locative or proper noun phrases missing. By appropriately completing these sentences, they can be tailored to a particular situation and output to the observer.

2.6 The Implementation

Many of the illustrations included throughout the preceding pages were provided by an implementation of the ideas contained in this thesis. The implementation uses a Spatial Data camera, attached to a controller and monitor,

and driven by an Interdata 7/32 mini-computer. The entire procedure set, its utility functions and a kernel program are written in POP-10 (an implementation of POP-2 [2,4]) and reside on a DECsystem-10, except for FRAME which, although "surfacing" with the other procedures, communicates over telephone lines with its main procedure body, written in assembly language on the Interdata 7/32.

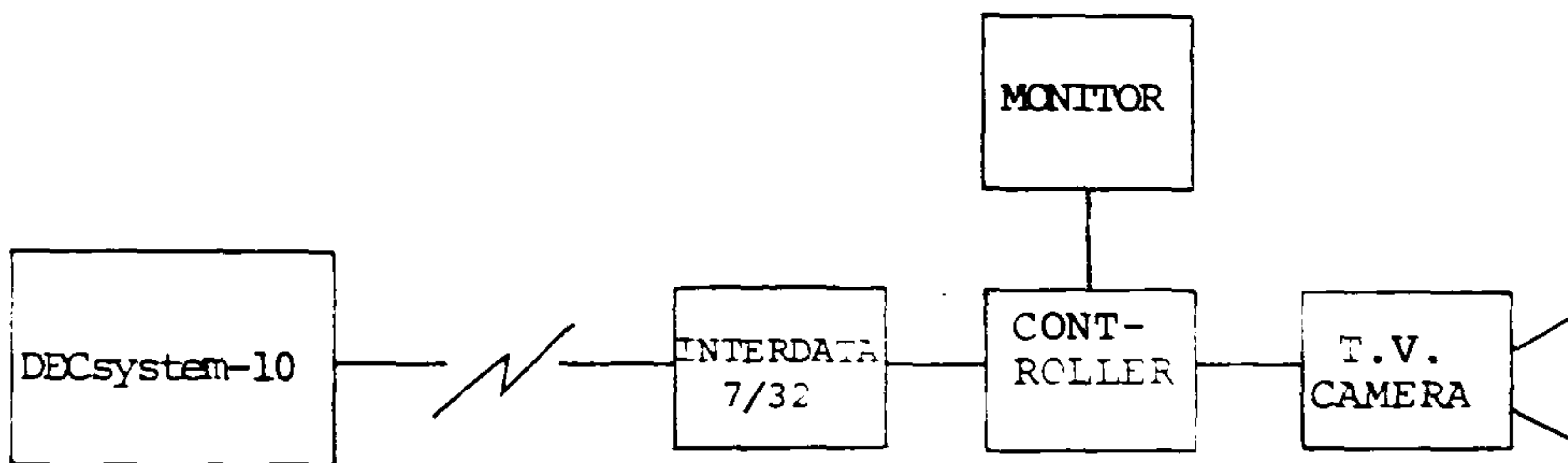


Figure 6. Configuration for implementation

To simplify the location of missing phrases for the output sentences, certain information produced by the locative and identification procedures is centralised in the packets introduced by GROUP. Most procedures can generate a descriptive sentence, the production of which may be turned on or off depending on the depth of description required.

The implementation performs adequately in real time with periods involving humans in the laboratory, provided the number of significant objects remains small.

The efficiency could be significantly improved by off loading the entire low level system onto the mini-computer and handcoding it in assembly language, thereby enabling the central program to concern itself only with high level concepts.

CHAPTER 3

Conclusions

The representational scheme employed by the system described in this thesis is firmly based upon those ideas discussed at the end of chapter 1 and at the beginning of chapter 2.

The system demonstrates that the construction of a real time representation based upon a sequence of representations for instances of the scene is consistent with the capabilities of a low level system and, indeed, complimentary to the generation of this preliminary representational level.

Instances of the representation are most suitable for real time applications satisfying the criteria deemed desirable in chapter 2. A detailed description of the current state of the world, including object locations, motion primitives and higher level situations in which these objects are presently involved, is immediately available to

an application process.

The simplicity of the low level system reaffirms that certain significant motion concepts can be recognised in a scene without the need for complicated analyses of instances of the scene. This is particularly true when the basic picture operators are subject to high level guidance, again demonstrated by this system.

Basing the system upon a set of procedures, each concerned with a particular concept and associated with clearly identifiable elements of the representation, has several advantages, developed to the full by the actor formalism proposed by Hewitt[6]. The flow of information between procedures lays bare the important aspects of the system's operation without having to consider the internal structure of each procedure. In addition, the organisation is extremely modular, enabling manageable subsystems to be identified and understood in isolation. The system can, therefore, be modified or expanded simply and without error.

Model driven analysis has received considerable attention in recent literature, particularly with relation to scene analysis. The system described in this thesis, demonstrates how top-down recognition can be effectively used for the purpose of tracking an object, provided sufficient knowledge is available on which to base the required inferences.

Certain recognition problems were resolved and general recognition performed efficiently by this top-down approach, although it is not clear that its ability to handle partial occlusion of an object by the background or deterioration of an object's resultant set of forms could not also be handled by optimising consistency, the system's, "backup", bottom-up approach.

Badler[1] suggests that objects might be initially located by a bottom-up analysis and subsequently tracked by top-down methods. The system generalises this methodology by enabling a bottom-up analysis to be performed whenever the required inferences cannot be made for a top-down analysis, in particular, when an object first enters the scene.

Part of the original objective was to design a system which could support applications such as security or night monitoring of a hospital ward. By adding appropriate application programs and making suitable modifications to the high level system, the current system could support such applications provided the external light sources remained constant and change was both infrequent and limited.

In summary, the methodology proposed by this thesis is consistent with current trends and observations in this field and proposes a truly practical system. Given the expected advances in hardware technology, the first of which are already on the horizon, future systems will undoubtedly

support more sophisticated picture operators. However, the present conceptual framework and procedural representation are essentially independent of these basic picture operators and can, therefore, provide a useful foundation for future research.

BIBLIOGRAPHY

1. Badler, N.I., Temporal Scene Analysis: Conceptual Descriptions of Object Movements. T.R.-80, Department of Computer Science, University of Toronto, 1975.
2. Burstall R.M., Collins J.S. and Popplestone R.J., Programming in POP-2. Edinburgh University Press, 1971.
3. Charniak, E., Toward a Model of Children's Story Comprehension. T.R.-266, M.I.T., A.I. Laboratory, 1972.
4. Davies, D.J.M., POP-10 User's Manual. Technical Report, Department of Computer Science, University of Western Ontario, 1976.
5. Dixon, A.H., Personal communication, 1976.
6. Hewitt, C., Bishop, P. and Steiger, R., A Universal Modular ACTOR Formalism for Artificial Intelligence. in Proc. 3rd I.J.C.A.I., pp. 235-245.
7. Jones, V.C., Tracking: An Approach to Dynamic Vision and Hand-Eye Coordination. Technical Report R-696; UILU-ENG 75-2231, Coordinated Science Laboratory, University of Illinois, 1975.
8. Lamontagne, C., A new experimental paradigm for the investigation of the secondary system of human visual motion perception. Perception, Volume 2, 1975, pp. 167-180.
9. McCarthy, J., Abrahams, P.W., Edwards, D.J., Hart, T.P. and Levin, M.I., LISP 1.5 Programmer's Manual. M.I.T., Comput. Center, Cambridge, Mass., 1962.

10. Michotte, A., The Emotional Significance of Movement. in Feelings and Emotions (ed. Reymont, M.L.), McGraw Hill, 1950.
11. Miller, G.A., English Verbs of Motion: A case study in Semantics and Lexical memory. in Coding Processes in Human Memory (ed. Melto and Martin), V.H. Winston and Sons, 1972.
12. Potter, J., The Extraction and Utilisation of Motion in Scene Description. Ph.D. Thesis, University of Winsconsin, 1974.
13. Sussman, G.J. and McDermott, D.V., CONNIVER Reference Manual. A.I. Memo 259, M.I.T., A.I. Laboratory, 1972.
14. Tsotsos, J.K., A Prototype Motion Understanding System. T.R.-93, Department of Computer Science, University of Toronto, 1976.
15. Weir, S., The Perception of Motion: Actions, Motives and Feelings. Research Report No.13, Department of Artificial Intelligence, University of Edinburgh, 1975.
16. Winston, P., The M.I.T. Robot. Machine Intelligence 7, 1972, pp. 431-463.

VITA

NAME: David Crossland Hogg

PLACE OF BIRTH: Leamington Spa, England.

YEAR OF BIRTH: 1954

POST-SECONDARY
EDUCATION AND
DEGREES: University of Warwick,
Coventry, England.
1972-1975 B.Sc.

University of Western Ontario,
London, Ontario.
1975-1976 M.Sc.

RELATED WORK
EXPERIENCE: Teaching Assistant
University of Western Ontario
1975-1976

Research Assistant
University of Western Ontario
1976 to date

389037

