Temporal Scene Analysis: Conceptual Descriptions of Object Movements

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/hms
Part of the Engineering Commons, and the Graphics and Human Computer Interfaces Commons

Recommended Citation

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/hms/201
For more information, please contact repository@pobox.upenn.edu,
Temporal Scene Analysis: Conceptual Descriptions of Object Movements

Disciplines
Computer Sciences | Engineering | Graphics and Human Computer Interfaces

This thesis or dissertation is available at ScholarlyCommons: http://repository.upenn.edu/hms/201
TEMPORAL SCENE ANALYSIS:
CONCEPTUAL DESCRIPTIONS
OF OBJECT MOVEMENTS

Norman I. Badler

Technical Report No. 80
February, 1975

A Thesis submitted to the School
of Graduate Studies in partial
fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
University of Toronto
Toronto, Ontario, Canada

Abstract

The dynamic nature of visual perception and the description of visual scenes motivate this investigation of a methodology for the conceptual description of changes in a visual scenario. A scenario (or motion picture) illustrates a natural, three-dimensional, man-made environment, and the proposed system aligns with two-dimensional coordinates of object features in an idealized, spherical projection of that environment. Object elements, indicated by changes in object feature locations, are described at successively higher levels by procedures which manipulate a graph-theoretic data base of objects and events.

The primary contributions of this thesis are the determination of a suitable representation for objects and events and algorithms for the generation of conceptual descriptions from a wide variety of scenarios. The event presentation is obtained by investigating the kinds of lower-level data that can be readily computed from the picture sequence. Events are contextually defined by the environment and consist of a hierarchy of concepts. The lowest level contains detailed information on object trajectory and rotation, intermediate level corresponds to certain adverbs and positions characterizing direction, and higher levels to the notion of repetition and event sequences. The highest level solves recognition of specific motion verbs. It is shown that there is a close correspondence between the event representation and a case structure for motion verbs in English, thus independently supporting the validity of the proposed system. Examples are given for several of its descriptive capabilities.
Acknowledgements

I wish to thank many people who have helped directly or indirectly to shape and guide this thesis.

I am very grateful to Jerry Feldman, my external reader, for his criticisms and discussion of this dissertation.

My thesis committee has always been helpful, encouraging, and understanding: John Mylopoulos, Ron Baecker, Les Mezei, and Derek Corneil. As my supervisor, John Mylopoulos initially encouraged my interests in artificial intelligence and suggested the problem of describing scenes. Our discussions have done much to clarify the present work and to define its context. As my second reader, Ron Baecker has vastly improved this thesis by his careful, critical readings of various drafts.

Bob Futrelle and Ron Baecker suggested the bouncing ball example. Tom Britton helped produce the film version of Example 1. John Tsotsos made many suggestions as he began implementing the system at Toronto.

Peter Roosen-Runge introduced me to computational linguistics which pushed this thesis beyond ad hoc descriptive results. The lpak development group, especially John Mylopoulos, Lou Nelli, and Phil Cohen, influenced the choice of the relational representation. Aravind Joshi provided the valuable reference to George Miller's work on motion verbs.

I am grateful for the generous financial assistance provided by Defense Research Board grant 9820-11, the National Research Council of Canada, the University of Toronto, the Department of Computer Science, and Henry Kramer. Otherwise this work could not have been undertaken.

Finally, I must thank my wife Ginny for her patience and encouragement throughout this effort, and my son Jeremy who arrived in the midst of it all bringing additional incentive for its completion.
To Jeremy
Table of Contents

I. Introduction.................................................................................. 1
  1. Relevance and contributions.................................................. 1
  2. Scenarios and assumptions.................................................... 8
  3. A sample scenario description.............................................. 28
II. A data base for temporal descriptions........................................ 33
  1. Pictorial data......................................................................... 33
  2. Conceptual data...................................................................... 36
III. A representation methodology for temporal information processing........................................................................... 39
  1. The representation of change................................................ 39
     1.1. States and changes.......................................................... 40
     1.2. Property queues.............................................................. 42
     1.3. Demons........................................................................... 45
  2. The representation of objects.................................................. 47
     2.1. TYPE............................................................................ 50
     2.2. SUB-PART..................................................................... 50
     2.3. VISIBILITY.................................................................... 52
     2.4. MORIETY....................................................................... 53
     2.5. LOCATION.................................................................... 54
          2.5.1. ADJACENCY.............................................................. 55
          2.5.2. SUPPORT-PLANF.................................................... 62
          2.5.3. ELEVATION............................................................. 63
          2.5.4. IMAGE.................................................................... 63
     2.6. ORIENTATION................................................................ 63
     2.7. SIZE............................................................................. 64
  3. The representation of events..................................................... 65
     3.1. Low level properties........................................................ 69
          3.1.1. TRAJECTORY.............................................................. 69
          3.1.2. REFERENCE.............................................................. 70
          3.1.3. VELOCITY................................................................. 72
          3.1.4. ANGULAR-VELOCITY and AXIS.............................. 73
     3.2. Higher level: DIRECTION................................................ 74
          3.2.1. Adverbial representation........................................... 77
          3.2.2. Directional adverbs.................................................. 78
     3.3. SUBJECT, AGENT, and INSTRUMENT............................... 92
          3.3.1. Actors and non-actors.............................................. 92
          3.3.2. A note on causality.................................................. 95
1.5.2. Plane maps ........................................ 1
1.5.3. Distance determination .......................... 2

2. Recognizing objects .................................... 2
   2.1. Structural models .................................. 2
   2.2. The recognition algorithm ......................... 2
   2.3. Object correlation .................................. 2

VI. Conclusions ............................................. 2
References .................................................. 2

Appendix I. Sample linguistic descriptions of events
Appendix II. English motion verbs
<table>
<thead>
<tr>
<th>List of Illustrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.1. The concept hierarchy</td>
</tr>
<tr>
<td>I.2. Example 1: A natural scene</td>
</tr>
<tr>
<td>I.3. Example 2: The bouncing ball</td>
</tr>
<tr>
<td>I.4. Example 3: Object interactions</td>
</tr>
<tr>
<td>I.5. Example 4: The walking man (after Muybridge)</td>
</tr>
<tr>
<td>II.1. The spherical coordinate system</td>
</tr>
<tr>
<td>II.2. A sample graph</td>
</tr>
<tr>
<td>III.1. PERSON object graph—parts</td>
</tr>
<tr>
<td>III.2. PERSON object graph—connections</td>
</tr>
<tr>
<td>III.3. Two cases of SURROUNDBY</td>
</tr>
<tr>
<td>IV.1. Trajectory derivatives</td>
</tr>
<tr>
<td>IV.2. The car and the house</td>
</tr>
<tr>
<td>IV.3. Car and house data base (time 1.00)</td>
</tr>
<tr>
<td>IV.4. Car and house data base (time 5.00)</td>
</tr>
<tr>
<td>IV.5. Car and house data base (time 7.00)</td>
</tr>
<tr>
<td>IV.6. Car and house data base (time 12.00)</td>
</tr>
<tr>
<td>IV.7. The ball data base (time 0.85)</td>
</tr>
<tr>
<td>IV.8. The ball data base (time 1.00)</td>
</tr>
<tr>
<td>IV.9. The ball data base (time 1.30)</td>
</tr>
<tr>
<td>IV.10. Levers and a pulley (after Sloman)</td>
</tr>
<tr>
<td>IV.11. Intermediate arrows</td>
</tr>
<tr>
<td>IV.12. Simple pulley</td>
</tr>
<tr>
<td>IV.13. Simple pulley representation</td>
</tr>
<tr>
<td>IV.14. Another pulley system</td>
</tr>
<tr>
<td>IV.15. Representation of Figure IV.14</td>
</tr>
<tr>
<td>IV.16. Other pulley arrangements</td>
</tr>
<tr>
<td>IV.17. Example 3</td>
</tr>
<tr>
<td>IV.18. Example 3 data base</td>
</tr>
<tr>
<td>IV.19. Lever patterns</td>
</tr>
<tr>
<td>V.1. Parallel lines in spherical projection</td>
</tr>
<tr>
<td>V.2. Intersecting arcs</td>
</tr>
<tr>
<td>V.3. Observer movement</td>
</tr>
<tr>
<td>V.4. Trajectory prediction</td>
</tr>
<tr>
<td>V.5. Compound rotation, vertical axis</td>
</tr>
<tr>
<td>V.6. Compound rotation, horizontal axis</td>
</tr>
</tbody>
</table>
Chapter II. **Introduction**

In thinking of language as a means of communicating ideas between people it is easy to overlook what was probably one of the earliest motivations for language. Were the world a stationary place there would be little use for language beyond filling in details of the static scene; communication would be optimized by a static representation — an ideogram, a drawing, or a photograph. However, the world we live in is a dynamic, changing place, whether or not we are the agents of the changes or observers, and in organizing this world into meaningful activities it is essential that language be capable of expressing dynamic processes. To this end it does so admirably with verbs, indicating processes, and nouns, indicating things. This thesis proposes a model for the organization of the visual world into conceptual structures and shows that the organization arises quite naturally from actual observations of objects and changes in the environment. The fact of living in a dynamic world has produced a conceptual structure fundamentally based on the description of visually perceived scenes in time.

II.1. **Relevance and contributions**

The famous Turing test [Turing(1950)] for the evaluation of machine intelligence, where a human experimenter attempts to discriminate man from machine on the basis of responses to interrogation, is susceptible to superficial aspects of conversation such as those exploited by ELIZA [Weizenbaum(1966)]. As research since then has shown, machine intelligence depends strongly on representational structures that allow understanding of a concept semantically, that is, in terms of a context of other concepts. In Winograd's blocks world [Winograd(1971)] there is just this kind of data base where facts and procedures have an equal role in understanding natural language descriptions and commands in a simple environment.

Although systems such as Winograd's knowledge-based blocks world have emphasized dialogues or question-answering in their
natural language abilities, it is interesting to investigate the generation of descriptions: someone has to start the conversation. In the blocks world the initial description is provided by the programmer, though ultimately it is to come from a robot vision system. In this branch of artificial intelligence research the problem of generating a description of a simple scene, let alone a complex natural environment, has proved formidable. The robot systems approach to descriptions of stationary scenes is exemplified by the projects at MIT [Winston(1970, 1972), Waltz(1972)] and Edinburgh [Lambert et al(1973)]. In a more natural environment than block worlds, the Stanford Research Institute group works with office scenes [Tenenbaum(1973)].

If a machine cannot match the descriptive capabilities of a human in a pictorial Turing test where it is shown a single photograph, then it would seem even less likely that the machine would do well if placed in a dynamic environment and asked to describe the activities perceived. Intuition notwithstanding, there are numerous reasons why such a pursuit is more likely to succeed: the possibility of deferring judgments, the computational advantages in multiple views, and the semantic significance of object movements.

This thesis will present a computational methodology for the description of events based on object movements. The temporal scene analysis will be performed on idealized image sequences from which object features have been extracted. We will not be concerned with low level picture processing or picture operators, but will assume that accurate image locations of point-like object features (such as vertices) are supplied with each frame. Changes in these features indicate movement of objects or the observer, and it is these movements which we want to describe.

The significance of the methodology is that changes are described at successively higher levels: levels closely related to the concepts people use to describe object movements. Although we are not claiming that language developed certain structures because people are "wired" according to our plan, we
believe that this thesis presents the first comprehensive computational framework for the simulation of English-like descriptions of the dynamic visual world. It is based on the transformation of pictorial and conceptual knowledge of objects in a natural environment into a case-structured [Fillmore(1968)] semantic network [Quillian(1969), Rumelhart et al(1972), Simmons(1973)] for a subset of motion verbs in English.

The primary contributions of this thesis are the representations for objects (section III.2) and events (section III.3) which provide a conceptual vocabulary for the description of changes in the environment. Object representations have appeared elsewhere [Winston(1970), Guzman(1971), Ambler et al(1973)], but none of these have considered object properties relevant to motion. Much more work has been done in graphical data structures, since object motion is an intrinsic part of dynamic display capabilities. These graphic systems, however, do not elaborate object descriptions beyond the need for coordinate transformations and compound motion of one part relative to the independent movements of another part. Two examples of such systems are SKETCHPAD [Sutherland(1963)] and GENESYS [Baecker(1969)]. The object data structure of the graphic system [Radler(1972)] which produced some of the scenarios illustrated here is similarly restricted.

Our representation for motion is unique. There have been very few attempts to describe motion in other than special-purpose applications, such as cloud tracking [Leese et al(1970)], certain medical image processing procedures [Ledley(1973)], and traffic flow analysis [Wolferts(1974)]. One attempt [Spegel(1973)] at descriptions of more general movements seems to have an awkward vocabulary and a non-intuitive descriptive procedure. The robot projects have seemed to avoid motion, mostly from practical problems arising from image analysis [Baumgart(1973)], but also because robot activities are usually based on state descriptions of static scenes [Winograd(1974), Fikes et al(1972)]. The notable exception is by Hendrix(1973), where linear programming methods are used to model continuous processes. A mobile robot with some cognitive
abilities is described by Schmidt (1971), but his primary goal is robot guidance in a stationary (roadway) environment.

Moving away from visual scenes, the representation of knowledge in question-answering systems must accommodate dynamic processes. The representation of time concepts is investigated by Bruce (1972), but repetitions are inadequately handled. General semantic networks [Quillian (1969), Winograd (1971), Rumelhart et al. (1972), Simmons (1973), Cohen (1974)] represent verbs and their cases, although the selection of cases is always based on an analysis of linguistic constructions.

Our contribution (section III.3) is the development of a particular case structure for a generic motion verb which is shown to arise naturally from the kind of data that can be obtained from visual images and knowledge of the objects in the scenario. It is thus tied to visual perceptions and conceptions about the world. Since the representation is a case structure, it has firm attachments to conceptual foundations of natural language. Our analysis receives independent support from a psychological investigation by Miller (1972) which yields a possible case structure underlying English verbs of motion. (Miller's list of such verbs, with some additions, appears in Appendix II.) Section III.3.5 shows the relationship between the two case structures.

Although our final descriptions resemble English sentences, there has been no attempt to do any general English generation from the semantic network [Simmons and Slocum (1972), Schank et al. (1973)]. We assume that this is possible by the choice of the representation.

In Chapter IV we show that the object and event (verb) representations do in fact lead to reasonable descriptions of a variety of scenarios. The descriptions were obtained by hand simulation of selected examples, but we feel that the process is straightforward enough that computer implementation would not have yielded fundamentally new insights into the problems involved. Given a data base of objects, their features, and spatial relationships to other objects, the motion concepts are determined by mathematical computations (section V.1) or by
pattern matching in the graph-theoretic data base (sections IV.1 and IV.2). The mathematical algorithms of Chapter V were implemented and tested; the general methods of Chapter IV have been implemented and tested in other contexts [Charniak (1972), Cohen (1974)], and have hopefully been well-motivated in the present discussion.

The final chapter shows that many of the fundamental concepts used in the description of motion can be determined from an idealized model of a single "eye." We give algorithms for computing, from spherical coordinates (section II.1) of object features: object surface slant, observer movements, object trajectories, object rotations, compound motions, and spatial relations. Some of these have appeared in Badler (1974). There are implications for scene analysis primarily in the determination of spatial relations such as used by Winston (1970). The second part of Chapter V describes an object recognition methodology for image sequences based on an algorithm for static scenes proposed by Barrow et al (1972). The major conclusion is that low-level differencing of pictorial structures, whether points, lines, or regions, is inadequate by itself for motion description. It can only say which portions of the picture sequence are changing, but the interpretation of these changes requires some higher-level knowledge about possible objects and their appearances in the scene.

Although only a limited number of examples are explored in this thesis, discussions throughout indicate the applicability of the various concepts and should leave the impression that the methods are not local, ad hoc techniques, but rather are of general scope within the specified restrictions (section I.2).

Based on the general discussion so far, we can list several goals for the scenario description system.

(1) Start with a data base of objects and object feature locations in each image.

(2) Produce a reasonable (and semantically correct) conceptual description of object or observer motions.

(3) Make the inferences between (1) and (2) as simple as possible consistent with adequate descriptive capabilities.

(4) Restrictions on object movements must be dictated by the naturalness of the objects, by the terrestrial
environment, and by the non-interference (or passivity) of the observer.

By using (1) we can effectively bypass the difficult problems which would arise from the consideration of real sequences of digitized images. We do not think this unreasonable, since people unconsciously organize the world into objects, unaware of whatever visual processes might be active.

The descriptions in (2) will be examined in section I,3 where an example can best motivate what we are aiming for. Our methodology for obtaining these descriptions satisfies (3) by describing changes (section III,1) level by level. Figure I,1 shows the organization of the conceptual hierarchy. Changes in object feature locations in the image sequence are described in terms of trajectories and rotations, or if some objects are known to be immovable, as observer movements. These trajectories and rotations are described with respect to the context of surrounding objects by certain adverbs and prepositional phrases called directional adverbials. They form the basis for events describing object motions. Certain condensations are applied to compact the event description, remove redundancies, or produce more natural descriptions. Sequences of events may produce higher level events such as repetitions. Finally, motion verbs may be recognized from the events if English-like output is desired.

It must be pointed out that a description of a scenario is different than an explanation of what was observed. The difference is one of level of interpretation. An explanation involves additional facts and inferences and is a higher level description. Our descriptions are inferred from pictorial information and certain object properties. In order to explain the motions of objects, we would need to develop a large database of possible semantic relations between objects and this is not the topic we are addressing. The proposed system would say that a dog is running and a man is running and the dog comes after the man, but saying that the dog is chasing the man requires much more knowledge about dogs and men than we will be using. These inference problems are beyond the scope of this investigation.
Figure I.1. The concept hierarchy

The consequences of the fourth goal will be elaborated as a list of assumptions in the next section.
1.2. Scenarios and assumptions

We will use a spherical projection (section II.1) of the three-dimensional world to produce some of the illustrated two-dimensional scenarios. There is no fundamental dependence upon this particular projection for the event description process, but it does provide several benefits in the acquisition of movement data from the scenario. Section V.1 discusses its advantages in detail.

Figure 1.2 shows a sequence of image frames from a motion picture. This example of a dynamic scenario will illustrate most of the assumptions the system will be built on.

Assumption (1) The input is a sequence of discrete, static images rather than a continuous signal.

This is hardly a descriptive restriction but more a practical one: if a scenario can be described then a motion picture of it can be. The immediate advantage is that there is a definite frame to analyze at each instant so that problems of actual real-time operation can be ignored (although real-time efficiency is still desirable). Separate frames also provide a natural internal time signal.

The disadvantages to this assumption are few but important for later extensions of the system. One is that a discrete signal (images) requires the sampling rate to be taken into account, that is, events having a frequency of occurrence at or below the sampling rate will be distorted or even masked between frames. For example, the railway signals swing with a period of two seconds; if the sampling interval (here one-fifth of a second) were lengthened to one second or more, the back and forth movement would be seriously impaired. The selection of a sampling rate is beyond the scope of this thesis.

Assumption (2) The sequence appears relatively continuous to a human observer.

That is, the image does not change "too much" between frames and could be projected as a motion picture. This is the case in Figures 1.2, 1.3, and 1.5. Cartoon strips are not generally of this nature.

Another disadvantage to this discrete image assumption is that certain physical events (none appear in Figures 1.2 through
Figure 7.2a. Example 1: A natural scene
Figure 1.2b. Example 1: A natural scene
Figure 1.2d. Example 1: A natural scene
Figure I.2e. Example 1: A natural scene
Figure 7.2f. Example 1: A natural scene
Figure 7.2q. Example 1: A natural scene
Figure 7.2h. Example 1: A natural scene
Figure 7.21. Example 1: A natural scene
Figure 1.2j. Example 1: A natural scene
Figure 1.2k. Example 1: A natural scene
Figure 1.21. Example 1: A natural scene
Figure 1.2n. Example 1: A natural scene
I.5) are more difficult to perceive in a frame-by-frame manner (see Futrelle [1974], for example). For the most part, however, we will avoid this situation by another assumption on objects which will be mentioned below. The third disadvantage is that the frames must be obtained instantaneously (as they effectively are in motion pictures) precluding simple adaptation to serial, real-time, video scanning of actual scenes. Blurring can be ignored since all the moving object images (car, signal, train) appear sharp and well-defined, hence features are precisely locatable.

Another assumption will be made regarding the input sequence:

Assumption (3) The input contains recognizable objects in a natural environment.

This system does not purport to describe any motion picture. The world will be populated with houses, people, cars, roads, and the like, as Figure I.2 suggests. By making this requirement the system will be able to use knowledge about objects and a terrestrial environment to describe events in common linguistic terms. The point is that language probably arose from a need to describe concrete entities and observations and that abstractions came later, although the original natural scenes would hardly have resembled Figure I.2.

The class of possible events will be restricted as follows.

Assumption (4) The integrity and form (in three-dimensions) of objects must be preserved, although jointed objects are allowed.

Thus no breaking, bending, melting, exploding, crumpling, folding, and so on, are allowed. These concepts involve changing physical characteristics of objects. What events remain involve translatory or rotational motions in three-dimensional space of rigid (Figures I.2 and I.3) or jointed (Figures I.4 and I.5) objects.

The observer, himself an object in the world (section V.1.2), is allowed to move.

Assumption (5) The observer is mobile, but passive, observing the world as a motion picture.

Since the observer is not assumed to be a physically mobile automaton, he need not know the actual distance he covers (in
the sense of measuring wheel rotations or the like. For the same reason, the observer is not assumed to possess any range-finding devices. The single eye must be used for depth perception (section V.1.5.3) although the technique of motion parallax can readily substitute for the lack of binocular vision should the observer's viewpoint change. Figure I.2 shows translation and rotation of the observer's line of sight. Figure I.5 admits two interpretations: either the observer is fixed and the man walks around him or else the observer is moving and the man walks beside him.

Each of Figures I.2 through I.5 illustrates a different capability a description-generating system should possess. In Figure I.2 all the objects move independently, the signals, the car on the road, and the train on the tracks. The observer, too, is de-coupled from any other object movement. Several types of motions are illustrated: linear translation (the train and the car), rotation about an external axis (the turning car) and about an internal axis (the swinging signals). The first two are also composed with translation and rotation of the observer. These are the fundamental motions the system must understand.

At a higher level, objects can interact with each other. For example, the hand and ball in Figure I.3 interact to cause the ball to drop while the ball and the ground interact to cause the ball to bounce. The interactions can also involve intermediaries, such as a bat which is used to hit a ball. Interactions are the most complex movements the system must describe, even though the individual objects exhibit the simpler fundamental motions.

In the limit, description may depend more on knowledge about the objects involved and their possible interactions rather than actual observation. In Figure I.4, if the man moves his hand downward, several things may move, one of which will be the block. The system can describe this situation were it shown the result as a movie, but it is also possible to use some knowledge about the movement-transmission properties of pulleys and levers to predict the result on the block. Section IV.3.2
Figure 7.3a. Example 2: The bouncing ball
Figure 1.3b. Example 2: The bouncing ball
will give a possible descriptive solution to this and similar problems, simpler versions of which are "classical" artificial intelligence problems [Sloman(1971), Baker(1973)]. By a "descriptive solution" we mean one in which the problem statement and the result are conceptual descriptions of the movements; moreover it is possible to do this without resorting to low level rotational formulas or the like. Problem solving is not, however, the primary thrust of the thesis, only a side issue.

Since object interactions are the most complex events we will deal with, we would like to generalize the set of objects to those having moving parts. This is motivated of course by

Figure 1.4. Example 3: Object interactions
the fact that people (jointed objects) are crucially important "real" entities in a natural environment and their motions cannot be ignored. An illustration from Muybridge (1955) (Figure I.5) provides some data on the walking motions of a man, requiring a discussion of the representation, recognition and description of relative movements (the arm with respect to the body, for example). For computational purposes we will use features extracted by hand from these photographs using the actual grid data.

As a final capability, certain activities are characterized by repetition (the signals, the bouncing ball and the man's arms and legs). Actual duplication of activities is not necessary as the bouncing ball shows: the height of each bounce is different. The system should recognize such simple instances of repeated movements and provide, for later development, a representation suitable for the discovery of much larger repeated units in the temporal scenario it observes.

We will often refer to Figures I.2 through I.5 as Examples 1 through 4, respectively, leading up to an analysis of all four examples in Chapter IV.

1.3. A sample scenario description

We will go through a simple example taken from the movie of Example 1: the swinging railway crossing signal. Throughout we will try to indicate where in the thesis individual algorithms or concepts are discussed so that this may serve as an introduction and guide. The question we want to answer is: How is the motion of the signal described from our observation of the movie?

In order to know that it is a signal in the first place, we need to have a model of what a signal looks like when stationary (section III.2). We recognize the instance of the signal in the first frame by finding its parts, the supporting pole, the cross bars and the arm (section V.2). We also expect to find it near the intersection of the road and the tracks, that is, at the crossing (section III.2.5). We know that the signal arm is a movable part (section III.2.4) of the signal, usually rotating
Figure 1.5. Example 4: The walking man (after Muybridge)
about the axis (section III.2.6) at the connection with the horizontal bar. We do not always need (or have) this explicit movement capability information. Fortunately the movement itself often provides it (section V.1.3.2).

At time 2.20, the second frame, the object recognition algorithm (section V.2.3) looks for the signal in the same place, but finds that the signal arm has changed location even though the rest of the signal has not. Since the signal is a fixed object (section III.2.4), the change in the image cannot be attributed to movement of the observer (section V.1.2), but must be caused by movement of the arm. A new orientation angle is computed from the displacement of the feature points of the arm (section V.1.3.2). The differing orientations of the arm indicate that it has rotated or turned.

The signal arm has started rotating. We can describe its direction of movement (section III.3.1) in terms of the rotation, CLOCKWISE (section III.3.2.2), and in terms of the spatial direction relative to the observer, LEFT (sections II.1 and V.1.5.1). In fact, since we see the front of the signal, we can even say it is MOVING RIGHT-SIDeways (with respect to its front, remember) (sections III.3.2.2 and V.1.5.1). We have already seen it START TURNING CLOCKWISE.

Between times 2.40 and 2.60 it appears stationary, that is, its image in frame 2.60 matches that in the previous frame 2.40. Hence it STOPS TURNING. We might say that the first event (section III.3) could be described as

1. The SIGNAL STARTS TURNING CLOCKWISE, MOVING RIGHT-SIDeways, then STOPS TURNING.

We have used TURNING and MOVING to indicate changes in orientation and location, respectively (section III.3.1).

At time 2.80 the rotation STARTS again, this time LEFT-SIDeways and COUNTERCLOCKWISE (section III.3.2.2). Between times 3.40 and 3.60 it is stationary, so STOPS is again appropriate. We now have

2. The SIGNAL STARTS TURNING COUNTERCLOCKWISE, MOVING LEFT-SIDeways, then STOPS TURNING.

The sequence (1) and (2) repeats several times. Notice that the duration of the rotational movement is not explicitly
mentioned in the description, though the observation times have been mentioned (section III.3.4). We are not worrying about finding "complete" cycles in the repetition (section IV.2.2). Statements (1) and (2) are the lowest level events (section IV.1) for the signal arm.

At an intermediate level there is a repetition (section IV.2.2) of the rightward and leftward movements which may be described as TO-AND-FRO (section III.3.2.2). We could say

3. The SIGNAL MOVES TO-AND-FRO, repeating the motions described in (1) and (2).

where MOVES is a general indicator of change. The description in (3) is at a higher level than (1) and (2) simply because they are sub-events of (3) (section III.3.4.2).

The description in (3) holds until after time 13.00, when the signal is seen in the same location in several frames, that is, the repetition has stopped as well as the individual rotations. Thus we should really say for (3):

3. The SIGNAL MOVES TO-AND-FRO, repeating the motions described in (1) and (2), then STOPS.

The highest level describes the signal's motion in terms of specific motion verbs (Appendix II and section IV.3.1). We saw TURNS, STARTS and STOPS in the three events above. We can obtain a more specific concept from (3), however, namely SWINGS. Because CLOCKWISE and COUNTERCLOCKWISE appear on (1) and (2), respectively, we choose SWINGS over an alternative RECIPROCATES. The latter could equally well describe (3) were the rotational movements not indicated at the lower level. Hence (3) becomes:

3'. The SIGNAL SWINGS TO-AND-FRO, then STOPS.

The use of SWINGS implies the rotation, while TO-AND-FRO remains in (3') to indicate the repetition. Thus (1) and (2) are redundant and (3') is a good description of the movement of the signal.

With the exception of other verbs inferred from the identity and function of a crossing signal (ANNOUNCES and INDICATES), neither of which are motion verbs, the description (3') is very similar to that used by human observers of the same scenario (Appendix I). That is, the most important concept was SWINGS, while TO-AND-FRO and STOPS were also mentioned. Some
observers interpreted the movement with a similar verb WAVES (though it seems slightly inappropriate (section IV.3.1.1)), the general verb MOVES, or the non-motion verbs ANNOUNCES and INDICATES. Even TO-AND-FRO seems to be considered redundant by most observers using SWINGS.

Soon we will see how descriptions such as these are obtained from object movements in image sequences. In Chapter IV the legs of the walking man of Example 4 will be used to illustrate the same descriptive process outlined above.
Chapter II. A database for temporal description

This chapter gives a brief exposition of some of the basic data structures we will use to describe picture sequences, objects and events. Implementation details for some of the higher level structures will not be necessary, nor are we concerned with data formats for some low level pictorial structures such as curves and regions. Considerations for the latter will be mentioned when required.

II.1. Pictorial data

In order to discuss certain aspects of the description process it is necessary to introduce the coordinate conventions used for sequences of images. Although the description generation process is not strictly dependent on any particular pictorial representation this one will provide us with algorithms for directions and trajectories, as well as some means for correlating successive images without constraining the observer to remain immobile. A full exposition of these properties is contained in section V.1.

Points of three-dimensional Euclidean space are denoted by triples \((x, y, z)\) in arbitrary units in orthogonal directions \(X\), \(Y\), and \(Z\) from an origin \(O=(0,0,0)\). A different coordinatization of the same space is based on angular measures from an origin \(V\). For now assume \(V\) and \(O\) label the same point. A transformation \(\Phi\) maps a spherical coordinate point, denoted by \((\Theta, \Phi, r)\) into its corresponding Euclidean point \((x, y, z)\) via

\[
(x, y, z) = \Phi((\Theta, \Phi, r)\).
\]

where

\[
x = r \cos \Theta \sin \Phi
\]
\[
y = r \sin \Theta \sin \Phi
\]
\[
z = r \cos \Phi
\]

The inverse transformation \(S\) is easily found to be

\[
(\Theta, \Phi, r) = S((x, y, z))
\]

where

\[
r = \sqrt{x^2 + y^2 + z^2}
\]
\[
\Theta = \arctan\left(\frac{y}{x}\right)
\]
\[
\Phi = \arccos\left(\frac{z}{r}\right)
\]
The geometry of this situation is illustrated in Figure II.1. It is convenient to consider the ray from the origin O through $(0,0,1)^0$ (where $\phi=0$ and $\theta$ is irrelevant) as the pole of the coordinatization and thus restrict $\phi$ to lie in the range $0 \leq \phi \leq \pi$. The radius $r$ is always nonnegative and $-\pi \leq \theta \leq \pi$. The expression "mod $\pi$" will be assumed to reduce an angle to this range of values. If $\theta$ and $\phi$ are individually expressed in degrees ($^\circ$), the concluding "$^\circ$" is unnecessary.

The following interpretation is given to this system. Let $V$ be the viewpoint of the observer. Without loss of generality assume $V=O$. Then $V=(0,0,0)^0$ is the center of the spherical coordinate system. Let $R=\{R(\theta, \phi)\}$, for $\theta$, $\phi$ restricted as above, be the set of all rays from $V$ to visible points in the space. For each such ray associate a light attribute (intensity or color) $L(R(\theta, \phi))$. For line drawings $L$ will be a binary predicate over suitably quantized $\theta$ and $\phi$ coordinate pairs.
Now consider the sphere defined by \( r = 1 \). Assuming that all the rays \( R(\theta, \varphi) \) have length greater than 1, define \( \text{Im} \), the image of the world, to be the set
\[
\{ ((\theta, \varphi, r)^0, L) \mid L = L(R(\theta, \varphi)) \}
\]
In general, \( \text{Im} \) is a function of \( V \), the viewpoint, and \( t \), the time of observation. If specification of either is required the notation \( \text{Im}(V, t) \) will be used. \( \text{Im} \) will often be called the spherical projection of the world visible from \( V \). Points of \( \text{Im} \) are characterized by a pair of spherical coordinates since the radius is always 1. This pair will be denoted by \( (\theta, \varphi)^0 \); it should be unambiguous in context. Thus
\[
\text{Im}(V, t) = \{ ((\theta, \varphi)^0, L) \mid L = L(P(\theta, \varphi)) \}
\]

The transformation \( P \) projects a spherical coordinate triple onto the sphere \( S \):
\[
(\theta, \varphi)^0 = P((\theta, \varphi, r)^0)
\]
The projection from \( (x, y, z) \) to \( (\theta, \varphi)^0 \) is now
\[
P(S((x, y, z)))
\]
or more simply,
\[
P_S((x, y, z))
\]
Note that \( P_S((x, y, z)) \) is just the "visual" location of the point \( (x, y, z) \) and \( L(P_S((x, y, z))) \) is the intensity of that point. If \( P_S(A) = (\theta, \varphi)^0 \), then \( P((\theta, \varphi, r)^0) \) gives the homogeneous coordinates of a point \( P_S(A) \) without the scale factor fourth coordinate (since the actual position of \( A \) in space is not known a priori). Homogeneous coordinates were used for scene analysis by Roberts(1965).

So far the definition of \( \text{Im} \) has not been bound to any particular environmental characteristics except point visibility. Some of these are easily incorporated into the model. Because of gravity the observer understands what "up" is. For the most part the ground he moves on is perpendicular to his up direction. Suppose that the Euclidean coordinates of this world are chosen so that the positive \( Z \) direction corresponds to "up." The environment important to the observer generally lies near the ground, parallel to the \( XY \) plane, and hence near the equatorial plane \( (\varphi = \pi/2) \) of the spherical system. The direction \( (0, \pi/2)^0 \) is in FRONT of the observer, \( (\pi, \pi/2)^0 \) is
BACK, \((-\pi/2, \pi/2)^{\circ}\) is directly to his RIGHT and \((\pi/2, \pi/2)^{\circ}\) directly to his LEFT. The direction \((0, 0)^{\circ}\) as UP has just been mentioned, similarly \((0, \pi)^{\circ}\) is DOWN.

The distance between two spherical projection points \(A^{(1)}\) and \(A^{(2)}\) is the arclength between them on \(S\). It is easily computed from the law of cosines as

\[
\arccos(A^{(1)}, A^{(2)}) = \arccos(VA^{(1)} \cdot VA^{(2)}) \\
= \arccos(\sin \theta(A^{(1)}) \sin \theta(A^{(2)}) \cos(\theta(A^{(1)}) - \theta(A^{(2)})) + \cos \theta(A^{(1)}) \cos \theta(A^{(2)})).
\]

Only monocular vision will be considered; in this thesis binocular vision will be unnecessary as much of the same information can be obtained quite readily from successive views based at spatially separated observer positions, or even from a single image. This will be discussed in section V.1.5.

The coordinatization \((\theta, \phi)^{\circ}\) can be imagined as a "flat" picture. If a window is placed over the sphere \(S\) so that only a small area is seen (say \(-\pi/4 \leq \theta \leq \pi/4\) and \(\pi/4 \leq \phi \leq 3\pi/4\)) the result will be a "reasonable" approximation to a standard \(XY\)-coordinatized picture array: \(X\) corresponding to \(-\theta\) (\(\theta\) increases from right to left) and \(Y\) corresponding to \(-\phi\) (\(\phi\) increases from top to bottom). Figures I.2 and I.3 were produced in exactly this manner using a \((\theta, \phi)^{\circ}\) window rather than the entire spherical view. This causes the apparent curvature of the straight lines; left and right are approximately at the left and right ends of each individual image in Figure I.2.

II.2. Conceptual data

Variables will be indicated by capital letter strings including the special symbol hyphen (-). Let \(U\) denote the space of possible values for a variable in a data base which includes as primitive data types real numbers, character strings, lists and nodes. A string or list of length zero is represented by the reserved variable NIL. Notation for lists will be LISP-like, thus

\[(A \ B \ C \ (D \ E))\]

however references to lists will normally be via a functional notation on the fields (elements) in the list. This is intended to correspond with programmer-defined data types in
The programming language SNOBOL [Griswold et al. (1968)] contains the types of structures required.

A graph consists of a set of nodes

\[ \{ N(i) \} \quad i=1, \ldots, n \]

and a set of edges

\[ \{ M(j) \} \quad j=1, \ldots, m \]

where the \( M(j) \) are lists in one of two forms.

The first form is a pair indicating an intransitive edge (a "loop" in graph-theoretic terminology):

\[ M(j) = p(N(i) \ r v) \]

where \( p \) is a string in \( \{ v \in \mathbb{V} \} \), and the left element indicates the source (and sink) node. There may only be one such \( M(j) \) associated with \( N(i) \) having the property name \( p \), and we can therefore refer to \( p \) unambiguously as a property of the node \( N(i) \) with value \( v \). The value will often be written:

\[ p(N(i)) = v. \]

The second form is also a pair and indicates a transitive edge between different nodes:

\[ M(j) = r(N(i) \ r N(k)) \]

where \( r \) is a string and \( i \neq k \). The left element indicates the source of the directed edge and the right element the sink. There may be only one such edge between \( N(i) \) and \( N(k) \) having the relation name \( r \), and we can therefore refer to \( r \) unambiguously as a relation between the nodes \( N(i) \) and \( N(k) \). The sink will sometimes be written:

\[ r(N(i)) = N(k). \]

For example, the graph in Figure II.2 consists of three nodes pointed at by variables HOUSE, CAR, and ROAD, and relations

\begin{align*}
\text{LEFT-OF} & (\text{CAR HOUSE}) \\
\text{SUPPORTED-BY} & (\text{CAR ROAD}) \\
\text{ABOVE} & (\text{CAR ROAD}) \\
\text{IN-FRONT-OF} & (\text{ROAD HOUSE}) .
\end{align*}

MOBILE and FIXED are possible values for the MOBILITY property of an object. Figure II.2 also contains a legend for the symbols used in data base illustrations. An arrow head at both ends of an edge indicates the symmetry of that particular relation.
Every node is assumed to have an intransitive edge with property KIND which can be used to distinguish different semantic groups of nodes in the data base node space.

Most of the remaining notation should be self-explanatory to anyone familiar with a modern high-level computer language. We will use "$:=$" for assignment to differentiate it from the relational operator "$=". Functions are written in the "function-name(argument, argument, ...)
form. Otherwise any special notation is explained when needed. The algorithms have therefore been kept as language-independent as possible.
Chapter III. A representation methodology for Temporal Information Processing

In the natural environments we are dealing with there are two sorts of entities, objects and events. The purpose of this chapter will be to define these entity types in terms of representational structures in a data base composed of graphs and lists. Primitive concepts for the representation of objects and events will be discussed here, although Chapter V will return to the problem of the recognition of these primitives from the image sequence. A discussion of the representation of change introduces the underlying description methodology.

III.1. The representation of change

The essence of time-dependent data can be summarized by the word "change." Images that do not exhibit any changes are "atemporal"; they may as well be a single image. A change implies a deviation or alteration from some recognized value to another. Thus a natural dichotomy in data arises: on the one hand specific data from the most recent pictures is being observed, on the other hand general patterns or processes must be synthesized from past data on what was observed.

In describing the events in sequences of static images, the knowledge of the present is of a different sort than knowledge of the past. For example, specific data on object location in the current image must be evaluated with respect to previous motion descriptions in order to determine whether the new data matches that description.
1.1. States and changes

The problem of representing changing data goes back to the roots of artificial intelligence research into problem solving. It can be thought of as an attempt to arrive at a given description (solution) from initial conditions (hypotheses) using selected observations (operators or rules of inference). Predicate calculus formulations for descriptive information ran into the problem of specifying what formulas (facts) were valid after rules of inference were applied. By introducing a state variable in each clause, assertions before and after rule applications could be distinguished. When applied, rules changed state variables so as to insure conflicting assertions differed in at least one literal.

This approach led to the well-known frame problem [Raphael(1971), Hayes(1971b)]. The first feasible programming solution to the problem of maintaining the slightly-changed data from state to state was given by Hewitt(1972) in PLANNER's state-structured data base. A refinement of this concept appears as the data structure offered by CONNIVER [Sussman and McDermott(1972), McDermott and Sussman(1972)]. The basic idea is that data is stored in a declarative manner in a data base state; when an inference is made or contradictory information provided or hypothesized, a new state is generated and only the changed data is stored in the new state. Older states are global to the new one so that nonspecification of a value in the new state assumes the last (most recent) value. In this sense the past states are a pattern (an explicit list) of values against which new values are checked for compatibility.

If we look at the situation from a linguistic point of view the notion of a state is ill-defined. We do not always describe scenes by saying what has changed, but usually by describing the changes. That is, not "the car was here, then it was there, now there," and so on, but "the car went from here to there." Examples of such descriptions, if they are not immediately obvious, may be found in Appendix I. An arbitrary change may perhaps be best described by the state approach: the state of the world before and after the change characterizes the change.
completely. In dealing with natural objects in natural environments, however, additional structure is imposed on the description, namely that some changes are more meaningful than others. If this is true we would expect some basic linguistic indicator of changes and a corresponding vocabulary of significant changes. The obvious candidate in English is the verb. A computational definition for a verb is that it is the name of a procedure which, given particular inputs and outputs, is capable of transforming the input into the output. Because of its descriptive usage, a verb is also the name of a pattern. We will give computational descriptions of several verbs of motion in English in section IV.3.1.1.

Because the concept of a verb can include only descriptions for which a given verb is known, a more general term will be used which at the same time incorporates the specificity of verbs and also the capability to describe more general, but still meaningful, changes or groups of changes. We will call this an event. An event describes some meaningful change which may or may not have a specific verb or verbs associated with it. It will turn out, however, that with an appropriate definition of "meaningful" the correspondence is surprisingly close.

We need a representation for an event. Going back to the definition of a verb, an event requires the specification of input and output states. But note that the output state can be taken to be the current (present) state of the world and the input state need not be specified if the event describes changes. Knowing the output or final state and the event, an approximation to the input state leading to the choice of that event description may be re-constructed. There is no need to explicitly store many previous states, and therefore little need to provide complex structures for them. In this regard a problem with descriptive ambiguity necessitating alternative descriptive states and hence representational structures for these alternatives may arise. A CONNIVER-like data structure would be useful in these circumstances. The ambiguity in the descriptions generated here is discussed in sections IV.1.3 and IV.3.1.1.
1.2. Property queues

The difference between state-structuring and the event representation is that static information in previous states is discarded in favor of the "name" of the change which describes consecutive state differences. There is a net loss of information in the abstraction of changes into higher level descriptions. Although some of the abstractions are reversible, most destroy some piece of data or other, such as an image location or actual trajectory angle.

An event will be represented by a node, named an event node (KIND(node)=EVENT), of an event graph. The various properties of an event node will be discussed fully in section III.3, but the structure of several of these properties will consist of a list of appropriate property values in temporally reverse order (newest in front). As data on object movements in the scene is obtained various abstracted concepts are pushed onto the temporal lists. For example, the following diagram represents an event node which we will see developed in section IV.1.4.2.

```
event B.3
SUBJECT BALL
AGENT GRAVITY
VELOCITY (*/+4*)
TRAJECTORY (*/(90°, 116.6°) (90°, 90°))
DIRECTION (\{TOWARD GROUND\} DOWN WESTWARD)
START-TIME 1.00
END-TIME (*/1.15 1.00)
```

The properties or relations are written to the left; their values to the right. The temporal lists are shown for properties VELOCITY, TRAJECTORY, and END-TIME. The "*" represents the front of the queue. All the notation will be explained in Chapters III and IV.

When new information signals that some meaningful event has been observed, the event node is split: the old one is terminated and a new one is spawned which inherits certain properties of the old, but indicates the differing properties of the new. This process is explained in Chapter IV. By choosing an appropriate set of termination conditions we can make events correspond to our own notions of what movements seem significant in a scene.
Rather than allow the description to extend backward in

time ("horizontally"), gaining in size and in redundant
information, the description becomes more abstract and compact
("vertically") at the expense of precision. For example, a
person walking will exhibit temporally regular movements of the
legs and arms as well as a specific forward-directed movement of
the body. Thus by only saying he is walking, we preclude any
possibility of saying where his left arm was at some particular
instant. We can strip away considerable lower level detail and
produce a compact abstraction of his activity for some period of
time.

At a lower level, however, the property list queues are not
allowed to grow indiscriminately. If we are only interested in
"significant" descriptions, much information obtained from the
pictures is only needed until higher level abstractions are
made. In most cases all that is required is a queue of at most
three elements: the front element is the new observed value of
the property (and acts therefore as a temporary variable to hold
this data), the second holds the current value of the property,
and the third holds the previous current value, if any. These
have corresponding names

NEW-property
CURRENT-property
LAST-property

where, if no prefix is specified, the value of a particular
property is its CURRENT value. Some properties will not require
this structure; the detailed section III.3 on the event node
representation will elaborate this introduction.

Besides reserving a place for new data on the front of the
property queue, this mechanism allows flexible communication
between processing modules. (The organization of these modules
is discussed in sections III.1.3 and V.2.) If one module
supplies low level data while others scan the event property
lists to find significant changes, then one can always know the
precise state of the other by checking whether the information
has been updated at the current clock (frame) time. The front
of the queue and the time of the last update compared to the
current time signals whether the property is "ready" for
abstraction; if the last update time differs the property is "busy." The next frame is requested when all active event nodes are ready.

It should be clear now that the purpose of the property queues is to allow direct comparison between current values (and their derivatives if they are time-dependent) and new data. The lowest level event nodes are constructed from information about the changing properties of the objects in the image sequence. In order to detect a change in object properties we must know what the object looked like in the picture before. Since objects change in their visual properties a representation structure analogous to that for events is proposed for them. An object is represented as an object node (KIND(node)=OBJECT), with a corresponding set of object properties.

Properties of an object and properties of an event will be distinguished by asking whether the property may be detected from a single frame. For example, size, location, and identity can be determined from a single image hence these are object properties. (This is not to say that in any particular image such information is obtainable, but rather that it is obtainable from some image, perhaps idealized.) On the other hand, movement, trajectory of movement, velocity and rotation are determined from sequences and are therefore event properties.

Nevertheless, certain object properties are advantageous to store as short queues. The previous picture coordinates of an object need to be saved as a CURRENT-LOCATION value, and the actually determined new position as a NEW-LOCATION. The difference between the two may lead to the generation of an event node to describe the transformation. But there is another important reason for saving LAST-LOCATION values as well. We will see in section V.1.3.1 that the pictorial location of points of objects moving in certain ways may be predicted on the basis of some conservative assumptions, and this prediction requires two known locations to "shoot" a prediction as to the third (next). Similar concerns apply for a few other object node properties, though prediction is less a factor.
1.3. Demons

In this section we describe the general process by which selected changes in the data base generate higher level concepts. The methodology utilizes the "demons" proposed by Charniak (1972). [See also Cohen (1974).]

A demon is a procedure which lies dormant until a certain set of assertions is present in the data base. We will refer to this set of assertions as pre-condition 1. When pre-condition 1 is satisfied (matched in the data base) the demon is activated. This places a demon into an active list. If pre-condition 1 is NIL the demon is always placed in the active list.

In order to execute the body of the demon (called the postcondition), another set of assertions must be present in the data base. This set is called pre-condition 2. If a demon is active (in the active list) and its pre-condition 2 is satisfied, it is executed. In Charniak's terminology, pre-condition 2 is its expectation.

There is a priority in the active list associated with the conceptual level of the demon. The list is organized so that lowest level demons appear first, higher level demons following in order. Thus demons for object recognition will be invoked before motion description is attempted. After those two levels the directional terms will be tried, then repeated events and description condensations, and finally motion verbs.

If a demon adds or removes information, demons of that conceptual level must be rechecked to insure that all relevant inferences are made. We will see this part of the process especially in section IV.3.2.

During the processing of the data from a single frame, a demon can only appear in the active list once with the same set of argument bindings (the objects or properties in its preconditions). Otherwise the demon would be continually re-executed, producing duplicate results. Of course, copies of the same demon may appear in the active list with different bindings.

Should the same demon be re-activated in the succeeding frame, it may determine for itself if the data it instantiates
is redundant or not. For example, an object recognition demon expects to be re-executed every frame to update object feature locations, even if they are unchanged. But a directional concept such as FORWARD may not be asserted again if it already exists in the appropriate place.

Once a demon is active it remains in the active list no matter how many frames are processed. A demon is returned to dormancy if its precondition becomes inapplicable.

The knowledge required by our demons and the scope of our problem domain makes our demon structure simpler than Charniak's for several reasons.

1. The hierarchic description isolates at each level (and within each level) certain sets of demons eligible for activation.

For example, demons which recognize objects are a different class than those which identify specific directional concepts, and those in turn are separate from ones which recognize particular motion verbs. This enables more efficient retrieval of the appropriate demons given the insertion or deletion of a certain datum.

2. Each set of demons is restricted in size by the vocabulary of concepts.

Thus we do not need endless "mini-worlds" to understand all the implications of a particular concept at this level because the concepts we are using are reasonably finite. We use about 50 directional concepts, less than half a dozen velocity concepts, and a small set of description construction and condensation demons.

3. The set of preconditions for each demon consists primarily of lower level assertions.

The hierarchy of concepts in the description of motion allows the set of precondition assertions to be taken from same or lower level data. Control of demon execution is such that preconditions of higher level demons are tested after all lower level demons have been executed. In general, demons add information at their own or higher levels. Circular chains of demon activations are thereby less likely.

-46-
III.2. The representation of objects

We will be dealing with objects which are free to move about and interact with other objects, but which are otherwise assumed to maintain constant (three-dimensional) shape, size, rigidity, weight and identity. Exceptions in the form of jointed objects (people, for example) are allowed if the non-rigidities are localized in point-like or hinge-like joints. Changes in any of these constant properties will be referred to as CHANGES-CHARACTERISTIC; these will not be examined in this thesis. However the properties themselves will be useful during the description building process.

The objects which the system recognizes are embodied in sets of nodes (KIND(node)=OBJECT-MODEL) which indicate the structural model of an object (section III.2.2) as well as additional properties or relations. Object models will be assumed to be innate knowledge in this system. Since the models are graph-theoretic, however, a learning scheme such as Winston's (1970) could be applied to model acquisition. The system would require "feedback" during the training phase from an external "teacher," and this is not incorporated into the design. Section V.2 describes how object models can be used to recognize objects in each image.

The instantiation of an object model by recognition of that type of object in a picture is represented by an object node. The properties on a object node are determined in one of three ways. They may be observed from the image coordinates of its visible features, computed by algorithms presented here and elsewhere, or copied from the model node. For example, the SUB-PART relations between object nodes are copied from the object model so that parts not actually seen can be added (for example, an arm on a person). Also, non-pictorial knowledge about a person can be supplied (what is his FRONT, what joints are movable, what is his approximate height and so on).

In actuality, copying values for every property for each instance of a model object node discovered can become quite costly in space, while copying so much data costs a significant amount of time. Thus a scheme of "exceptions" will be used;
properties that are likely to be constant from instance to instance of an object are retrieved from the model unless contrary indications are given on the particular object node being examined. References to the model supply most of the information about the object as a physical entity — size, shape, structure, mobility, directions of movement, and rotational axes. In addition, contextual information regarding its location or orientation may be stored in the model — a building must rest on the ground, a train is guided by rails, people usually have their head uppermost and walk on their feet, and so on.

The basic object properties of interest for our application are its SUB-PARTs, its MOBILITY, its LOCATION and ORIENTATION (since changes in these indicate movement), and its VISIBILITY (since occlusion may occur either through distance or by passage behind other objects). Of course, we will want to know the model TYPE of the object if it has been recognized (named). And, although it is not crucial to the basic descriptive scheme, knowing the approximate SIZE of an object is very useful. The combined set of LOCATION and ORIENTATION relations for objects will be called locative relations.

The following list summarizes the object node properties and relations and their values. Indented lists indicate subclasses of concepts. These will be discussed in the following sections.

**TYPE**

The object model node of which an object node is a recognized instance. If TYPE=NIL the object is unidentified.

**SUB-PART**

An object node corresponding to a named subpart of the object or else a locally important contact point set.

**VISIBILITY**

An object is either VISIBLE or NOT-VISIBLE. If occluded then the OCCLUDED-BY object is indicated.
MOBILITY
The movement capabilities of the object. The hierarchy of possible concept values is:

- MOBILITY
  - FIXED
  - MOBILE
  - MOTILE
  - ANIMATE

LOCATION
The spatial (ADJACENCY, SUPPORT-PLANE and ELEVATION) and image (IMAGE) locations of the object.

- ADJACENCY
  - NEAR-OF
  - OPPOSED-TO
  - AT
- CONTACT
  - SUPPORTED-BY
  - GUIDED-BY
  - CONNECTED-TO
    - CONNECTED-TO-RIGID
    - CONNECTED-TO-LINE
    - CONNECTED-TO-POINT
- SURROUNDED-BY
- CONTAINED-IN
- COVERED-BY

- SUPPORT-PLANE
  - LEFT-OF
  - RIGHT-OF
  - IN-FRONT-OF
  - IN-BACK-OF

- ELEVATION
  - ABOVE
  - BELOW

- IMAGE
  - FEATURE
    - POINT
    - LINE
    - CURVE
    - REGION
    - SURFACE

ORIENTATION
In the object model, intrinsic oriented features.

- FRONT
- BACK
- SIDE
  - LEFT-SIDE
  - RIGHT-SIDE
- TOP
- BOTTOM
- INSIDE
- OUTSIDE
- AXES

On an object node, actual directions of all these in space.

SIZE
A list of known distances between certain pairs of features of an object.
2.1. **TYPE**

**TYPE** indicates the model from which this object was recognized, for example, as a car, a person or a house. The value of **TYPE** must be contained in the object model. Properties not specified on this object node are determined from the object model node pointed at by **TYPE**. **TYPE** is a relation between an object node and a model node. The **TYPE** of a model node is a true property, indicating the string name of the object.

If **TYPE**(object-node)=NIL then this node represents an unidentified object (but it is an object); therefore an object is considered "recognized" if **TYPE**#NIL. **TYPE** will not require a queue structure; we are not interested in metamorphic transformations.

Since object nodes themselves provide unique "names," no further distinction need be made between objects of the same **TYPE** in the data base.

2.2. **SUB-PART**

Every object and part of an object that is recognized as such has an object node associated with it. The **TYPE** of a **SUB-PART** of an object is the object part model recognizing it. The decomposition of an object into parts is indicated by **SUB-PART** relations from the containing to the contained part, or equivalently, by **PART-OF** relations in the reverse direction. **SUB-PART** relations are, of course, transitive. The **SUB-PART** edges on the object model node define the object's structural model. We will see the role of the **SUB-PART** edges in object recognition in section 5.2.1.

Since an object may be a **SUB-PART** of more than one object, the **SUB-PART** edges form an acyclic directed graph. (No object can be a part of itself.) For example, in Figure III.1 a **SHOULDER** is a **SUB-PART** of both a **BODY** and an **ARM**. All **SUB-PART**s of an object are therefore reachable from a single node (called the **parent** object node) representing a single physical object. The **SUB-PART** edges partially order the parts of an object.
Note: All edges shown are SUB-PART relations.

Figure IIT.1. PERSON object graph—parts
The nodes at the end of SUB-PART chains will have lists of features: points, lines, regions and so on, stored in a property called IMAGE (section III.2.5.4). Although the decomposition of an object into parts can go as far down as the point, line and curve level, a division between higher level object nodes and these lower level pictorial structures will be made on the basis of which parts have common words associated with them. If a single term exists for some part then it is a candidate for an object node.

Under space or time constraints we might not generate nodes for all parts but only for those with some pre-determined significance. Thus for a person we might have lower level object nodes for named parts (Figure III.1). If certain parts are not of current interest, say toes or ears, instantiation of object nodes for these parts can be deferred. Deferment may also be based on moving objects, that is, no subpart of an object is given an object node unless that part is seen to move independently of the parent object. The arms of a person need not be considered separate objects as long as the person does not move them relative to his body.

Object parts without common names (points or lines, for example) may be necessary for the indication of points of contact or connection. (See sections III.2.5.1 and III.3.3.1.) These may be temporarily ascribed object node status while the contact relation applies.

2.3. VISIBILITY

Every object is either VISIBLE or NOT-VISIBLE at any one time. A partially occluded object is VISIBLE as long as it exhibits enough features in the image to identify them as belonging to the object. If there are insufficient features to make this association, the object is NOT-VISIBLE.

The occluding object is indicated (if known) by an OCCLUDED-BY relation from the occluded object. If the object disappeared because it left the picture boundaries or became too small, no OCCLUDED-BY relations are asserted.
VISIBILITY values may be queued; changes in VISIBILITY are significant events, but only if the occluded object was moving. (See section III.3.2.1: AWAY and BEHIND.)

2.4. MOBILITY

A given object in a particular context may have a restricted set of motions. It may be FIXED in one spot or it may be MOBILE. If MOBILE, it may or may not be capable of producing the observed motion (MOTILE versus NOT-MOTILE). If it is MOTILE it might be responsible for its own motion: ANIMATE. As the terms suggest, animals and machines are MOTILE, while only animals are ANIMATE.

MOBILITY is usually ascertained from the object model, depending mostly on the object's TYPE. Unfortunately the effective MOBILITY, that is the MOBILITY of an object in a particular situation, is determined by the set of locative relations currently asserted as well as some external knowledge such as KILLED or DISMANTLED or KAPUT. Even though a person is ANIMATE, he may lose some of his movement capabilities by constraining relations, for example, if he is strapped onto a bed. This problem of determining effective MOBILITY is strongly related to deductive problems of causality and ability [McCarthy and Hayes (1969), Hayes (1970, 1971a)]. The distinction between effectively MOTILE and NOT-MOTILE objects will be required higher up in the descriptive process (section IV.3), but for now we will assume that effective MOBILITY and actual MOBILITY are identical.

The MOBILITY property for each part of an object must be specified, for although movements are transferred hereditarily down connected parts, the MOBILITY property is not. For example, even though a person is ANIMATE, his arm is not: it must be CONNECTED-TO him at the shoulder via a joint so that it is only MOBILE (subject to the connection restrictions).

Only one part of a single object may be ANIMATE or MOTILE, the parent object node. Therefore any subparts must be MOBILE, even if they can never change their relation to the object they are part of. (They would be related by CONNECTED-TO-RIGID to
the next higher level object. A similar comment applies to subparts of a MOBILE object.

An object is FIXED if it is never expected to exhibit movement of any kind. Usually it is attached to the ground. Objects which are FIXED provide valuable reference features for correlating successive images and quantifying observer movement (section V.1.2).

FIXED objects may have movable subparts: a house can be FIXED while its doors and windows are MOBILE. Any object that is CONNECTED-TO-RIGID to a FIXED object must itself be FIXED.

2.5. LOCATION

If objects are moving in the scenario there must be a way to describe where they are located, both in the actual images and in space. The former is necessary so that the object may be looked for later; the latter is necessary because our perception of a three-dimensional world is in terms of three-dimensional locative relations. In the LOCATION property of an object both kinds of descriptions are stored. In a manner which should become clear as the description-building process unfolds, the naturalness of the final event descriptions depends strongly on good locational information.

In the object model, LOCATION indicates contextual guidelines on objects. These are guidelines, not absolute criteria, so that objects may be recognized "out of context." Examples of LOCATION guidelines on an object model might be

NEAR-TO(HOUSE ROAD)
AT(SIGNAL INTERSECTION)
ABOVE(BRIDGE ROAD)
IN-FRONT-OF(ENGINE BOXCAR)
GUIDED-BY(TRAIN PAILS)
CONNECTED-TO-LINE(DOOR WALL)
SUPPORTED-BY(CAR ROOF).

There are four classes of locational information, broadly characterized by the terms ADJACENCY, SUPPORT-PLANE, ELEVATION and IMAGE. The following sections will give basic characteristics expected from these terms and their subclasses. Algorithms for the recognition of some of these relations from the image or image sequence are given in section V.1.5.
2.5.1 ADJACENCY

ADJACENCY describes a metric or connection relationship between two objects. Rather than base its value on numerical distance, which would be an absolute quantity if it could be measured with certainty, concepts are chosen which permit relative (and hence subjective) judgments. The descriptions generated by these relative concepts are flexible enough to encourage this subjectivity.

The concepts placed under the property of ADJACENCY form the following hierarchy:

ADJACENCY
  FAR-FROM
  NEAR-TO
  AT
  CONTACT
  SUPPORTED-BY
  GUIDED-BY
  CONNECTED-TO
    CONNECTED-TO-RIGID
    CONNECTED-TO-LINE
    CONNECTED-TO-POINT
  SURROUNDED-BY
  CONTAINED-IN
  COVERED-BY

These are discussed in the following sections.

FAR-FROM, NFAR-TO and AT

The distance between two objects is classified as either FAR-FROM, NFAR-TO, or AT. All are asymmetric relations. Section 4.1.5.3 gives an algorithm for these concepts from other locational information.

The image size of the smaller object is compared against the spatial distance between the two objects. If one object is moving, then its image size is the basis of comparison, regardless of the size of the other. The reasoning here is that the stationary object will have a more constant image size since the image of the moving object grows and shrinks, especially near the observer, yet the actual metric separation may be changing only slightly.
CONTACT

A subclass of AT is CONTACT; an object is close enough to another to actually touch. It is obviously a symmetric relation. CONTACT relations are probably the most important information in a moving environment because they indicate object interactions.

The problem of determining CONTACT has been investigated by Fahlman (1973). His techniques are not always directly applicable in this environment because explicit three-dimensional models of objects are not assumed stored in the database. His simplest algorithm, relying on separating planes, requires three-dimensional coordinates of object vertices. No single algorithm will work for all static CONTACT relations when absolute three-dimensional information is not known, nor would we expect one to be found. Depth is crucial and therefore some, perhaps rough, three-dimensional model must be available. Two-dimensional image intersections are not sufficient for spatial contact.

CONTACT relations can sometimes be determined by a common boundary point of two objects, neither of which occludes the other. This is a special case of the AT algorithm, but it opens the possibility of a depth mis-judgement (alignment without contact). It can be used for situations such as simple object interactions (for example, the pulleys in Example 3).

Fortunately the situation with regard to CONTACT relations between moving objects is more straightforward. In section III.3.3.1 we will see how CONTACT relations may be inferred from changes in the path of an object. This type of CONTACT is more important to the movement description than the static type, because the latter are usually satisfactorily approximated by AT. The decision as to whether CONTACT did or did not occur between objects in a particular image sequence is the most common source of ambiguity in the description.

Within the class of CONTACT relations are subclasses indicative of constraints: SUPPORTED-BY, GUIDED-BY, and CONNECTED-TO. They relate motion of one object to another through definite physical interaction. For example, an object
SUPPORTED-BY a person's hand moves with the hand, his arm CONNECTED-TO his body also moves with him, and a train GUIDED-BY tracks follows their path.

When possible, CONTACT relations specify the parts of the objects in actual contact: the arm is CONNECTED-TO the shoulder, not the body; the board is SUPPORTED-BY the support at the intersection line. Parts of an object might require temporary promotion to object nodes in order to keep CONTACT a valid edge relation.

SUPPORTED-BY

An object which contacts an upward-facing portion of another object is SUPPORTED-BY that object. The normal to the point of the surface (section V.1.1) where contact occurs must have some vertical component in order that gravity may act and insure the support relation. This is not a sufficient condition, for example, counterweights may place an upward force on such a contact point. Balanced objects may have no net support either. Sideways pressure will enable fingers to grip an object where this SUPPORTED-BY definition fails to apply (Example 2). The intricacies of support will not be investigated here. (As was just mentioned, see Fahlman(1973).)

GUIDED-BY

These constraints are usually dictated by the object model, but must be associated with actual objects in the description. For example:

GUIDED-BY(TRAIN RAILS)
GUIDED-BY(CAR ROAD)
GUIDED-BY(WINDOW SLOTS)

The third example shows that guides need not provide support. (Assume the windows slide up and down).

The purpose of a guide is mainly to simplify the process of predicting movements, for if the direction of the guide is known the direction of movement of an object so constrained is determined. Guides are normally fixed in the environment and provide handy reference directions for a moving observer.
freeing him from some effort in correlating the successive feature images of the guided object.

Guides may be useful in determining objects to which an ALONG preposition may be related, enabling a more compact description of a path. The point of contact between guide and guided is assumed to move:

Trains moved along the tracks.
Cars drove along the road.

Otherwise another constraining relation (for example, SUPPORTED-BY, is acting):

The truck carried five cars.
The people were moved along by the escalator.

**CONNECTED-TO**

There are three forms of the CONNECTED-TO relation

CONNECTED-TO-RIGID
CONNECTED-TO-LINE
CONNECTED-TO-POINT

representing CONTACT through, respectively, a rigid bond, an axle or hinged joint (line), and a point attachment. CONNECTED-TO-RIGID transmits motion rigidly, as expected. A CONNECTED-TO-LINE relation implies that rotational motion is possible, although the joint may or may not be indicated as one of the AXES of either CONNECTED-TO object. (To see the latter case, consider a board which is just placed across a support to form a lever. Neither the board nor the support may have had AXES at the mutual line of contact. For the former case there are many examples: swinging signals, doors, elbows, and wheels on axles.) CONNECTED-TO-POINT is a similar relation and the same comment as above applies: the point of contact need not be intrinsically specified in either object model description but may be derived from observation of a connection. The difference between CONNECTED-TO-POINT and CONNECTED-TO-LINE is only that the movement of the former is not restricted to rotation in a single plane (perpendicular to the line): for example, CONNECTED-TO-POINT (RIGHT-UPPER-ARM RIGHT-SHOULDER) and CONNECTED-TO-POINT (ROOF CEILING).

Connections at points and lines would ideally require suitable descriptions of the permissible movements, taking into
account the local geometry of the objects in the vicinity of the joint. These will not be pursued here. The movements observed will be assumed "legal" for descriptive purposes.

Connections at joints of any other type must incorporate some non-rotational motion and are considered guides. Otherwise the joint would have to deform linearly, contrary to one of the basic assumptions of the system.

The modifiers POINT, LINE, and RIGID are used somewhat redundantly (the contacted parts of the two objects often provide the same information), but allow specification of the connection type without a continual re-evaluation of the contact geometry. The three types are also semantically distinct as constraining relations.

There are good reasons to separate PART-OF and CONNECTED-TO relations: the connection may be broken but the part is still associated with the object. A car may lose a tire or the knob might fall off a door, but technically these are still parts of their objects. Furthermore, any CONNECTED-TO relation is necessarily symmetric, while PART-OF is not.

But the most important reason is that the motions of various parts of an object are described hierarchically: for example, the hand of a walking person describes an arc about the shoulder, not some complex spatial curve incorporating the translatory motion of the body. These motion descriptions are assured by a tree of symmetric CONNECTED-TO edges defining the joints and rigid attachments between object parts related to the same parent node. Figure III.2 shows the connections for the PERSON object subparts of Figure III.1. Note that the connections of higher level parts (for example, BODY and LEFT-LEG) need not be specified if lower level subparts indicate them (LEFT-HIP and LEFT-UPPER-LEG).

The joint node of an object part is defined as the object part node or some SUB-PART descendant of it which is related by a chain of CONNECTED-TO edges to the parent object such that the chain does not pass through SUB-PART descendants of the original part. Thus the LEFT-HIP is the joint of the LEFT-LEG.
Note: All unlabelled edges are SUB-PART relations.

C-T-R means CONNECTED-TO-RIGID
C-T-L means CONNECTED-TO-LINE
C-T-P means CONNECTED-TO-POINT

Figure III.2. PFPSON object graph—connections
SURROUNDED-BY

One object is SURROUNDED-BY another object, part of an object, or a set of objects if it includes relations to them from one of the following sets of SUPPORT-PLANE and ELEVATION relations:

\{LEFT-OF, IN-FRONT-OF, RIGHT-OF, IN-BACK-OF\} \hspace{1cm} (1)
\{LEFT-OF, ABOVE, RIGHT-OF\} \hspace{1cm} (2)
\{IN-FRONT-OF, ABOVE, IN-BACK-OF\} \hspace{1cm} (3).

Any three relations from (1) suffice. The specification of BELOW in (2) and (3) will not be necessary as support constraints will usually supply that relation by default.

This definition admits some situations which fall on the border of "surroundedness," but the simple definition is worth the slight extension of the concept. Thus A is SURROUNDED-BY B in Figure III.3a because part of A is surrounded on three sides by B. Since there is no constraint on object boundaries, C is SURROUNDED-BY D in Figure III.3b.

CONTAINED-IN

There are two subclasses of the SURROUNDED-BY relation, one of which is CONTAINED-IN. Something CONTAINED-IN another object must be SURROUNDED-BY its INSIDE, implying of course that the object have an INSIDE (see section III.2.6):

The car contained five people.
The bottle contained a marble.

![Diagram](a)

![Diagram](b)

Figure III.3. Two cases of SURROUNDED-BY
COVERED-BY

The second subclass within SURROUNDED-BY is COVERED-BY. COVERED-BY is implied when sets (2) or (3) of the relations in the definition of SURROUNDED-BY are applicable. Note that if two opposite "arms" of D in Figure III.3b were missing, the COVERED-BY(C, D) relation would still hold.

2.5.2 SUPPORT-BLACK

Since the world is relatively flat and level, relations between objects on the same horizontal support plane locate these objects in a coordinate-free manner. The basic concepts are LEFT-OF, RIGHT-OF, IN-FRONT-OF, and IN-BACK-OF. Considerable advantages in using the spherical image model appear in the algorithms for these four relations (section V.7.5.1). The algorithm also generates gradations between these concepts, for example, IN-BACK-OF-AND-LEFT-OF, IN-FRONT-OF-AND-RIGHT-OF, and even SLIGHTLY-IN-FRONT-OF, LEFT-OF-AND-SLIGHTLY-IN-FRONT-OF, and so on.

The computation can also provide the relation independent of the actual observer position, dependent instead on environmental reference points or an intrinsic orientation of the object referred to. This often simplifies descriptions because observations of locational relations need not indicate the location of the observer. The indication of reference points is considered to be a property of events (see section III.3.1.2).
2.5.3. ELEVATION

Objects do not only have locations in support planes, they also have height or support plane elevation differences. The relations ABOVE and BELOW supply this information in a slightly restricted way. The restriction is that ABOVE and BELOW should apply only to objects occupying the same relative vertical location. Determining the appropriate relation in this case is trivial; the difficulty lies in establishing "same relative vertical location." This thesis will assume that such information is normally obtained from chains of SUPPORTED-BY relations, is discovered from different views through the spatial layout or depth algorithms, or else is "obvious" (an airplane ABOVE the ground).

2.5.4. IMAGE

This property will include any and all image-specific coordinates for FEATURES of the object. FEATURES may be any convenient pictorial structure: POINTs, LINES, CURVES, REGIONS or SURFACES. IMAGE values are saved in a property queue arrangement so that last values, as well as current values, can be used for location predictions and verification (section V.2).

2.6. ORIENTATION

ORIENTATION, as we will use the concept here, has two related meanings. The first is used on object model nodes to indicate the parts or features of an object which are associated with the views FRONT, BACK, LEFT-SIDE, RIGHT-SIDE, TOP, BOTTOM, INSIDE, and OUTSIDE, or are considered to be or indicate AXES. AXES may also consist of a view pair, if opposites, such as (TOP BOTTOM) or (FRONT BACK). The role of these views in object recognition is described in section V.2.1.

Only rarely will an instantiated object node differ in this property from its model, hence we will ignore the possibility. Usually the model will contain more properties rather than fewer, anyway. For example, an INSIDE to a house when the observer is quite unconcerned about whether a particular house
be seen has an INSIDE or not. Exceptions arise mostly with AXES; some may be observed in the object that were not expected from the model — particularly if the object is in free fall or is otherwise moved by external forces. Therefore local constraints on objects such as CONNECTED-TO-LINE relations will indicate axes of rotation.

The second meaning for ORIENTATION therefore applies to object nodes rather than object models. It is a mapping from the idealized FRONT, BACK, LEFT-SIDE, RIGHT-SIDE, TOP, and BOTTOM views into a set of direction vectors in space. When the object is recognized, the views which are seen enable a rough estimate of the direction of these parts with respect to the observer's frame of reference (see section III.3.1.2). The location of the observer determines whether the INSIDE or the OUTSIDE was recognized, thus these do not have directional values. If the observer or another object is on the inside of this object, then it is SURROUNDED-PY or CONTAINED-IN the object.

2.7 SIZE

The SIZE property of an object is usually specified in the object model, if at all. It is a list of feature points of the object with known distances between them. The most useful measurement is the height of the object, if it has a size comparable with the height of the viewpoint above the ground.

If SIZE values are known with fair accuracy, depth and distance determination becomes easier (see section V.1.5.3).
III.3. The representation of events

In order to discuss the description of events based on motion, a representation is presented for an event node. The creation and maintenance of event nodes is described in the detailed algorithms in Chapter IV. These algorithms will be taken as the definition of what constitutes an event. Before that, however, the properties of the event node will be explained and this should give a reasonable impression of what things are relevant to an event.

In the following the SUBJECT of the event will be the object whose movements are being described. We will return to this later, but the notation is useful now.

Since verbs are used to describe changes in the environment of objects, we require a generic [Miller (1972)] or canonical [Hendrix et al. (1973)] verb to describe motion. Following Miller's analysis, the choice of CHANGES-LOCATION might be sufficient; we could use TRAVELS or the composite concept COMES-GOES just as well. Schank's (1973) concept PTRANS (change of location) could be used also.

Care must be taken to define "location" in these concepts, however. Regardless of whether we take the definition of a location as a point or a volume in space, there is still a class of movements for which CHANGES-LOCATION does not sound like an adequate characterization: for example, a sphere may be spinning but it would not be changing location. Since Miller is mainly concerned with actions of animate objects, it is understandable that this distinction would not be as important to his analysis, although he claims to provide a case structure for such verbs as SPINS, ROTATES, TWISTS, and so on. He does this by defining CHANGES-LOCATION as a paraphrase for CHANGES-LOCATION-OR-POSITION, thus embedding the orientation (position) changes of an object into a new (implicit) definition for location.

The distinction in the literature between "location" and "position" is not at all clear. Miller does not define "location" or "position" at all. Martin (1973) defines the two indirectly by saying:
We talk of the size of a location, but not the size of a position. A position is where an object is placed; we adjust a position, not a location. (page 6)

Presumably Martin means that a location refers to a spatial point, area or volume, so that we can talk of its extent. A position is a relation between two objects; the relation is spatially dimensionless and it can be adjusted. A change in position presupposes a change in location, while a change in location may or may not result in a change of position; thus the concept of location is more general. Consider a block on a table: if the table is moved, the block changes location (it moves, too) but not position (with respect to the table); but if the block is held while the table is moved, the block does not change location (it does not move) yet its position varies. Movement of the block is therefore signalled by a change in spatial location because a change in position may or may not be evident depending on the context.

In the following discussion, location must not be confused with relative notions of where an event transpires. Location is not a property of events, but of objects. A moving object may exhibit a change of location; it is this change and its characteristics which are significant for the event. Similarly, locative relations (CONTACT, CONTAINED-IN, SUPPORTED-BY, etc.) between objects during an event are properties of the objects involved, are implicit in the description of the actors and non-actors in the scenario (section III.3.3.1), or are transformed into directional relations (section III.3.2).

The possible ambiguity of "position" is eliminated by using the concept of "orientation," that is, the directions of features of an object with respect to some fixed frame of reference. This certainly accommodates the class of motions for which CHANGES-LOCATION might be construed to fail. Since OR is not meant to be exclusive, the final generic concept for spatial movement is CHANGES-LOCATION-AND/OR-CHANGES-ORIENTATION. Were the restriction on rigidity and integrity of objects dropped, for example, to allow them to break or melt or bend, then this augmented class of perceived movements would require the extension of the concept by -AND/OR-CHANGES-CHARACTERISTIC.
The event representation must simultaneously store information for both changes in location of the moved object and changes in its orientation. There are two sets of properties for storing information with respect to these two categories. For changes in location the event requires a TRAJECTORY and a VELOCITY; for changes in orientation, an ANGULAR VELOCITY about some AXIS. Because angular changes and trajectories are not in themselves normally characterizations of meaningful relations in conceptual descriptions of events ("The car is moving with heading (80°, 90°)") an additional property DIRECTION is included. DIRECTION is intended to be a first step toward higher level characterizations of sets of relationship changes perceived in the scene.

This list summarizes the event node cases to be discussed in the following sections.

SUBJECT
AGENT
INSTRUCTION
REFERENCE
DIRECTION
TRAJECTORY
VELOCITY
AXIS
ANGULAR VELOCITY
NEXT
START-TIME
END-TIME
REPFAT-PATH

Basic definitions for event node properties and some of their admissible values are presented below.

SUBJECT
An object node or list of such nodes corresponding to the object(s) moved. Note that rigidly attached subparts of an object do not appear in the list since motion of the dominating object implies the same underlying motion for its lower level parts.

AGENT
An object node or list of such nodes corresponding to a motile agent contacting the SUBJECT.
INSTRUMENT
An object node or list of such nodes corresponding to a moving object contacting the SUBJECT.

REFERENCE
A pair of object features in a horizontal plane which are used to fix absolute directions in the environment independent of the observer's frame of reference.

DIRECTION
A temporal list of directional adverbials and their associated objects. The set of possible concepts is given in the section on adverbials.

TRAJECTORY
The angular direction of change of LOCATION of the SUBJECT.

VELOCITY
The approximate magnitude of the velocity of the SUBJECT along the TRAJECTORY; the property includes a RATES list containing STARTS, STOPS, INCREASES and DECREASES.

AXIS
For an ORIENTATION change, AXIS is the approximate direction of the axis of rotation.

ANGULAR VELOCITY
The approximate magnitude of the angular change per unit time of the SUBJECT around the AXIS; the property includes a RATES list containing STARTS, STOPS, INCREASES and DECREASES.

NEXT
The event node which immediately follows this event node in time and has the same SUBJECT. There is an implicit PREVIOUS property encoding the inverse relation.

START-TIME
A number or a pointer into a time line giving the onset of an event. If this event forms part of a REPEAT-PATH list, then this event will have been observed more than once. In this case START-TIME will be that of the last occurrence.
END-TIME
Same as START-TIME except that it indicates the termination time of the event.

REPEAT-PATH
A list of event nodes which form a repeating sequence. The list must be ordered by a chain of NEXT edges, where the END-TIME of one is the START-TIME of the next.

3.1. Low level properties
Changes in object feature locations activate the demons which compute trajectory directions and rotation angles. Some estimate of the velocity may also be available. Low level data on these primitive motions is placed in event node properties where it can be analyzed by other demons.

3.1.1. TRAJECTORY
The TRAJECTORY is the direction of change in location of the SUBJECT and may be computed by one of the algorithms presented in section V.1.3.1. The direction is a spherical projection image point and represents the heading relative to a fixed coordinatization of the world at the point of the observer.

When the trajectory of the SUBJECT is not linear it will be necessary to discuss the change of trajectory: its derivative. Unlike changes in speed (acceleration), pictorial changes in trajectory may be quantified to a reasonable degree of accuracy. Enough information for the computation of derivatives is contained in the queue of values for TRAJECTORY and END-TIME (see section III.3.4.1).
3.1.2. REFERENCE

It would be nice if the TRAJECTORY were independent of the particular location and orientation of the observer so that he could move in the environment, but this requires knowledge of an intrinsic coordinatization for the environment. The solution is to assume the world to have a coordinate system based on directions analogous to the spherical coordinate system of the observer and to assume that the observer's eye moves along lines parallel to the ground.

A direction in the ground plane must be selected as one reference axis and another in an orthogonal direction as a second. It is natural to choose one reference direction as NORTH and another as UP. To complete the coordinatization we define a spherical system \((\theta, \phi)^{+}\) with NORTH as \((0, \pi/2)^{+}\) and UP as \((0, 0)^{+}\). There is no difficulty defining the directions for SOUTH, EAST and WEST (assuming a right-handed coordinate system). As a result we have a coordinate system for the environment independent of the location of the observer, provided he knows one reference direction (NORTH), since UP is quite distinct. In fact, as far as an observer of any scene is concerned, what is NORTH is completely arbitrary, although once chosen it must be adhered to for proper orientation.

We should note that although the reference should be chosen on the basis of fixed visible features, there is absolutely no requirement that these features remain in sight. They only serve to initially orient the observer's frame of reference \([ (\theta, \phi)^{0} ] \) to the scene; from there he can maintain the reference merely by noting any changes in his own orientation. Changes in his location are irrelevant because the directional coordinates in the world are valid for infinite lateral motion in the ground plane, that is, what is NORTH at one spot is NORTH miles away. For "small" vertical displacements of the observer, the \(\theta\) component will remain the same although \(\phi\) will change slightly.

If the observer has faith in the reliability of the orientation of his own coordinate system then no reference direction need be chosen at all; it can simply be the observer's own \(\theta = 0\), that is, NORTH is \((0, \pi/2)^{0}\).
The only information needed to reconstruct an actual direction is just the reference direction NORTH. (UP is always assumed known.) By our assumption of a passive observer, we can rely only on visual features as a baseline. We just saw that in fact only one visual reference line was required. This is the function of the event node property REFERENCE. REFERENCE points to two object feature points lying in the same horizontal plane. No numerical description can be stored, precisely because these points anchor the coordinate system. Instead the physical relation they have to an identifiable object serves to make them locatable (if the scene is viewed again), and therefore to reorient the observer. If no REFERENCE is specified the observer's frame of reference is assumed, but this makes the event depend on the current orientation of the observer.

What all this has to do with TRAJECTORY should be more apparent now. The TRAJECTORY of the SUBJECT is computed relative to the world coordinates $(\theta, \phi)$ and it is this direction which is stored in the TRAJECTORY property.

In spite of the necessity of orienting a particular event, REFERENCE is unimportant at higher descriptive levels. People seem to prefer descriptions of trajectories relative to objects rather than absolute directions, relying on the latter only when no other fiducials are available (or identifiable, as during direction-giving conversations). See the explanation of DIRECTION below.

REFERENCE will be used here mostly to allow spatial and temporal discontinuities in the visual scenario. If the observer moves any significant distance he is better off reorienting himself to the new environment rather than attempting to describe the new one in terms of the coordinatization of the old. This is especially crucial for the passive observer of a motion picture where the "in-between" portion of a change in location or orientation may not be seen at all. When a new scene is presented a REFERENCE must be determined. Any events perceived until the next spatial or temporal discontinuity will then be anchored to the same REFERENCE value. This allows freedom in orienting descriptions to the particular scene being
viewed. If, after being shown a scene of a building, we are placed inside someone's room, it is doubtful that the horizontal orientation of the room to the outside is initially of importance to a description of the events transpiring within. If the relation between the two places does become known, say by looking out of a window, then we only need to rotate all trajectories (not already described in direction lists) by the orientation difference. (See section V.1.2.)

3.1.3. VELOCITY

As the SUBJECT moves along in some TRAJECTORY it has a velocity whose magnitude is saved in the property VELOCITY. Only nonnegative values are stored. A VELOCITY of zero means that the SUBJECT is not changing location; positive, it is. VELOCITY therefore signals CHANGES-LOCATION. Negative velocities are unnecessary because knowing the TRAJECTORY and ORIENTATION of the SUBJECT, the direction may be described at a higher level as FORWARD, BACKWARD, SIDEWAYS, etc. Thus a car goes BACKWARD with positive velocity rather than FORWARD with negative velocity.

By approximate magnitude we mean that the velocity need not be expressed in absolute units such as miles/hour or the like, simply because such values are difficult to obtain accurately from the picture sequence unless a SIZE value for a moving object is known. Rather some arbitrarily quantized ratio of distance to time can be used. In section V.1.5.2 we will give an algorithm for spatial layout that will allow the specification of a single reference distance to fix certain distances in the remainder of the scene (at least in a horizontal ground plane). If SIZE is unknown then an acceptable approximation is simply the length of the displacement (of the center of mass, say) divided by the time interval. Even these numerical quantities may be unnecessary where the velocity is irrelevant and only the raw movement is important. A "+" will be used in this case.
Besides the numeric values or "+", there are four concepts recognized by simple demons. STARTS, STOPS, INCREASES and DECREASES will form their own list following the numeric list:

(NEW-VELOCITY CURRENT-VELOCITY LAST-VELOCITY RATES).

RATES is a push-down list describing the form of the VELOCITY changes: STARTS and STOPS for changes from or to zero; and INCREASES and DECREASES for significant changes from the current (non-zero) value, if these are desired in the description. For example, the successive velocities (0, 4, 8, 8, 5, 0) might appear as

(0 5 8 (STOPS DECREASES INCREASES STARTS))
after the presentation and description of the final value. If the values could not be measured accurately, the sequence might have been (0, +, +, +, +, 0), yielding

(0 ++ (STOPS STARTS)).

3.1.4. ANGULAR-VELOCITY and AXIS

The other possible form of movement considered here is a change of orientation, signalled by ANGULAR-VELOCITY and AXIS. As the names imply, ANGULAR-VELOCITY is the rate of angular orientation change per unit time about the actual or virtual AXIS.

Since AXIS is a direction in space it must be computed relative to a REFERENCE, but this is a minor step once the direction with respect to the observer is found. It is probably sufficient to allow AXIS a very coarsely quantized space of directional values: for example simple combinations of spatially adjacent relations UP, DOWN, NORTH, SOUTH, EAST and WEST. We will often be unable to compute the AXIS as an actual numerical direction, but locating its octant in space is not too difficult (see section V.1.3.2).

The ANGULAR-VELOCITY is the approximate magnitude of the orientation change of the SUBJECT per unit time. This property operates analogously to VELOCITY; in particular, it also must have a nonnegative value. Angular rotations do not have exactly the same distance difficulties VELOCITY has, but projections of rotations may distort the displacements involved (see section
V.1.4. The two senses of rotation, CLOCKWISE and COUNTERCLOCKWISE, are determined by the direction of the AXIS (it is a vector after all) which in turn is dependent on the change of ORIENTATION of the SUBJECT.

The structure of ANGULAR VELOCITY is precisely the same as that of VELOCITY. Both VELOCITY and ANGULAR VELOCITY are necessary properties for an event since an object may be exhibiting both kinds of motion simultaneously. For example, a bowling ball may be spinning slowly while traveling fast.

3.2. Higher level: DIRECTION

The property DIRECTION was mentioned in passing above. It provides a context for the location or orientation change of the SUBJECT by specifying adverbs or prepositions with their objects. DIRECTION concepts include:

1. relationships between the ORIENTATION of the SUBJECT and its TRAJECTORY — BACKWARD, FORWARD, SIDeways; also relationships between the rotation of the SUBJECT and the AXIS — AROUND(2), OVER(3) and also CLOCKWISE and COUNTERCLOCKWISE;

2. relationships between the TRAJECTORY of the SUBJECT and directions in the world — DOWN(WARD), UP(WARD), and also NORTHWARD, SOUTHWARD, EASTWARD, WESTWARD, following the discussion under REFERENCE;

3. relationships changing between the SUBJECT and other objects — ACROSS, AGAINST, ALONG, APART, AROUND(1), AWAY, AWAY-FROM, BEHIND, BY, FROM, IN(1), INTO, OFF, OFF-OF, ON, ONTO, OUT(1), OUT-OF, OVER(1 and 2), THROUGH, TO, TOGETHER(2), UNDER;

4. relationships indicative of source and target — AWAY-FROM, IN-THE-DIRECTION-OF, IN(2), INWARD, OUT(2), OUTWARD, TOWARD;

5. relationships between the path of the SUBJECT and other moving objects in a similar path — AFTER, AHEAD-OF, ALONG, APART, TOGETHER(1), WITH;

6. relationships between this event and a previous event — BACK-AND-FORTH, TO-AND-FROM, UP-AND-DOWN, BACK, THROUGH.

A complete list of these terms, which we call directional adverbials, appears in section III.3.2.2 along with their definitions. Discussion of the type (5) and (6) adverbials also appears in section IV.2.2. The numbers in parenthesis indicate different senses of the same word.

Categories (1) and (2) are essentially derivable from information already stored in the event node. Thus (1) can be
computed from the current TRAJECTORY or AXIS on the event node and the current ORIENTATION of the SUBJECT. Since later events may change the ORIENTATION with respect to the path and since the ORIENTATION is a property of the SUBJECT, not of the event, it seems desirable to encode the relationship in DIRECTION. Later on, knowing the TRAJECTORY and the DIRECTION one can reconstruct the ORIENTATION, at least to a fair approximation.

As for (2), we note that it is no more than a conceptual description of the directions in the absolute coordinatization of the world discussed under the property REFERENCE. It is less precise, of course.

The third category describes the changes in the static description of the scene as the SUBJECT moves. These adverbials describe changes in the state of the world by incorporating the change into an event node. Some even describe changes between events: THROUGH as an INTO followed by an OUT-OF. The frequency of occurrence of these words in natural language descriptions attests to their descriptive usefulness.

The fourth group is generally used for current events only, to describe movement towards or away from a particular object or location. This could be used to determine possible goals or even impending collisions. Once the event is terminated (superceded by another event for that SUBJECT), this directional information will generally be absorbed into a description by a type (3) adverbial: thus AWAY-FROM by FROM, TOWARD or IN-THE-DIRECTION-OF by TO, and so on.

Categories (5) and (6) are higher level concepts, relating the events of separate SUBJECTs (5) or previous events of the same SUBJECT (6). Further discussion of these will be postponed until section IV.2.2.

The DIRECTION property is a list in reverse temporal order, latest recognized adverbial first. For example,

He left the store, passed by the post office on his way, and pulled into his garage when he got home.

would have a DIRECTION list

((INTO garage) (TO home) (BY post-office) (FROM store)).
To more faithfully represent simultaneous adverbials, a DIRECTION element may itself be a list of adverbials and their objects. For example,

*He jumped over three rocks.*

would have a DIRECTION list

\[( (OVIF rock1) (OVPP rock2) (OVER rock3) )\].

The DIRECTION list provides access to lower level information on where particular relationships were made or broken during an event. It may be used to establish locative rather than locational properties for an event in the following sense. Consider:

*John hit Mary on the head while they were in the park.*

The locative case of the (linguistic) event is the general vicinity ("in the park") while the locative case is the point of activity or contact ("on the head"). The distinction in our representation is indirect: location values must be obtained from the LOCATION properties of the various objects involved; locative values are available through the directional adverbials and their indication of CONTACT to specific parts of an object.

The difficulty in precisely defining the boundary between location and locative makes it even more important to provide access to all location information surrounding the event without "pigeon-holing" it prematurely. Thus we would claim that no special location property is required for events. From other considerations, Miller (1972) concurs; there is no location-like case in his analysis of a case structure for motion verbs. DIRECTION indicates locations sufficiently.
3.2.1. Adverbial representation

The directional adverbials are represented by demons. The postcondition essentially instantiates the adverbial name on the DIRECTION list, although it can include an arbitrary graph transformation. For example, AROUND(3), BACK and THROUGH add higher level event nodes. Section III.3.2.2 gives the pre- and postconditions for most of the directional adverbials. We will use boxes to indicate the patterns and resulting information placed in the database. Additional criteria will be specified in English to avoid excessive notation. Some of the definitions are best expressed this way, either because we will examine the definition later or because the adverbial demon depends on low level information supplied by the algorithms in section V.1.

By the definition of directional adverbials, the SUBJECT of the event must move in order for its relations to match an active demon. This implies that the second precondition always contains an implicit test for motion, such as a CURRENT-VELOCITY or CURRENT-ANGULAR-VELOCITY which is non-zero, even though none is indicated in the demon descriptions below. In this way we can use the adverbial demons "as is" in the description of diagrams. (See section IV.3.2.)

No such assumption will be made for the first precondition, however. Unfortunately, efficiency suffers because certain object configurations may activate demons for adverbials even though these demons are rarely or never executed. For example, a chair is a MOBILE object and its presence at one side of a room would at least activate the ACROSS and OFF-OF demons. We could postpone the activation of demons until an object is actually observed to move, but although this seems more natural, we will not make it a universal condition. Such a motion condition is utilized where relations are made (for example, AGAINST, BEHIND, IN, ONTO, or OVER) rather than broken (ACROSS, FROM, OFF-OF, or OUT-OF). In an actual system where efficiency is an important consideration, observed motions could trigger retrospective analysis of the previous data base state to determine likely adverbials. Such a state would, of course, need to be saved; this is not presently the case because static
relations between objects are stored before adverbial demons are activated.

The function PCONS adds an element to the front of a list; if several adverbials add elements to a DIRECTION list in parallel, those items are combined into a single list as described in section III.3.2.

3.2.2. Directional adverbials

This section gives definitions for the following set of directional adverbials:

- ACROSS
- AFTER
- AGAINST
- AHEAD-OF
- ALONG
- APART
- AROUND
- AWAY
- AWAY-FROM
- BACK
- BACK-AND-FORTH
- BACKWARD
- BEHIND
- BY
- CLOCKWISE
- COUNTERCLOCKWISE
- DOWN
- DOWNWARD
- FORWARD
- FROM
- IN-THE-DIRECTION-OF
- INTO
- INWARD
- OFF
- OFF-OF
- ON
- ONTO
- ONWARD
- OUT
- OUT-OF
- OUTWARD
- OVER
- SIDEWAYS
- THROUGH
- TO
- TO-AND-FROM
- TOGETHER
- TOWARD
- UNDER
- UP
- UP-AND-DOWN
- UPWARD
- WITH

An adverbial is included in this list if it has a meaning which involves the movement of one or more of the related objects. It is recognized that many of these terms have additional meanings in static situations, but our interest here is only in the dynamic cases.

The SUBJECT will always be denoted by the variable X. Other objects will be denoted by variables Y, Z, W, and so on, as required. There are no restrictions on the movements of these other objects since the adverbials are dependent on changing relations between object X and its context of other objects.

Phrases preceded by (RH) are quoted or paraphrased from the Random House Dictionary of the English Language (1966). References to LOCATION relations will indicate the most general
term applicable, for example, CONTACT means any one of CONTACT, SUPPORTED-BY, GUIDED-BY or CONNECTED-TO.

ACROSS

(PH) from one side to the other of. A NEAR-TO relation with one side of an object is broken and replaced by a similar relation with the other side. There is an implicit sense of passage ABOVE the object.

precondition 1, ACROSS

\[
\begin{align*}
&\text{NEAR-TO}(X, S_1) \\
&\text{SUB-PART}(Y, S_1) \text{ for some object } Y \text{ and SUB-PART [chain] to object } S_1. \\
&\text{FRONT or BACK or LEFT-SIDE or RIGHT-SIDE}(Y, S_1). \\
&\text{(ACROSS remains active as long as NEAR-TO}(X, Y) \text{ and ABOVE}(X, Y) \text{ hold)}
\end{align*}
\]

precondition 2, ACROSS

\[
\begin{align*}
&\text{NEAR-TO}(X, S_2) \\
&\text{SUB-PART}(Y, S_2) \text{ for a SUB-PART [chain] to object } S_2. \\
&\text{FRONT or BACK or LEFT-SIDE or RIGHT-SIDE}(Y, S_2) \text{ where } S_1 \neq S_2 \text{ and at least one of the ORIENTATION relations to } S_1 \text{ (from precondition 1) no longer holds.}
\end{align*}
\]

\[\Rightarrow \text{ postcondition} \]

\[
\begin{align*}
&\text{SUBJECT } X \\
&\text{DIRECTION } \text{PCONS}((\text{ACROSS } Y), \text{ DIRECTION})
\end{align*}
\]

This definition is more restrictive than the dictionary definition because the latter does not constrain the movement to a reasonable neighborhood of the object gone ACROSS. The somewhat awkward precondition 1 forces the proximity.

AFTER

(PH) behind in position. The SUBJECT moves in the same trajectory at the same physical point though at a later time than the object followed. AFTER is therefore a higher level concept than the basic directional relations. See section IV.2.3.

AGAINST

The onset of a CONTACT relation where none had existed immediately before. There are two independent cases, so they are written as separate demons for clarity.
precondition 1, AGAINST

\[\text{NOT-CONTACT}(X, Y) \text{ for some object } Y\]
\[\text{CURRENT-VELOCITY} \neq 0 \text{ or CURRENT-ANGULAR-VELOCITY} \neq 0\]
\[\text{for some object } X\]

precondition 2, AGAINST

\[\text{CONTACT}(X, Y) \text{ and } Y \text{ is a non-actor}\]

=> postcondition

\[\text{SUBJECT } X\]
\[\text{DIRECTION PCONS}((\text{AGAINST } Y), \text{ DIRECTION})\]

Or, in the case where actual contact is not observed:

precondition 1, AGAINST

\[\text{A significant TRAJECTORY or ANGULAR-VELOCITY/AXIS}\]
\[\text{change of a NOT-MOTILE SUBJEC}T X.}\]

precondition 2, AGAINST

\[\text{There is an object } Y \text{ such that NOT-SUPPORTED-BY}(X, Y),\]
\[\text{AT}(X, Y) \text{ and } Y \text{ is the closest such object to } X, \text{ and}\]
\[Y \text{ is a non-actor.}\]

=> postcondition

\[\text{SUBJECT } X\]
\[\text{DIRECTION PCONS}(((\text{OFF-OF } Y) (\text{AGAINST } Y)), \text{ DIRECTION})\]

Actors and non-actors are defined in section III.3.3.1; significant trajectory and rotation changes in section IV.1.2. The trajectory or rotation change implies motion of the SUBJECT in precondition 1. Precondition 2 applies whether the object bounces off the ground (it would still be unsupported after the trajectory change) or collides with an object while supported by the ground. In either case there is an implied CONTACT change, although no new CONTACT relation is asserted. In section IV.1.4.2 we will see how this adverbial works. If the (inferred) object of contact is an actor (rather than a non-actor), see TOGETHER (2).

AHEAD-OF

(RH) in front of; before. This is the same as AFTER except that the SUBJECT precedes the other object temporally.
ALONG

(RH) moving parallel to the length or direction of. This involves a test for a trajectory roughly parallel to some feature or boundary of an object. Again this is a higher level concept because the movement must be related to characteristics of another object. See section IV.2.3.

APART

(RH) into pieces or parts. See section IV.2.3. The opposite of TOGETHER.

AROUND

(1) (RH) reached by making a turn or partial circuit about. This is detected by a change in SUPPORT-PLANE relations with respect to an object.

precondition 1, AROUND(1)

\[ \text{NEAR-TO}(X \ Y) \text{ for some object } Y \]

precondition 2, AROUND(1)

\[ \text{NEAR-TO}(X \ Y) \]

Assertion of any ordered sequence of four SUPPORT-PLANE(X Y) relations (such as LEFT-OF, then IN-FRONT-OF, then RIGHT-OF, then IN-BACK-OF), where no more than two hold concurrently

\[ \Rightarrow \text{postcondition} \]

\[ \text{SUBJECT X DIRECTION CONS((AROUND Y), DIRECTION)} \]

The AROUND(1) demon must keep a list of found relations so precondition 2 may be tested over a period of time. When executed, the demon erases its memory of the list, so that a single new relation (such as LEFT-OF) does not invoke the demon again. It must try to satisfy precondition 2 (at least) all over again.

(2) (RH) so as to revolve or rotate about a center or axis. Information on rotation may be obtained from the AXIS and ANGULAR VELOCITY properties of an event node. This sense will always be implicit if these properties are defined for an event.
(RH) from a place; to another place. This preposition does not have any precise directional component (see AWAY-FROM). We will use a definition of disappearance through increasing distance or occlusion by the observer's field of view (caused by SUBJECT movement rather than observer movement). Occlusion by another object is described by BEHIND.

precondition 1, AWAY

\[
\text{VISIBLE}(X) \\
\text{CURRENT-VELOCITY} \neq 0 \text{ for object } X
\]

precondition 2, AWAY

\[
\text{NOT-VISIBLE}(X) \\
\text{NOT-OCLUDED-BY}(X \ Y) \text{ for any object } Y
\]

=> postcondition

\[
\text{SUBJECT } X \\
\text{DIRECTION } \text{PCONS}((\text{AWAY } \ Y), \text{DIRECTION})
\]

**AWAY-FROM**

AWAY with the former location specified. This would be used at the onset of the movement away. The former proximity is indicated by a NEAR-TO relation; the departure by its change into FAR-FROM. In addition, AWAY-FROM is the opposite of IN-THE-DIRECTION-OF.

precondition 1, AWAY-FROM

\[
\text{NEAR-TO}(X \ Y) \text{ for some object } Y
\]

precondition 2, AWAY-FROM

\[
\text{FAR-FROM}(X \ Y)
\]

=> postcondition

\[
\text{SUBJECT } X \\
\text{DIRECTION } \text{PCONS}((\text{AWAY-FROM } \ Y), \text{DIRECTION})
\]
**BACK**

(RH) at or toward the original starting point or place. The SUBJECT can come BACK only after having gone AWAY. See section IV.2.2.

**BACK-AND-FORTH**

(RH) from side to side. BACK-AND-FORTH describes movement along BACK and FRONT orientation directions of the particular SUBJECT involved; a repetition of BACKWARD and FORWARD. If the object has no BACK and FRONT orientations, BACK-AND-FORTH describes movement in (visual) depth. See section IV.2.2.

**BACKWARD**

(RH) toward the back or rear. The SUBJECT must have an orientation indication for BACK. The TRAJECTORY of the SUBJECT and its BACK orientation direction differ by less than π/2. See FORWARD for demon.

**BEHIND**

The onset of an OCCLUDED-BY relation between the SUBJECT and another object. The SUBJECT must be moving prior to the occlusion.

precondition 1, BEHIND

\[
\text{VISIBLE}(x) \quad \text{CURRENT-VELOCITY}_x \neq 0 \text{ for object } x
\]

precondition 2, BEHIND

\[
\text{OCCLUDED-BY}(x, y) \text{ for some object } y
\]

\[ \implies \text{postcondition} \]

\[
\text{SUBJECT}_x \quad \text{DICTION-POCS}((\text{BEHIND}_y), \text{DIRECTION})
\]

**BY**

(RH) to and beyond a point near something. There is a connotation of changing the SUPPORT-PLANE relations rather than the ELEVATION ones. The SUBJECT moves BY an object (NPP-TO it at some point) intersecting the plane \( \Theta (\text{TRAJECTORY}) \pm \pi/2 \).
CLOCKWISE

Relative to some point on the AXIS line, the angle between the direction vector of the AXIS and the direction vector from the observer to that chosen AXIS point is less than \(\pi/2\) (minus some small threshold). The threshold prevents judgments when CLOCKWISE is irrelevant. Since AXIS is a property of an event, the direction vector depends on temporal data. This is not to be confused with AXES of an object which are static and inherently bi-directional.

COUNTERCLOCKWISE

As for CLOCKWISE except the angle described is between \(\pi/2\) (plus a threshold) and \(\pi\) (minus a threshold).

DOWN

(RH) from higher to lower; in descending direction; toward or into a lower position. If \(\phi(T\text{RAJECTORY}) > \pi/2\) then the movement is DOWN.

DOWNWARD

(RH) from a higher to a lower place. This is not much different than the first sense of DOWN. We might take into account a particular DOWN orientation of the object, but this cannot conflict with the vertical sense of direction which always takes precedence. (A chair turned sideways and dropped is not moving SIDeways but rather DOWNWARD.) A value for \(\phi(T\text{RAJECTORY})\) very near \(\pi\) indicates DOWNWARD. See the comment under TOWARD.
FORWARD

(RH) movement in a forward direction. This requires that the object have an intrinsic FRONT orientation. The trajectory of the object and its FRONT orientation direction differ by less than π/2.

Certain adverbials are placed specially in the DIRECTION list, to accommodate possible vision system failure to determine object orientation uniquely at the start of movement. The adverbials FORWARD, BACKWARD, LEFT-SIDWAYS and RIGHT-SIDWAYS are appended at the end of the DIRECTION list. The reasoning behind this requires some understanding of the event generation process: an event node can have only one such adverbial.

FROM

(RH) used to specify a starting point in spatial movement. This will be a place which the SUBJECT was AT, but is now further away (NOT-AT).

precondition 1, FROM

[AT(X Y) for some object Y]

precondition 2, FROM

[NOT-AT(X Y)]

=> postcondition

[SUBJECT X DIRECTION MOVES((FROM Y), DIRECTION)]

IN

(RH) used to indicate motion from outside to a point within. This may be construed as motion towards the observer or towards his point of reference. For example:

He walked in.

We can assume that the observer is occupying the same place, as opposed to INTO, where the observer may have no locational relation to the thing gone into. This definition requires that the object have an INSIDE, and will usually be signalled by the making of a SURROUNDED-BY
relation to the object. There also seems to be a connotation of entrance through a proper aperture for passage.

precondition 1, IN(1)

\[
\text{NOT-SURROUNDED-BY}(X Y) \text{ for some object } Y \text{ having an ORIENTATION INSIDE CURRENT VELOCITY} \neq 0 \text{ for object } X
\]

precondition 2, IN(1)

\[
\text{SURROUNDED-BY}(X Y) \\
\text{SURROUNDED-BY(OBSERVER } Y)
\]

=> postcondition

\[
\text{SUBJECT } X \\
\text{DIRECTION PCONS((IN } Y), \text{ DIRECTION})
\]

(2) used to indicate motion towards a central point. This was obtained by analogy to the second sense of OUT. However, determining what is an appropriate "central point" may be difficult pictorially and might be left for higher level processing.

**IN-THE-DIRECTION-OF**

An object in the path of the SUBJECT's motion to which an otherwise numerical TRAJECTORY may be related. The object should have a point making an angle of no more than about \(\pi/4\) with the TRAJECTORY direction.

**INTO**

(RH) to the inside of; (RH) used to indicate entry, inclusion, or introduction in a place. The object involved requires an INSIDE.

precondition 1, INTO(1)

\[
\text{NOT-SURROUNDED-BY}(X Y) \text{ for some object } Y \text{ having an ORIENTATION INSIDE CURRENT VELOCITY} \neq 0 \text{ for object } X
\]

precondition 2, INTO(1)

\[
\text{SURROUNDED-BY}(X Y)
\]
**=> postcondition**

<table>
<thead>
<tr>
<th>SUBJECT X</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRECTION PCONS((INTO Y), DIRECTION)</td>
</tr>
</tbody>
</table>

**INWARD**

(RH) toward the inside, interior, or center. Once the SUBJECT is INSIDE the object, further motion from the point of entrance or towards the center is described by INWARD.

**OFF**

(RH) so as to be no longer supported or attached; (RH) so as to be no longer covering or enclosing; (RH) away from a path, course, etc.; (RH) so as to no longer be supported by, attached to, on, resting on, or unified with. This definition is quite succinct: OFF-OF describes the breaking of a constraint relation between the SUBJECT and another object (possibly one it is part of). The SUBJECT must have been SUPPORTED-BY, GUIDED-BY or CONNECTED-TO the other object, else that object must have been COVERED-BY the SUBJECT. See OFF-OF.

**OFF-OF**

OFF with the constraining object mentioned.

**precondition 1, OFF-OF**

<table>
<thead>
<tr>
<th>SUPPORTED-BY(X U)</th>
</tr>
</thead>
<tbody>
<tr>
<td>or GUIDED-BY(X V)</td>
</tr>
<tr>
<td>or CONNECTED-TO(X W)</td>
</tr>
<tr>
<td>or COVERED-BY(Y X)</td>
</tr>
<tr>
<td>for some object U or V or W or Y.</td>
</tr>
</tbody>
</table>

**precondition 2, OFF-OF**

<table>
<thead>
<tr>
<th>NOT-SUPPORTED-BY(X U)</th>
</tr>
</thead>
<tbody>
<tr>
<td>or NOT-GUIDED-BY(X V)</td>
</tr>
<tr>
<td>or NOT-CONNECTED-TO(X W)</td>
</tr>
<tr>
<td>or NOT-COVERED-BY(Y X)</td>
</tr>
</tbody>
</table>

**=> postcondition**

<table>
<thead>
<tr>
<th>SUBJECT X</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRECTION PCONS((OFF-OF Z), DIRECTION)</td>
</tr>
<tr>
<td>where Z=U or V or W or Y as appropriate</td>
</tr>
</tbody>
</table>
ON

(RH) into or onto a position of being supported or attached; (RH) into or onto a position of covering or wrapping. Just as OFF signalled the breaking of certain constraining relations, ON signals the making of the same constraints.

ONTO

(RH) to a place or position on. ON with the destination emphasized. It will be preceded by an AGAINST relation. The demon is obtained as the opposite of OFF-OF, with an additional CURRENT-VELOCITY=0 test in precondition 1.

ONWARD

(RH) used to indicate movement in continuance of a course. This is more useful in the final linguistic description than in the generated event structure because ONWARD can only indicate that an event node is currently active, with a nonzero CURRENT-VELOCITY.

OUT

(1) (RH) used to indicate movement from the inside to the outside of something. In general, a breaking of a SURROUNDED-BY relation.

(2) (RH) used to indicate movement away from a central point. These two senses are precise antonyms for IN.

OUT-OF

OUT(1) with the former location specified. The demon is obtained as the opposite of INTO.

OUTWARD

(RH) toward the outside; (RH) proceeding or directed toward the outside or exterior, or away from a central point. The opposite of INWARD.

OVER

(1) (RH) so as to rest on or cover. This describes the making of a COVERED-BY relation.
precondition 1, OWPV(1)
\[
\begin{align*}
&\text{NOT-COVERED-BY}(Z, X) \text{ for some object } Z \\
&\text{CURRENT-VELOCITY} \neq 0 \text{ for object } X
\end{align*}
\]

precondition 2, OWPV(1)
\[
\begin{align*}
&\text{COVERED-BY}(Z, X)
\end{align*}
\]

\[\Rightarrow \text{postcondition}\]
\[
\begin{align*}
&\text{SUBJECT } X \\
&\text{DIRECTION } \text{CONS}((\text{OVPV}(1), Z), \text{DIRECTION})
\end{align*}
\]

(2) (RH) from side to side of; to the other side of; across. Seemingly synonymous with ACROSS, but actually closer to BY where the passage is in a vertical direction. An ABOVF relation is required, but not NEAR-TO.

(3) (RH) to bring the upper end or side down or under. This represents a particular type of movement where the orientation of the object is changed in the indicated manner, as in "turn over." It is thus an AROUND(3) motion where the axis lies nearly in the horizontal plane.

**SIDEWAYS**

Movement in a SIDE orientation direction, where the SUBJECT has such. Technically, there are two terms here, LEFT-SIDEWAYS and RIGHT-SIDEWAYS, depending on the direction of motion relative to the LEFT- or RIGHT-SIDE, respectively. The demon operates like that of FORWARD.

**THROUGH**

(RH) in at one end, side, or surface and out at the other. THROUGH is a higher level concept building on sequences of INTO and OUT-OF relations to different (opposite) parts of the same object. See section IV.2.2.

**TO**

(RH) used for expressing motion toward a point, person, place, or thing approached and reached, as opposed to "from." The opposite of FROM: the making of an AT relation.
TO-AND-FRO

Same as BACK-AND-FORTH except the movement is with respect to the e coordinate of the picture or else side-to-side with respect to the FRONT orientation of the object, that is, a repeated RIGHT-SIDeways and LEFT-SIDeways.

TOGETHER

(1) (RH) into one gathering, company, mass, place, or body. Two or more objects may begin to experience like events and hence may be considered as one. The event nodes for each are combined into a single node with a list of SUBJECTs, and the new node's DIRECTION contains TOGETHER. See section IV.2.3.

(2) (RH) into union, proximity, contact, or collision. There is also a sense of TOGETHER implying contact or collision. It resembles the demon for AGAINST except the (inferred) object of contact is an actor (see section III.3.3.1). The TOGETHER adverbial appears in each object's DIRECTION list. If the contact was inferred, APART is asserted just as OFF-OF was for AGAINST. Thus a ball and bat come TOGETHER (then move APART), but a ball rolls AGAINST (then bounces OFF-OF) a wall.

precondition 1, TOGETHER(2)

\[
\text{NOT-CONTACT}(X, Y) \text{ for some object } Y \\
\text{CURRENT-VELOCITY} \neq 0 \text{ or CURRENT-ANGULAR-VELOCITY} \neq 0
\]

for two objects X and Y

precondition 2, TOGETHER(2)

\[
\text{CONTACT}(X, Y) \text{ and } Y \text{ is an actor}
\]

=> postcondition

\[
\text{SUBJECT } X \\
\text{DIRECTION PCONS}((\text{TOGETHER}(2), Y), \text{DIRECTION}) \\
\text{SUBJECT } Y \\
\text{DIRECTION PCONS}((\text{TOGETHER}(2), X), \text{DIRECTION})
\]

Or, in the case where actual contact is not observed:

precondition 1, TOGETHER(2)

\[
\text{A significant TRAJECTORY or ANGULAR-VELOCITY/AXIS change for some object } X
\]
precondition 2, TOGETHER(2)
There is an object Y such that NOT-SUPPORTED-by(x Y),
AT(x Y) and Y is the closest such object to x, and
Y is a actor

=> postcondition
SUBJECT X
DIRECTION PCONS(((APART Y) (TOGETHER(2) Y)), DIRECTION)
SUBJECT Y
DIRECTION PCONS(((APART Y) (TOGETHER(2) Y)), DIRECTION)

TOWARD
(RH) in the direction of. The actual direction angle is
used rather than a cone of angles, since TOWARD seems more
specific than IN-THE-DIRECTION-OF. The latter may be off-
course by a considerable angle.

He threw the ball skyward [TOWARD the sky] in John's
direction [IN-THE-DIRECTION-OF John], although it
landed far to his side.

UNDER
(RH) below or beneath something. UNDER occurs with the
making of a BELOW relation. It parallels OVER in other
respects.

UP
(RH) to or toward a more elevated position. (RH) to any
point that is considered higher. This is the opposite of
DOWN, indicated when $\phi$(TRAJECTORY) < \pi/2.

UP-AND-DOWN
Same as BACK-AND-FORTH except the movement is with respect
to the $\phi$ coordinate of the picture, thus UP(WARD) and
DOWN(WARD). By the comment under DOWNWARD, this is
independent of object orientation.

UPWARD
(RH) toward a higher place or position. The opposite of
DOWNWARD. Again something moves UPWARD without any regard
for its intrinsic orientation. UPWARD requires
$\phi$(TRAJECTORY) near zero. See the comment under TOWARD.
WITH

(PP) accompanied by, accompanying. Objects once brought
TOGETHER(1) move WITH one another until they move APART.
See section IV.2.3.

3.3. SUBJECT, AGENT, and INSTRUMENT

According to Miller, a great many motion verbs are
characterized by directional adverbials and a knowledge of which
objects are acting or are being acted upon. This last
distinction involves discriminating the "actors" in a particular
movement sequence.

3.3.1. Actors and non-actors

Several actors participate in an event perceived as a
movement in one of the actors, the SUBJECT. This object or set
of objects is the focus of the event; a description of its
movement may or may not turn out to be important, since many
moving objects may be involved. Thus in

John hit the ball with a stick.

there are at least two moving objects (John, stick, and probably
the ball), as well as John's body motions. Assume that every
MOBILE object is the SUBJECT of an event node. Rigidly attached
subparts of a moving object do not require their own event nodes
since they inherit the underlying motion from the dominating
object.

There are two other actors in the event, the AGENT and the
INSTRUMENT. Both must have had some CONTACT relationship with
the SUBJECT, perhaps indirectly through chains of CONTACT-type
relations with MOBILE objects, and themselves must have been
moving prior to the time of contact. Objects which have been
contacted but which are not initially moving will appear as
objects of changes in locative relationships, and will appear in
the DIRECTION list: for example, a ball that bounces AGAINST a
parked car.

Because some moving objects may be involved in an event but
not as AGENTS or INSTRUMENTS (that is, as non-actors), we will

-92-
give a more general definition of the actor/non-actor distinction. Consider a ball being thrown at a passing car, hitting it. There are three possible situations:

1. The ball bounced off the front of the car.
2. The ball bounced off the side (or the roof) of the car.
3. The ball bounced off the rear of the car.

In the first case it is likely that the ball bounced away with a greater speed than it had initially. It is reasonable to claim the car accelerated the ball and acted causally in the rebounding motion of the ball. Consider the second case; the car is not acting as a agent of force for the bounce, since the situation is exactly the same as

The ball bounced off the wall.

Although physics tells us that the wall (or car side) did in fact push back with a force equal to the impact of the ball, the motion of the car in the second case is clearly irrelevant to the resulting action. Now in the third case, where the ball bounces off the rear of the car, we might not have a change in the horizontal trajectory of the ball, only a slowing of its velocity. But again the car is clearly an actor by slowing the motion of the ball. If it were not an actor, the ball would have left at the same velocity as it arrived.

One conclusion is that an actor must be either a source (case 1) or a sink (case 3) of motion of the SUBJECT. The determination of source or sink depends on changes (positive or negative) in the VELOCITY or ANGULAR VELOCITY of the SUBJECT, rather than its TRAJECTORY, thus making it somewhat difficult to observe from the scene. Case 2 provides an alternative definition, namely that the TRAJECTORY of a non-actor is orthogonal to the TRAJECTORY of the SUBJECT. Where there is no movement of an object at all (for example, the wall) there is no corresponding TRAJECTORY vector, hence a default orthogonality. The test is made at the respective points of contact if possible. This method is an improvement over the first because trajectories are temporally-independent and reasonably computable while velocities are neither. Where the SUBJECT is rotating, changes in ANGULAR VELOCITY to or from zero or
reversals in AXIS direction will indicate that contacted objects are actors.

Either or both the AGENT and the INSTRUMENT may be missing; on the other hand either or both may be sets of objects.

Jack drove his car [SUBJECT] along the road.
The ball [SUBJECT] rolled ten feet.
John [AGENT] waved his arms [SUBJECTs].
Sam [AGENT] hit the ball [SUBJECT].
They [AGENTS] pushed the car [SUBJECT].
The car [AGENT] pushed over the pole [SUBJECT].
He [AGENT] hit the ball [SUBJECT] with a bat [INSTRUMENT].
Sam [AGENT] swung the bat [INSTRUMENT/SUBJECT] that propelled the ball [INSTRUMENT/SUBJECT] that bounced off a wall [object of adverbial; non-actor] and knocked over Mrs. Jones prize petunias [SUBJECTs].
The book [SUBJECT] fell over (because of gravity [AGENT]).
The windmill vanes [SUBJECT] turned in the wind [AGENT].
He [SUBJECT/AGENT] walked on his hands [INSTRUMENTs].

The distinction between the AGENT and the INSTRUMENT depends on the presence of a MOTILE value on the MOBILITY property of the object. An AGENT is MOTILE; an INSTRUMENT is NOT-MOTILE. The definition we will use for motility is that an object is motile if and only if it is capable of self-propulsion in the observed manner. Obviously this definition entails knowing the identity of an object and certain of its characteristics. The present state of a possibly motile object decides its actual motility. Thus some motile objects are people, dogs, cars and trucks, provided that they are alive or fit to operate.

The condition that the object must be capable of producing the motion it was observed to have could involve very complex deductions. Higher level programs must assess cause and effect issues. (see, for example, Hayes (1971a)).

In any particular situation we should locate the physical AGENT (a MOTILE or ANIMATE object) if possible, or else attribute movements to an invisible physical force. This last statement points out the one exception to the rule requiring an
AGENT to be a moving object contacting the SUBJECT, namely when that AGENT is an invisible physical force such as wind or gravity. They are certainly valid AGENTS since they are sources of motive force and do not act simply to transmit force from another AGENT as an INSTRUMENT does.

3.3.2. A note on causality

It is worth mentioning here the work of Michotte (1946). He obtained descriptions of events from human subjects by displaying sequences of simple geometric forms in motion. From a discussion in Arnheim (1969), we can see that Michotte's experiments indicate that the object property MOBILITY can often be inferred from the object's own movements. For example, an object seen to move without a visible impetus will be considered MOTILE. If such an object (A) touches another, stationary object (B) such that B begins to move, the motile object A is said to have caused the motion of B. Without other evidence, B would be considered MOBILE rather than MOTILE.

Michotte's investigations of more subtle interplays between speed, path, and direction indicate that causality (and hence AGENTS and INSTRUMENTs) can often be inferred for simple objects. Our definitions for AGENT and INSTRUMENT will not always yield perfect causal relationships as they utilize only a limited set of physical data. For the most part, our definitions will give reasonable results if the scenarios themselves are accurate representations of probable realities. We do not expect, for example, that the contact between the thrown ball and the moving car will cause the car to veer away as if the ball pushed it. Rather we would tend to attribute the causality to the car (or its implied driver) by saying in effect: 'the car caused itself to veer away because of the contact with the ball.'
3.4. **Time: START-TIME, END-TIME, REPEAT-PATH and NEXT**

Another dimension to an event is temporal: when does it begin and end, how long does it last, and what does it follow? The corresponding properties of the event node are START-TIME, END-TIME and NEXT. An additional property, REPEAT-PATH, enables descriptions of higher level and repetitious events.

3.4.1. **Representing time**

Time may be represented in several ways, depending on the form of the temporal information. Where there is only one observer and therefore only one source of input information, all events can be related to precisely-known clock times. The intervals between "pulses" need not be equal, as for example in Winograd's (1971) blocks world where the clock emits an integer "pulse" with every event. Since all information comes from a single source every event is comparable to every other and the time graph is simply linear. Whether one uses real numbers or a more associative data structure to represent time is not important here. We will call this type of representation "hard."

Where the observer has no such precise internal clock or where information on events comes from more than one source (vision and natural language conversation, for example) the representation must be "softened" into a partially-ordered set of events. Thus two events may be known but the description does not distinguish their actual temporal relationship (it is not known). A representation of this sort has been proposed and used by Bruce (1972). One problem with his arrangement is that the time relations become more important than the event relationships, whereas we would like the incoming stream of events to define the time graph implicitly. Time is a characteristic of an event, not vice versa. From either point of view, however, it is possible to obtain descriptions of temporally-related events in terms of linguistic concepts such as BEFORE, AFTER, DURING, WHILE, and UNTIL.
In this application a hard representation will be used since the presentation time of each frame defines an explicit point in time. The onset of a particular event is given by the property START-TIME and the final time for which this event node adequately characterizes the observed movements is given by the property END-TIME. The next event for the SUBJECT is obtained from the relation NEXT; there is an implicit property PREVIOUS encoding the inverse relation. NEXT partially orders the set of event nodes into an event graph. In general it is a graph, not a tree, since an event node may contain a list of SUBJECTs. These may combine or articulate their SUBJECTs' respective movements creating multiple NEXT-edge sinks or sources.

The property END-TIME is the only one of this set to have an interpretation for a queue of values. NEW-END-TIME is the time of the present frame and is filled in when data is collected for the SUBJECT of the event. CURRENT-END-TIME is the time of the last observation updating information on this node. LAST-END-TIME is the previous CURRENT-END-TIME. The intent is to provide enough short-term memory to compute derivatives for TRAJECTORY, VELOCITY, and if necessary, AXIS and ANGULAR VELOCITY. LAST-END-TIME will only be defined for currently active event nodes.

3.4.2. Repetition and event hierarchies

If events occur in a sequence that is repeated many times it is unwieldy to keep every event node around for each repetition. There should be a representation for repetitious events and a demon to determine when repetitions occur so that the event graph can be kept manageable. We would also like to model the natural human ability to perceive repetitious patterns. The problem of determining repetitions in the input will be discussed later in sections IV.2.2 and IV.2.3.

The representation of repetitious events is based on the event node property REPAT-PATH. This is an ordered list of NEXT-consecutive event nodes which are taken to form a repeating sequence. The START-TIME and END-TIME of an event in a repeated sequence correspond to the last occurrence of the sub-event in

-97-
the input. The END-TIME of the REPEAT-PATH event is the final time the repetition is effective; the START-TIME is the onset of the repetition (not the onset of its last cycle). If the repetition is currently active, there will be no NEXT edge emanating from the event node. In short, an event with a REPEAT-PATH list looks like any other single event. If REPEAT-PATH event nodes are nested, the START-TIME and END-TIME are those of the last observed repetition.

This representation admits repetitions of a single cycle. If FIRST and LAST retrieve the first and last elements of a list, and if

\[
\text{START-TIME}((\text{FIRST}((\text{REPEAT-PATH}(E)))) = \text{START-TIME}(E)
\]

\[
\text{END-TIME}((\text{LAST}((\text{REPEAT-PATH}(E)))) = \text{END-TIME}(E),
\]

then the REPEAT-PATH is repeated only once. Since REPEAT-PATH normally indicates events at a lower level there is a nice uniformity in interpreting a single repetition as a decomposition of the REPEAT-PATH event node into a sequence of lower level events. This property therefore gives rise to hierarchies of events, repetitious or not.

It might seem that this simple representation does not allow for optional events or multiple-alternative events at some point in the sequence. Independently of whether or not we want to consider such sequences as repetitions, these situations need to be described. One way would be to require an indicator on each element of the REPEAT-PATH list saying whether it was a necessary or optional node, or which of several alternatives might be selected at each iteration. A better approach is not to embellish a secondary structure (REPEAT-PATH), but to use the event structure itself. For example, suppose we watch a man run in a circle and each time he comes near us he makes a different face or gesture. The description in terms of events in this representation is a single event giving the REPEAT-PATH as two more events: the first of the man "running in a circle," and the second of him "making a face or gesture when he comes near us." Besides lying very close to the initial description of the scenario, this arrangement utilizes events as higher level patterns for the description of generic movements. These
generic patterns in turn are used in the analysis of movement data. An algorithm for finding repetitions, in particular, must perform exactly this kind of pattern matching; if it could not then the descriptive power of the event representation would be questionable.

3.5. Miller's cases

Miller (1972) uses several other cases in his analysis of motion verbs which may be considered as special combinations of values already allowed by our case system. We can characterize all of his cases except those involving knowledge of an object's abilities — the permissive and the propellant.

Motion
Our methodology only considered the description of motion. However, the canonical concepts of CHANGES-LOCATION and CHANGES-ORIENTATION are differentiated within lower level cases. They are not distinguished in Miller's concept, as we have already pointed out.

Reflexive-objective
The description always refers to motion of the SUBJECT (Miller's patient), hence is reflexive. The objective part would be obtained from linguistic transformations of the data about the actions of a particular AGENT on other objects.

Causative
The AGENT is the cause of the event.

Directional
The DIRECTION is the list of adverbials which describe the contextual direction of motion. We include more terms here than Miller seems to indicate can occupy his case.
Medium
The medium case in our world is easily retrieved by seeing whether or not the SUBJECT was SUPPORTED BY an object or objects throughout the event. (The directional adverbials ONTO and OFF-OF would indicate any change in support.) If supported, the medium is land or ground, otherwise air. Water is not considered an object in a world restricted by the initial assumptions to contain only rigid or jointed objects.

Instrumental
The INSTRUMENT is the intermediate object to transfer force from the AGENT. These cases should agree.

Inchoative
An inchoative case would be paraphrased as: "the SUBJECT became adjective," where adjective might be OPEN or CLOSED, for example. Interestingly, Miller actually uses this case in his analysis of motion verbs for only these concepts (OPENS, CLOSES and SHUTS). We will not take this approach because OPENS and CLOSES are usually associated with distinct subparts of an object (doors, windows, covers) and the direction of movement of the sub-object with respect to the containing object (wall, car, book), or the attachment/un-attachment of the sub-object to its parent object can indicate OPENS or CLOSES. Thus a separate inchoative case on events is unnecessary since information on OPENS and CLOSES may be derived from the knowledge of an object's identity, its relation to containing or attached objects, relative movement or dissociation of the two, and whether it can have OPENS/CLOSES as a characteristic.

Change-of-motion
This case is incorporated into the VELOCITY and ANGULAR-VELOCITY cases, particularly in the RATES lists.
Deictic

The deictic case (movement towards or away from the observer) is subsumed within the choice of adverbials (from type (2) mostly) since given any observer's location in the world, the relationship between the SUBJECT's motion and the observer's location may be derived.

Velocity

Rates of movement may be obtained from numeric values in the VELOCITY and ANGULAR VELOCITY cases. Notice that Miller's case would have to depend on object capabilities: whether a particular object were moving fast or slow.

The correspondence between the two case systems reinforces our belief that reasonable conceptual descriptions of dynamic scenarios may be generated by our methodology.
Chapter IV. Description Algorithms

Having determined a representation for objects and events we utilize it in the generation of conceptual descriptions for the scenarios discussed in Chapter I. The generation process is developed in this chapter, beginning with a demon which segments changes into first level event nodes, or primitive events. Various demons are discussed which condense the primitive events into more linguistic forms and eliminate certain redundancies in the description. Higher level events are derived from an analysis of certain directional adverbials and the concept of repetition. Finally, demons defined over event nodes are used to show how individual motion verbs might be recognized and how the manipulation of events applies to problem solving.

IV.1. The demon for primitive events

The initial goal is the presentation of a demon for building the first level event nodes. This demon is in charge of determining when instantiated data creates or terminates an event node. Later on, methods for eliminating superfluous information will be examined.

1.1. Some utility definitions

A null event node has all its movement properties NIL or zero except START-TIME and END-TIME. Null event nodes are useful for filling gaps in the description of an object's movements due to occlusion or disappearance through distance. They also arise naturally from movements that are not apparent from frame-to-frame, because feature displacements were too small to use in computing the motion.

An event node is said to be terminated if it has a non-NIL NEXT edge.

The function CREATE-EVENT-NODE (property pairs) creates an event node with the indicated property values, returning the node as a result. For example,

CREATE-EVENT-NODE((SUBJECT CAR2) (START-TIME 10.))
creates an event node for the particular object CAR2 as its SUBJFCT.

The time will be abbreviated by three variables TN, TC and TL. For a particular event node E:

\[
\begin{align*}
TN & := \text{NEW-END-TIME}(E); \\
TC & := \text{CURRENT-END-TIME}(E); \\
TL & := \text{LAST-END-TIME}(E).
\end{align*}
\]

TN is always equal to the current frame time. There is no restriction on TC representing the previous frame time, so there is no inherent restriction on TL, TC and TN representing consecutive image frames.

A utility function SHIFT manipulates property queues when they are updated. SHIFT(property) is defined by

\[
\begin{align*}
\text{LAST-property} & := \text{CURRENT-property}; \\
\text{CURRENT-property} & := \text{NEW-property}; \\
\text{NEW-property} & := * .
\end{align*}
\]

where '*' denotes the unfilled cell ready to accept new data when CURRENT-END-TIME(E) differs from the current frame time.

1.2. The creation and termination demon

There is one demon which controls the construction of the entire event graph. We will give its complicated duties in Algorithm 1 below which describes all the situations handled. Precondition 1 of this demon is the completion of all lower level demons for each frame. The various parts of Algorithm I describe its second precondition. When executed it creates, terminates, or simply updates the current event graph. In section IV.1.3 we will demonstrate that the events determined by this demon are psychologically "reasonable."

ALGORITHM I.
A.1.1. Creating event nodes.
A.1.1.1. An event node E is created when a MOBILE object first becomes visible and identifiable as an object.

\[
E := \text{CREATE EVENT-NODE}((\text{SUBJECT object-node}) \\
(\text{VELOCITY } (* 0. 0.)) \\
(\text{ANGULAR VELOCITY } (* 0. 0.)) \\
(\text{START-TIME NIL}) \\
(\text{END-TIME } (* \text{TN} \text{TN})))
\]

Any change in location or orientation of the SUBJECT will cause the termination of this node by A.1.2.7 or A.1.2.9. The
property queues for VELOCITY, ANGULAR VELOCITY and END-TIME are created to insure that all property fields are defined at termination. This gives a null starting event node for each object independent of its movement. The NIL START-TIME has the interpretation that we do not know what was happening prior to time TN.

Event nodes are created for the parent object only, if it is known to be jointed. This prevents some unnecessary event nodes from being created.

A.I.1.2. An event node E is created when a jointed part of the parent object with current event node EP is first observed to move relative to the parent.

\[
\begin{align*}
TC & := \text{CURRENT-END-TIME}(EP); \\
E & := \text{CREATE-EVENT-NODE}\{ \\
& \text{(SUBJECT object-part-node)} \\
& \text{(AGENT parent-object-node)} \\
& \text{(REFERENCE ... )} \\
& \text{(DIRECTION ... )} \\
& \text{(TRAJECTORY ... )} \\
& \text{(VELOCITY ... )} \\
& \text{(AXIS ... )} \\
& \text{(ANGULAR VELOCITY ... )} \\
& \text{(START-TIME TC)} \\
& \text{(END-TIME (TN TC TC))} \}.
\end{align*}
\]

The joint node was defined in section III.2.5.1. Any appropriate movement attributes are placed in their respective NEW-property positions. The node E is then immediately terminated (A.I.1.3), in order that the new node accurately represents the changed properties. For example, the AGENT might change or an ANGULAR VELOCITY might be introduced.

A.I.1.3. An event node E2 is created whenever another event node E1 is terminated (see A.I.2).

\[
\begin{align*}
TC & := \text{CURRENT-END-TIME}(E1); \\
\text{NEXT}(E1) & := \text{CREATE-EVENT-NODE}\{ \\
& \text{(AGENT ... )} \\
& \text{(INSTRUMENT ... )} \\
& \text{(REFERENCE ... )} \\
& \text{(DIRECTION ... )} \\
& \text{(TRAJECTORY SHIFT (TRAJECTORY (E1)))} \\
& \text{(VELOCITY SHIFT (VELOCITY (E1)))} \\
& \text{(AXIS SHIFT (AXIS (E1)))} \\
& \text{(ANGULAR VELOCITY SHIFT (ANGULAR VELOCITY (E1)))} \\
& \text{(START-TIME TC)} \\
& \text{(END-TIME SHIFT (END-TIME (E1)))} \}; \\
E2 & := \text{NEXT}(E1).
\end{align*}
\]
SUBJECT, AGENT, INSTRUMENT, REFERENCE, and DIRECTION property values are inserted as determined in the present frame, that is, those which were present at termination of the previous node.

A.I.1.4. An event node is created when a repeating event sequence is discovered or when higher level DIRECTION terms generate them (section IV.2.2). Condensation demons may generate new nodes in the process of deleting old ones (section IV.2.1).

A.I.2. Terminating event nodes.

An event node is terminated when there are significant changes in its properties. There is a case for nearly every property. The queue structure for all properties using queues is altered:

\[
\begin{align*}
\text{END-TIME}(E) & := \text{CURRENT-END-TIME}(E); \\
\text{TRAJECTORY}(E) & := \text{CURRENT-TRAJECTORY}(E); \\
\text{AXIS}(E) & := \text{CURRENT-AXIS}(E); \\
\text{VELOCITY}(E) & := (\text{CURRENT-VELOCITY}(E) \\
\text{ANGULAR-VELOCITY}(E) & := (\text{CURRENT-ANGULAR-VELOCITY}(E) \\
\text{RATES} & \text{VELOCITY}(E)) \\
\text{RATES} & \text{ANGULAR-VELOCITY}(E)) \\
\end{align*}
\]

The VELOCITY AND ANGULAR-VELOCITY queues are replaced by the final value and the RATES list of concepts.

Terminated nodes are always subjected to this condensation of their queues and thus a slight generalization of their information. Some changes in TRAJECTORY, for example, will be lost although the final value may be compared to TRAJECTORY(PREVIOUS(E)). The DIRECTION list is unaltered except that the terminating adverbial(s) may be added to DIRECTION(E) rather than DIRECTION(NEXT(E)) (see A.I.2.5).

By A.I.1.3 an event node (NEXT(E)) is always immediately created when E is terminated by any of the terminating conditions below.

A.I.2.1. Changes in SUBJECT.

The assumptions on object rigidity and permanence preclude changes in an object, but see the discussion on the type (5) adverbials in A.I.2.5.

A.I.2.2/3. Changes in AGENT and INSTRUMENT.

These must be preceded by changes in CONTACT relations between objects and the SUBJECT. See A.I.2.5 on DIRECTION.

A.I.2.4. Changes in REFERENCE.
A change in the reference features forces a termination in every event node referencing those features. As such changes are usually caused by spatial or temporal discontinuities in the scene it is unreasonable to expect any of these event nodes to continue in an active descriptive state for the present.

A.I.2.5 Changes in DIRECTION.

Looking at the directional adverbials, we will see that some are event node terminators, others are pre-empted by changes in other properties, and still others require more specialized considerations. Recall the classification used in the representation discussion:

1. relationships between the ORIENTATION of the SUBJECT and its TRAJECTORY -- BACKWARD, FORWARD, SIDWAYS; also relationships between the rotation of the SUBJECT and the AXIS -- AROUND(2), OVER(3) and also CLOCKWISE and COUNTERCLOCKWISE;

2. relationships between the TRAJECTORY of the SUBJECT and directions in the world -- DOWN(WARD), UP(WARD); and also NORTHWARD, SOUTHWARD, EASTWARD, WESTWARD;

3. relationships changing between the SUBJECT and other objects -- ACROSS, AGAINST, ALONG, APART, AROUND(1), AWAY, AWAY-FROM, BEHIND, BY, FROM, IN(1), INTO, OFF, OFF-OF, ON, ONTO, OUT(1), OUT-OF, OVER(1 and 2), THROUGH, TO, TOGETHER(2), UNDER;

4. relationships indicative of source and target -- AWAY-FROM, IN-THE-DIRECTION-OF, IN(2), INWARD, OUT(2), OUTWARD, TOWARD;

5. relationships between the path of the SUBJECT and other moving objects in a similar path -- AFTER, AHEAD-OF, ALONG, APART, TOGETHER(1), WITH;

6. relationships between this event and a previous event -- BACK-AND-FORTH, TO-AND-FROM, UP-AND-DOWN, BACK, THROUGH.

Changes in type (1) adverbials must be preceded by changes in TRAJECTORY, VELOCITY, AXIS, or ANGULAR-VELOCITY, because a relationship between an ORIENTATION and a TRAJECTORY (or AXIS) cannot change without at least one of the above four properties changing. These properties supply directional adverbials of type (1) on the new node.

Changes in CLOCKWISE and COUNTERCLOCKWISE are handled by A.I.2.8 and A.I.2.9 below. Changes in BACKWARD, FORWARD and SIDWAYS may occur with no orientation change if the TRAJECTORY has a non-zero derivative. For example, move a box in a circle while keeping its orientation constant. The box will move
alternately FORWARD, SIDEWAYS, BACKWARD, SIDEWAYS, and so on. Therefore changes in these adverbials terminate event nodes.

Changes in type (2) adverbials must be preceded by a change in TRAJECTORY, but some of these changes may not cause termination (see A.I.2.6). For example, the curving path of a ball thrown into the air might fail to curve sharply enough to cause the trajectory derivative to become large. It is also quite possible that for such a small object an accurate trajectory cannot be measured at all. It is, however, useful to say that the ball is going UP but now it is coming DOWN. Therefore the only changes causing termination here are when UP changes to DOWN or vice versa.

Changes in type (3) terminate event nodes if and only if there is a change in a CONTACT relation or a VISIBILITY relation. All type (3) terms are candidates except ALONG, APART, BY and THOUGH. APART, ALONG and THROUGH are treated specially in the type (5) and (6) groups. BY does not cause termination because no CONTACT relation is at issue and no question of VISIBILITY need arise separately from other terms. Of the remaining terms often associated with changing CONTACT relations (ACROSS, AGAINST, AWAY-FROM, FROM, IN, OFF-OF, ONTO, OVER, TO, UNDER), termination is dependent on the observation or inference of CONTACT. The change is therefore effectively instantaneous; either the old relation holds no longer (terminating its event node) or a new relation is established (terminating the old node and occupying a position in DIRECTION on the new node). If the making or breaking of a CONTACT relation is not ascertained, then the term merely gets stacked on the current node's DIRECTION list. This is exactly what happens with BY.

The situation with regard to VISIBILITY is similar, but more straightforward; something is or is not visible. The adverbials dependent on VISIBILITY are AWAY (if by disappearance), BEHIND (via occlusion), IN(TO) and OUT-OF (because the observer is likely to see the SUBJECT on only one of the INSIDE or the OUTSIDE), and perhaps UNDER (because the thing ABOVE may occlude the SUBJECT).
If the CONTACT is made or the VISIBILITY established, the adverbial goes into the new node's DIRECTION list. If the CONTACT is broken or VISIBILITY lost, the adverbial remains in the terminated node's DIRECTION list as one of its front elements.

Some of the type (4) terms are only indicators of current source and target and therefore do not cause termination when the target changes (TOWARD and IN-THE-DIRECTION-OF). The source cannot change so AWAY-FROM does not cause termination. INWARD changes to OUTWARD when a central point of the object is passed (no TRAJECTORY or other change need be involved) though OUTWARD changing to INWARD requires a TRAJECTORY change of about 180° somewhere. IN(2) and OUT(2) also may not exhibit a significant TRAJECTORY change and again must be based on some metric distance to a central point. (Think of a ball on an elastic band, first moving towards the band, past, then being drawn back at the limit of the band's stretch. This motion would be a repeated IN and OUT event.)

The type (5) terms relate paths of the SUBJECT to other objects. They cause termination of the old node when they come into effect, and terminate their own nodes when they cease to describe the path relationship. APART and TOGETHER(1) are unique in that they break and combine, respectively, SUBJECT, AGENT or INSTRUMENT lists, creating new event nodes. WITH does not cause termination because it is the description of events between TOGETHER and APART. Since AFTER and AHEAD-OF are temporally-displaced, similar events with differing SUBJECTs, they are analogous to repetitive events for the same SUBJECT: creating a new node when the repetition is discovered and terminating it when broken.

The only anomalous concept in this group is ALONG — the path of the SUBJECT parallels spatial features of another (perhaps stationary) object, rather than its path. ALONG is similar to WITH; termination comes with the making or breaking of the parallel relationship, but there are no separate concepts such as TOGETHER and APART to mark the change.
The type (6) directional concepts include higher level events and the basic repetitions. These terminate the current event nodes as described in section IV.2.2. The repeated events (BACK-AND-FORTH, etc.) are terminated when the repetition ceases. The other two (BACK and THROUGH) create event nodes containing themselves in the DIRECTION list, thus terminating whatever came before (see section IV.2.2).

A.I.2.6. Changes in TRAJECTORY.

The changes in TRAJECTORY that are most important are those that change its derivative significantly. Clearly a change in the derivative from or to zero is important — the start and end of a turn. A momentarily large derivative settling back to smaller values indicates a probable trajectory derivative discontinuity, that is, a collision. This clue is of the utmost importance in inferring CONTACT relations between objects when none were directly observed. If the actual point was not seen, the contacted object is likely to be the closest object to the SUBJECT related by (AT object).

The termination condition below selects the start of a turn (increase from zero), but not the end. This makes the condition simple, but also appropriate in the following sense: once the turn is begun, where it ends (or how it ends) is less important than the final trajectory which is always saved in the TRAJECTORY queue and remains after termination. The actual path of an object is less important than its changing locative relations, because the former are dependent on derivative thresholds and various sources of pictorial error, while the latter are dependent on the actual objects in the scene. It is interesting to note that children take this attitude with regards to the paths of objects in space [Piaget(1946)].

Since derivatives in a discrete space (the frame or view times) depend on the sampling rate, instantaneous changes in trajectory will usually be smoothed out to lesser derivative values. As the sampling interval increases, some changes may be swamped completely. One of our basic assumptions is a sampling rate which provides a faithful representation of the activities in the world.
Figure IV.1 shows a schematic path of a point and its successive trajectories. Let \( E \) be the current event node being considered. Intervals \( i^{1-2} \) and \( i^{2-3} \) are defined by:
\[
\begin{align*}
\text{i}^{1-2} & := \text{TC} - \text{TL}; \\
\text{i}^{2-3} & := \text{TN} - \text{TC}.
\end{align*}
\]

If
\[
\begin{align*}
\text{CURRENT-TRAJECTORY} & = \text{NIL} \text{ and NEW-TRAJECTORY} \neq \text{NIL} \\
\text{or} & \\
\text{CURRENT-TRAJECTORY} & \neq \text{NIL} \text{ and NEW-TRAJECTORY} = \text{NIL}
\end{align*}
\]
then this case is equivalent to a change of VELOCITY from or to zero and is handled by A.2.7 accordingly. If
\[
\begin{align*}
\text{CURRENT-TRAJECTORY} & = \text{NIL} \text{ and NEW-TRAJECTORY} = \text{NIL}
\end{align*}
\]
there is nothing to do. Therefore assume
\[
\begin{align*}
\text{CURRENT-TRAJECTORY} & = \text{NIL} \text{ and NEW-TRAJECTORY} = \text{NIL}.
\end{align*}
\]
If
\[
\begin{align*}
\text{LAST-TRAJECTORY}(E) & = \text{NIL}
\end{align*}
\]
then set
\[
\begin{align*}
\text{LAST-TRAJECTORY}(E) & := \text{CURRENT-TRAJECTORY}(E);
\end{align*}
\]
since this is only possible when \( E \) is created by a non-NIL trajectory. Define
\[
\begin{align*}
\text{D}^{1-2} & = \text{arc(LAST-TRAJECTORY}(E), \\
& \text{CURRENT-TRAJECTORY}(E))
\end{align*}
\]
where \text{arc} is the arclength (hence the angle) between two spherical points (section II.1).

(Not necessarily in one plane.)

Figure IV.1. Trajectory derivatives

-110-
The quantity 
\[ \frac{D_{1-2}}{i_{1-2}} \]
will be taken as the derivative of TRAJECTORY at the last observation time (TC).

Since the reliability of the trajectory computation depends on several factors, selecting an absolute threshold for trajectory error is difficult. We will use 5\% in \(2\pi\) or 180° as the threshold \(\delta\) for determining significant derivative changes.

When NEW-TRAJECTORY is measured, compute
\[ D_{2-3} = \arcsin(\text{CURRENT-TRAJECTORY}(F), \text{NEW-TRAJECTORY}(F)) \]
and terminate \(F\) if
\[ D_{2-3} > \delta/2 \]
and
\[ D_{2-3}/i_{2-3} > \text{FACTOR}(D_{1-2}/i_{1-2}). \]

A value \(\text{FACTOR} = 2\) seems to work; another value could be chosen by experimentation.

If termination does occur then LAST-TRAJECTORY is destroyed. Information on the sign of the derivative which might be obtained from LAST-TRAJECTORY is absorbed into the relations in the DIRECTION list — AROUND, OVER, ALONG, and so on.

A.1.2.7. Changes in VELOCITY.

A change in VELOCITY from zero to a positive value terminates the current event node and enters STARTS in the new node's VELOCITY RATES list. A change from some positive value to zero also causes termination but enters STOPS in the terminated node's VELOCITY RATES list. These changes are detected directly from pictorial data.

If the concepts LURCHES and SLACKENS are important, one can define similar thresholds and derivatives for VELOCITY changes analogous to those for TRAJECTORY changes. The threshold will of course depend on the accuracy to which VELOCITY values are known or estimated. We will not use these here.

A.1.2.8. Changes in AXIS.

This seems to be a very fuzzy area for linguistic terms, making the choice of terminating conditions difficult. The only one that seems useful is a reversal of rotation. Since ANGULAR VELOCITY is always non-negative, a reversal of rotation
corresponds to a change in AXIS direction by 180°, with no intermediate values. (If an interval of non-rotation before reversal were observed then termination for ANGULAR VELOCITY (A.I.2.9) will have taken effect. Whether or not the actual instant of reversal were observed the termination by AXIS would be invoked.) Any intermediate values for AXIS signal rotation of the AXIS itself, not rotational reversal.

Since an AXIS is a direction, there is at least the same potential for change as with TRAJECTORY. Determining the direction of AXIS accurately is more difficult; there may even be an error of almost 90° when view sequences alone are used. Orientation directions will be known with more accuracy if the object has recognizable rectangular features (see section v.1.1). This is enough to dampen any attempt to analyze subtle variations in AXIS direction changes, although derivatives with respect to time from, to or through zero might be important. Appealing once again to the kinds of motion we expect to see, however, these events are not commonly described. An object spinning, so it has an AXIS, and then moving so as to displace the AXIS angle, is a descriptively difficult situation. (Think of a spinning top just as it begins to slow down enough to "fall over." Certain movements in diving and gymnastics are also of this type; the concept TWISTS might be the most relevant.) Something changing its rate of AXIS change to zero is not common either; likewise a change through zero: again the image of a spinning top provides (in projection) motion of this type (PRECESSION).

What these last examples show is not that these changes in AXIS are cause for termination, but rather that the concomitant changes in object ORIENTATION are significant. If the AXIS intersects the SUBJECT then changes in ORIENTATION of the SUBJECT cause (or at least occur along with) changes in the AXIS direction, but the AXIS and its spinning motion need not even be present. Thus if one did not know tops spun, and watched an upright one slowly tip over and bounce about, the fact it was spinning can effect an explanation for the motion but it is not especially important for the description of the event (for
example, "tipped over"). This event is signalled by the making
or breaking of various CONTACT relations (CONTACT tip to CONTACT
side) and termination for this action is assured by relations in
the DIRECTION property.

A.I.1.2.8. Changes in ANGULAR-VELOCITY.

A change in ANGULAR-VELOCITY from zero to a positive value
terminates the current event node and enters STARTS in the new
node's ANGULAR-VELOCITY RATES list. A change from some positive
value to zero also causes termination but enters STOPS in the
terminated node's ANGULAR-VELOCITY RATES list. These changes
are detected directly from view or orientation data.

Since the situation on measuring accurate values for
ANGULAR-VELOCITY is similar to that for VELOCITY, angular
accelerations and decelerations will not be significant. The
same arguments therefore apply and so further termination
conditions for ANGULAR-VELOCITY will not be considered here.

A.I.1.2.10. Changes in NEXT are not meaningful.

A.I.1.2.11/12. Changes in START-TIME and END-TIME.

Changes in these properties are not meaningful unless a
REPEAT-PATH is being described, see A.I.2.13.

A.I.1.2.13. Changes in REPEAT-PATH.

When new data fails to match the appropriate sub-event node
of a REPEAT-PATH event node E, E is terminated. The definition
of "match" is given in section IV.2.2.

A.I.1.3. Maintaining event nodes.

If the new information does not cause termination of the
event node E, then the property queues are shifted:

\[
\begin{align*}
\text{TRAJECTORY}(E) & := \text{SHIFT}(\text{TRAJECTORY}(E)) ; \\
\text{VELOCITY}(E) & := \text{SHIFT}(\text{VELOCITY}(E)) ; \\
\text{AXIS}(E) & := \text{SHIFT}(\text{AXIS}(E)) ; \\
\text{ANGULAR-VELOCITY}(E) & := \text{SHIFT}(\text{ANGULAR-VELOCITY}(E)) ; \\
\text{END-TIME}(E) & := \text{SHIFT}(\text{END-TIME}(E)) .
\end{align*}
\]

Any object property queues are similarly updated even though no
termination algorithm for object nodes is defined.

A.I.1.3.1. Low level information in TRAJECTORY, VELOCITY,
AXIS and ANGULAR-VELOCITY is computed from changes in the
SUBJECT's image by demons using the algorithms described in
sections V.1.3 and V.1.4. Object nodes are updated with respect
to VISIBILITY, MOBILITY, LOCATION and ORIENTATION. Locative

-113-
relations to nearby objects are found and other changes in the static description are noted if discovered.

The RATES lists in VELOCITY and ANGULAR VELOCITY are updated by demons watching changes in these properties to or from zero or to or from some constant.

A.1.3.2 Intermediate level information in DIRECTION is updated by the adverbial demons as described in section III.3.2.2.

A.1.3.3 AGENTS, INSTRUMENTS, and multiple SUBJECTS are derived from MOBILITY properties of the objects involved and recognition of common paths, all higher level functions. One of the description condensation demons in section IV.2.1 produces AGENT and INSTRUMENT values on event nodes.

1.3 What does an event mean?

The termination algorithm for event nodes motivates a theorem. The claim we make is that the algorithm is the finest meaningful partition of the movements in the image sequence into distinct events. The hypothesis of the assertion is the natural environment being observed and the conceptual description desired. The conclusion is that an event node produced from this algorithm describes either the lack of motion or else an unimpeded, simple linear or smoothly curving (or rotating) motion of the SUBJECT with no CONTACT changes. In addition, the orientation directions of the SUBJECT (if they exist) do not deviate much from the trajectory.

The proof should almost be obvious by now. If the SUBJECT is MOBILE but stationary there is an event node to that effect. Otherwise some translation (LOCATION change) or rotation (ORIENTATION change) must occur. Neither can stop nor start without termination from VELOCITY or ANGULAR VELOCITY, hence the SUBJECT is moving. By the TRAJECTORY termination, there can be no significant change in the derivative of the TRAJECTORY (if changing LOCATION) so the movement is linear or smoothly curving in space. Similarly there can be no significant change in the derivative of the AXIS. Changes in TRAJECTORY or AXIS themselves are either preceded by changes in VELOCITY or
ANGULAR VELOCITY, respectively, or else are admissible under the smooth derivative condition. There can be no change in the AGENT or INSTRUMENT without a CONTACT relation change, and there can be none of these without the DIRECTION termination catching them. CONTACT changes normally arise from significant TRAJECTORY derivative changes anyway. The motion of the SUBJECT must therefore be effectively unimpeded and the CONTACT changes yield the significant changes in the scene description. These are placed at the beginning or at the end of the event depending on whether CONTACT was made or broken, respectively. By the condition on orientation adverbials FORWARD, BACKWARD and SIDEWAYS, the SUBJECT cannot alter its orientation with respect to the trajectory by more than 90°. If the viewpoint is changed then REFERENCE will catch it; this way an abrupt TRAJECTORY change as a result of observer movement will not affect the continuity of the SUBJECT's motion. Now if by moving in a uniform unimpeded manner the SUBJECT should break a repeated sequence of events, the REPEAT-PATH will find one time interval too long and will terminate the event (or else another adverbial will apply), so we always have one check on the basic motion if it does not change otherwise. Lastly, if the SUBJECT itself changes it can only disappear or change one of its physical characteristics. The former is noted by changes in DIRECTION; the latter is not allowed by our assumptions. Thus the assertion is proved.

From now on an event node with a NIL REPEAT-PATH will be called a primitive event. Those with a non-NIL REPEAT-PATH are higher level events and will be the subject of section IV.2.2.

Throughout all of Algorithm I there is only one basic source of possible ambiguity in the event description generated, namely whether or not an object actually contacted another. We have already seen that this is difficult to determine statically, and possible to determine dynamically, though imperfectly, from trajectory changes. We could propose that every possible choice of contact or non-contact signals the creation of two new data base states (event alternatives), one hypothesizing contact, the other not. A potential exists for an
exponential explosion of hypothetical events. Our alternative is that the contact or non-contact duality is implicitly stored in the single stream of events. If contact is actually observed or inferred, an AGAINST adverbial will be instantiated first on the appropriate event node. If contact was not determined, then another concept will apply (for example, BY or TO or OVER) in which the contact is optional: AT or even NEAR-TO may be sufficient. While going over a fence, we may or may not touch it. Here the choice is probably irrelevant, but in more important interactions between people, contact or non-contact may be the subject of much debate, especially since the choice of AGENTS depends on contact.

1.4. Two examples

The methods of this chapter can be applied to data from Examples 1 and 2 of the Introduction. We will simplify the setting in Example 1 to make the analysis more straightforward but Example 2 will be used as illustrated.

1.4.1. The car

We will describe a portion of the scene illustrated in the movie of Example 1. The sequence (Figure IV.2) shows the car leaving the driveway of the house, observed from a viewpoint different from that in the movie. The result will be equivalent however.

We will take the lower front edge of the house as the REFERENCE direction so that NORTH is towards the right of each frame. By our convention NORTH will correspond to $\theta=0$, thus we are looking at the point $(90^\circ,90^\circ)$ in the picture center.

To begin we generate a static description of the scene at time 1.00. Figure IV.3 gives some of the relevant relations between the objects. We will assume that driveways and roads are defined by their boundaries. The locative relations are determined with respect to the front orientation of the house, as described in section V.1.5.1. The orientation is determined in section V.1.1 by the plane slant algorithm to be

- 116 -
Figure IV.2a. The car and the house
Figure IV.2b. The car and the house
(-89.90, 90.00). The car is initially IN-BACK-OF-AND-RIGHT-OF
the house.

At time 1.00 an event node for each MOBILE object is
created, namely for the car:

```plaintext
event C.1

SUBJECT CAR
VELOCITY (* 0. 0.)
ANGULAR VELOCITY (* 0. 0.)
START-TIME NIL
END-TIME (* 1.00 1.00)
```

(We will not be interested in the events for the OBSERVER since
we know he will not move in this sequence.) By time 3.00 there
is enough displacement in the car image coordinates to compute
TRAJEKTORY. The fact SUPPORTED-BY(CAR DRIVEWAY) is used to
sharpen the computation by assuming the motion is parallel to
the ground plane.

In general the trajectory should be averaged over all
points, but we will use a few easily recognized points and
average over these. For convenience, Table IV.1 gives the rough
trajectory and orientation change values for the car between
time 1.00 and 3.00 and between each frame thereafter. Note that
the direction (-90.00, 90.00) is towards the observer, so
(0.00, 90.00) is at the right. Orientation changes and
trajectory changes (from successive differences) do not always
match since they are computed by different algorithms:
orientation from section V.1.1 and trajectory from section
V.1.3.1.

```
Table IV.1.

time-to-time mean trajectory mean orientation change
1.00 - 3.00 (-88.0, 90.0) (0.0, 0.0)
3.00 - 4.00 (-88.0, 90.0) (0.0, 0.0)
4.00 - 5.00 (-68.0, 90.0) (28.0, 0.0)
5.00 - 6.00 (-54.0, 90.0) (11.0, 0.0)
6.00 - 7.00 (-41.0, 90.0) (12.0, 0.0)
7.00 - 8.00 (-35.0, 90.0) (9.0, 0.0)
8.00 - 9.00 (-27.0, 90.0) (6.0, 0.0)
9.00 - 10.00 (-17.0, 90.0) (4.0, 0.0)
10.00 - 11.00 (-9.0, 90.0) (6.0, 0.0)
11.00 - 12.00 (0.0, 90.0) (6.0, 0.0)
12.00 - 13.00 (1.0, 90.0) (1.0, 0.0)
13.00 - 14.00 (1.0, 90.0) (-5.0, 0.0)
14.00 - 15.00 (-4.0, 90.0) (-5.0, 0.0)
```
Figure IV.3. Car and house data base (time 1.00)
At time 3.00 the event node C.1 is terminated by the change in location:

```
EVENT C.1
SUBJECT CAR
AGENT CAR
VELOCITY (* 0. 0.)
ANGULAR-VELOCITY (0. 0. 0.)
TRAJECTORY ((-88. 0. 90. 0.) NIL NIL)
START-TIME NIL
END-TIME (3.00 2.00 1.00)
```

becomes

```
EVENT C.1
SUBJECT CAR
AGENT CAR
VELOCITY (0. NIL)
ANGULAR-VELOCITY 0.
START-TIME NIL
END-TIME 2.00
```

```
EVENT C.2
SUBJECT CAR
AGENT CAR
VELOCITY (* * 0. (STARTS))
ANGULAR-VELOCITY (* 0. 0.)
TRAJECTORY (* (-88. 0. 90. 0.) NIL)
DIRECTION ({EASTWARD (TOWARD OBSERVER)})
START-TIME 2.00
END-TIME (* 3.00 2.00)
```

The EASTWARD direction comes directly from the REFERENCE direction. We also generate (TOWARD OBSERVER) since the observer lies in the (-90.0, 90.0) direction from the car front, that is, in the TRAJECTORY direction. Although the IN-BACK-OF(CAR HOUSE) relation is deleted, no changes are made to the DIRECTION list. The car is its own AGENT since it is MOTILE and nothing else was observed that would cause the movement to be attributed to any other force.

At time 4.00 the TRAJECTORY does not change, so C.2 becomes:

```
EVENT C.2
SUBJECT CAR
AGENT CAR
VELOCITY (* * + (STARTS))
ANGULAR-VELOCITY (* 0. 0.)
TRAJECTORY (* (-88. 0. 90. 0.) (-88. 0. 90. 0.))
DIRECTION ({EASTWARD (TOWARD OBSERVER)})
START-TIME 2.00
END-TIME (* 4.00 3.00)
```
At time 5.00 several things happen. (See Figure IV.4.) The relation SUPPORTED-BY(CAR ROAD) is added, while IN-BACK-OP(CAR DRIVEWAY) is deleted and replaced by IN-FRONT-OP(CAR DRIVEWAY). The SURROUNDED-BY(CAR DRIVEWAY) relation therefore remains. The CONTACT change triggers termination because of the adverbial (ONTO ROAD). In addition the TRAJECTORY begins to change significantly (A.I.2.6) and there is an orientation change of 28°. C.2 is split:

**event C.2**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT CAR</td>
</tr>
<tr>
<td>VELOCITY (+ (STARTS))</td>
</tr>
<tr>
<td>ANGULAR VELOCITY 0</td>
</tr>
<tr>
<td>TRAJECTORY (-88.0° 90.0°)</td>
</tr>
<tr>
<td>DIRECTION (((ONTO ROAD) (EASTWARD (TOWARD OBSERVER)))</td>
</tr>
<tr>
<td>START-TIME 2.00</td>
</tr>
<tr>
<td>END-TIME 4.00</td>
</tr>
</tbody>
</table>

**event C.3**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT CAR</td>
</tr>
<tr>
<td>VELOCITY (+ +)</td>
</tr>
<tr>
<td>ANGULAR VELOCITY (* 28.0. (STARTS))</td>
</tr>
<tr>
<td>AXIS (* (0.0 0.0) NIL)</td>
</tr>
<tr>
<td>TRAJECTORY (* (-88.0° 90.0°) (-88.0° 90.0°))</td>
</tr>
<tr>
<td>DIRECTION (((EASTWARD (TOWARD OBSERVER)))</td>
</tr>
<tr>
<td>START-TIME 4.00</td>
</tr>
<tr>
<td>END-TIME (* 5.00 4.00)</td>
</tr>
</tbody>
</table>

At time 6.00 a new locative relation IN-FRONT-OP(CAR HOUSE) is noted. Precondition 2 for (AROUND(1) HOUSE) now has two consecutive location relations on its list, but no demon is yet executable. The adverbial NORTHWARD is found. The TRAJECTORY change and AXIS direction remain effectively constant.

**event C.3**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT CAR</td>
</tr>
<tr>
<td>VELOCITY (+ +)</td>
</tr>
<tr>
<td>ANGULAR VELOCITY (* 11.28. (STARTS))</td>
</tr>
<tr>
<td>AXIS (* (0.0 0.0) (0.0 0.0))</td>
</tr>
<tr>
<td>TRAJECTORY (* (-58.0° 90.0°) (-68.0° 90.0°))</td>
</tr>
<tr>
<td>DIRECTION (NORTHWARD AND EASTWARD (EASTWARD (TOWARD OBSERVER)))</td>
</tr>
<tr>
<td>START-TIME 4.00</td>
</tr>
<tr>
<td>END-TIME (* 6.00 5.00)</td>
</tr>
</tbody>
</table>

At time 7.00 the locative relation LEFT-OP(CAR DRIVEWAY) disappears and consequently SURROUNDED-BY(CAR DRIVEWAY) is deleted. This creates (OUT-OF DRIVEWAY). The former AT(CAR DRIVEWAY) changes to NEAR-TO(CAR DRIVEWAY) creating (FROM
Figure IV.4. Car and house database (time 5.00)
DRIVEWAY. (See Figure IV.5.) We may also suppose that by this time picture analysis has determined that the car front is in fact the visible side; FORWARD is added at the end of DIRECTION by the definition in section III.3.2.2.

**event C.3**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
<th>AGENT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY</td>
<td>(* + +)</td>
</tr>
<tr>
<td>ANGULAR VELOCITY</td>
<td>(* 14.0 11.0 (STARTS))</td>
</tr>
<tr>
<td>AXIS</td>
<td>(* 0.0 0.0)</td>
</tr>
<tr>
<td>TRAJECTORY</td>
<td>(* -41.0 90.0)</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>((OUT-OF DRIVEWAY) (FROM DRIVEWAY))</td>
</tr>
<tr>
<td></td>
<td>(NORTHWARD-AND-EASTWARD)</td>
</tr>
<tr>
<td></td>
<td>(EASTWARD (TOWARD OBSERVER))</td>
</tr>
<tr>
<td></td>
<td>FORWARD</td>
</tr>
<tr>
<td>START-TIME</td>
<td>4.00</td>
</tr>
<tr>
<td>END-TIME</td>
<td>(* 7.00 6.00)</td>
</tr>
</tbody>
</table>

With time 8.00 SUPPORTED-BY(CAR DRIVEWAY) no longer holds, causing instantiation of (OFF-OF DRIVEWAY) on C.3 and terminating it.

**event C.3**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
<th>AGENT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY</td>
<td>(* NIL)</td>
</tr>
<tr>
<td>ANGULAR VELOCITY</td>
<td>(14.0 (STARTS))</td>
</tr>
<tr>
<td>AXIS</td>
<td>(0.0 0.0)</td>
</tr>
<tr>
<td>TRAJECTORY</td>
<td>(-41.0 90.0)</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>((OFF-OF DRIVEWAY)</td>
</tr>
<tr>
<td></td>
<td>(FROM DRIVEWAY))</td>
</tr>
<tr>
<td></td>
<td>(NORTHWARD-AND-EASTWARD)</td>
</tr>
<tr>
<td></td>
<td>(EASTWARD (TOWARD OBSERVER))</td>
</tr>
<tr>
<td></td>
<td>FORWARD</td>
</tr>
<tr>
<td>START-TIME</td>
<td>4.00</td>
</tr>
<tr>
<td>END-TIME</td>
<td>7.00</td>
</tr>
</tbody>
</table>

**event C.4**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
<th>AGENT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY</td>
<td>(* + +)</td>
</tr>
<tr>
<td>ANGULAR VELOCITY</td>
<td>(* 12.0 14.0)</td>
</tr>
<tr>
<td>AXIS</td>
<td>(* 0.0 0.0)</td>
</tr>
<tr>
<td>TRAJECTORY</td>
<td>(* -35.0 90.0)</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>((NORTHWARD-AND-EASTWARD FORWARD))</td>
</tr>
<tr>
<td>START-TIME</td>
<td>7.00</td>
</tr>
<tr>
<td>END-TIME</td>
<td>(* 8.00 7.00)</td>
</tr>
</tbody>
</table>

At time 9.00 the adverbial NORTHWARD is pushed onto DIRECTION(C.4) to indicate the new course.

At time 10.00 we find LEFT-OF(CAR HOUSE) and since LEFT-OF is the fourth consecutive location relation between the house (and no more than two held at one time), (AROUND(1) HOUSE) is placed in DIRECTION(C.4). C.4 also gets (AWAY-FROM DRIVEWAY) since FAR-FROM(CAR DRIVEWAY) replaces NPAR-TO(CAR DRIVEWAY).
At time 12.00 the ANGULAR VELOCITY drops to zero, causing termination of C.4. At the same time NEAR-TO (CAR HOUSE) changes to FAR-FROM (CAR HOUSE) (Figure IV.6) creating (AWAY-FROM HOUSE).

**event C.4**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
<th>AGENT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY (+ NIL)</td>
<td></td>
</tr>
<tr>
<td>ANGULAR-VELOCITY (12. (STOPS))</td>
<td></td>
</tr>
<tr>
<td>AXIS (0.0, 0.0)</td>
<td></td>
</tr>
<tr>
<td>TRAJECTORY (-90.0, 90.0)</td>
<td></td>
</tr>
<tr>
<td>DIRECTION (AWAY-FROM HOUSE) (AWAY-FROM DRIVEWAY) (AROUND (1) HOUSE) (NORTHWARD (NORTHWARD-AND-FASTWARD FORWARD))</td>
<td></td>
</tr>
<tr>
<td>START-TIME 11.00</td>
<td>END-TIME 11.00</td>
</tr>
</tbody>
</table>

**event C.5**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
<th>AGENT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY (* + *)</td>
<td></td>
</tr>
<tr>
<td>ANGULAR-VELOCITY (* 0.12.)</td>
<td></td>
</tr>
<tr>
<td>AXIS (* NIL) (0.0, 0.0)</td>
<td></td>
</tr>
<tr>
<td>TRAJECTORY (* (0.0, 90.0) (-90.0, 90.0))</td>
<td></td>
</tr>
<tr>
<td>DIRECTION (NORTHWARD FORWARD)</td>
<td></td>
</tr>
<tr>
<td>START-TIME 11.00</td>
<td>END-TIME (* 12.00 11.00)</td>
</tr>
</tbody>
</table>

Nothing else changes C.5 significantly until after time 15.00 when the car disappears from view, adding AWAY:

**event C.5**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
<th>AGENT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY (+ NIL)</td>
<td></td>
</tr>
<tr>
<td>ANGULAR-VELOCITY (0.0, NIL)</td>
<td></td>
</tr>
<tr>
<td>TRAJECTORY (0.0, 90.0)</td>
<td></td>
</tr>
<tr>
<td>DIRECTION (AWAY (NORTHWARD FORWARD))</td>
<td></td>
</tr>
<tr>
<td>START-TIME 11.00</td>
<td>END-TIME 15.00</td>
</tr>
</tbody>
</table>

**event C.6**

<table>
<thead>
<tr>
<th>SUBJECT CAR</th>
<th>AGENT CAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY (* NIL *)</td>
<td></td>
</tr>
<tr>
<td>ANGULAR-VELOCITY (* NIL 0.)</td>
<td></td>
</tr>
<tr>
<td>TRAJECTORY (* NIL (0.0, 90.0))</td>
<td></td>
</tr>
<tr>
<td>START-TIME 15.00</td>
<td>END-TIME (* 16.00 15.00)</td>
</tr>
</tbody>
</table>

If we "write out" these event nodes using the canonical verbs MOVES and TURNS with the adverbial phrases, we obtain the following lengthy, but accurate description:

C.1 There is a CAR.

C.2 The CAR STARTS MOVING TOWARD the OBSERVER and EASTWARD, then ONTO the ROAD.
Figure IV.6. Car and house data base (time 12.00)
C.3 The CAF, while GOING FORWARD, STARTS TURNING, MOVES TOWARD the OBSERVE and FASTWARD, then NORTHWARD-AND- EASTWARD, then FROM the DRIVEWAY and OUT-OF the DRIVEWAY, then OFF-CP the DRIVEWAY.

C.4 The CAF, while GOING FORWARD, MOVES NORTHWARD-AND- EASTWARD, then NORTHWARD, then AROUND the HOUSE and AWAY-FROM the DRIVEWAY, then AWAY-FROM the HOUSE and STOPS TURNING.

C.5 The CAF, while GOING FORWARD, MOVES NORTHWARD, then AWAY.

The canonical form follows easily from the case representation and the convention regarding DIRECTION list orderings. The directional adverbials FORWARD, BACKWARD and SIDEWAYS are interpreted as lasting the duration of the event, hence are written as "while GOING ..." clauses. STARTS is always interpreted at the beginning of the sentence, STOPS at the end. The termination conditions assure its correctness.

There is quite a lot of redundancy in this description but it is only the lowest level, after all. In section IV.2.1 we will return to this example to describe ways to condense it further.

1.4.2. The bouncing ball

For another example we take the beginning of the bouncing ball sequence, Example 2. We will assume that the visual size of the ball is too small for a satisfactory three-dimensional trajectory calculation so that the actual coordinate displacement of the center of the ball will be used to compute a two-dimensional trajectory (see section V.1.3.1). If the actual coordinates of the ball in the figures were used, that is, not as if they were part of a whole hemispherical view, then the ball would not be too small and the three-dimensional trajectory calculation would apply. This choice will not materially affect the result. Table IV.2 gives the trajectory directions between each frame.
Table IV.2.

<table>
<thead>
<tr>
<th>time-to-time</th>
<th>mean trajectory</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85 - 1.00</td>
<td>(90.0°, 90.0°)</td>
</tr>
<tr>
<td>1.00 - 1.15</td>
<td>(90.0°, 116.60°)</td>
</tr>
<tr>
<td>1.15 - 1.30</td>
<td>(90.0°, 135.0°)</td>
</tr>
<tr>
<td>1.30 - 1.45</td>
<td>(90.0°, 153.40°)</td>
</tr>
<tr>
<td>1.45 - 1.60</td>
<td>(90.0°, 161.50°)</td>
</tr>
<tr>
<td>1.60 - 1.75</td>
<td>(90.0°, 26.60°)</td>
</tr>
<tr>
<td>1.75 - 1.90</td>
<td>(90.0°, 33.70°)</td>
</tr>
<tr>
<td>1.90 - 2.05</td>
<td>(90.0°, 63.40°)</td>
</tr>
<tr>
<td>2.05 - 2.20</td>
<td>(90.0°, 90.0°)</td>
</tr>
<tr>
<td>2.20 - 2.35</td>
<td>(90.0°, 135.0°)</td>
</tr>
<tr>
<td>2.35 - 2.50</td>
<td>(90.0°, 146.20°)</td>
</tr>
<tr>
<td>2.50 - 2.65</td>
<td>(90.0°, 146.30°)</td>
</tr>
<tr>
<td>2.65 - 2.80</td>
<td>(90.0°, 33.70°)</td>
</tr>
<tr>
<td>2.80 - 3.05</td>
<td>(90.0°, 63.40°)</td>
</tr>
<tr>
<td>3.05 - 3.35</td>
<td>(90.0°, 90.0°)</td>
</tr>
<tr>
<td>3.35 - 3.60</td>
<td>(90.0°, 135.0°)</td>
</tr>
<tr>
<td>3.60 - 3.85</td>
<td>(90.0°, 45.0°)</td>
</tr>
<tr>
<td>3.85 - 4.00</td>
<td>(90.0°, 104.0°)</td>
</tr>
</tbody>
</table>

Figure IV.7 gives a part of the data base description of the static scene at time 0.85. (The GROUND is excluded for clarity.) The SURROUNDED-BY relation is found from LEFT-OF, ABOVE, and RIGHT-OF relations. No SUPPORTED-BY (BALL HAND) is present as the contact surfaces are vertical. The initial event nodes are:

**event B.1**

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>BALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td>PERSON</td>
</tr>
<tr>
<td>VELOCITY</td>
<td>(* 0. 0.)</td>
</tr>
<tr>
<td>START-TIME</td>
<td>NIL</td>
</tr>
<tr>
<td>END-TIME</td>
<td>(* 0.85 0.85)</td>
</tr>
</tbody>
</table>

**event H.1**

<table>
<thead>
<tr>
<th>SUBJECT</th>
<th>HAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td>PERSON</td>
</tr>
<tr>
<td>VELOCITY</td>
<td>(* 0. 0.)</td>
</tr>
<tr>
<td>START-TIME</td>
<td>NIL</td>
</tr>
<tr>
<td>END-TIME</td>
<td>(* 0.85 0.85)</td>
</tr>
</tbody>
</table>

(We will ignore the ANGULAR-VELOCITY property in the diagrams as it plays no role in this example.) The events for the ball will be derived first.

At time 1.00 the ball is observed to change location. STARTS is easily recognized. The TRAJECTORY is computed from the displacement as described, hence towards (90.0°, 90.0°). The DIRECTION adverbials indicate WESTWARD, or relative to the observer's own coordinate frame, (LEFTWARD OBSERVER). We will
Figure IV.7. The ball database (time 0.85)

Figure IV.8. The ball database (time 1.00)

Figure IV.9. The ball database (time 1.30)
use the former for convenience. Something else has happened, too, for the static description is now as in Figure IV.8; that is, the two CONTACT relations are broken. Thus we also obtain (FROM HAND). These changes insure the termination of B.1 and place (FROM HAND) in B.1. Since the ball is now completely unsupported the AGENT GRAVITY acts.

**event B.1**

<table>
<thead>
<tr>
<th>SUBJECT BALL</th>
<th>AGENT PERSON</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTRUMENT HAND</td>
<td></td>
</tr>
<tr>
<td>VELOCITY (0, NIL)</td>
<td></td>
</tr>
<tr>
<td>DIRECTION ((FROM HAND))</td>
<td></td>
</tr>
<tr>
<td>START-TIME NIL</td>
<td></td>
</tr>
<tr>
<td>END-TIME 0.85</td>
<td></td>
</tr>
</tbody>
</table>

**event B.2**

<table>
<thead>
<tr>
<th>SUBJECT BALL</th>
<th>AGENT GRAVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY (+ 0, (STARTS))</td>
<td></td>
</tr>
<tr>
<td>TRAJECOTORY (*(90.0, 90.0), NIL)</td>
<td></td>
</tr>
<tr>
<td>DIRECTION (WESTWARD)</td>
<td></td>
</tr>
<tr>
<td>START-TIME 0.85</td>
<td></td>
</tr>
<tr>
<td>END-TIME (1.00, 0.85)</td>
<td></td>
</tr>
</tbody>
</table>

At time 1.15 termination occurs again, triggered by the TRAJECOTORY derivative change of 26.6° (A.I.2.6).

**event B.2**

<table>
<thead>
<tr>
<th>SUBJECT BALL</th>
<th>AGENT GRAVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY (+ {STARTS})</td>
<td></td>
</tr>
<tr>
<td>TRAJECOTORY (90.0, 90.0)</td>
<td></td>
</tr>
<tr>
<td>DIRECTION (WESTWARD)</td>
<td></td>
</tr>
<tr>
<td>START-TIME 0.85</td>
<td></td>
</tr>
<tr>
<td>END-TIME 1.00</td>
<td></td>
</tr>
</tbody>
</table>

**event B.3**

<table>
<thead>
<tr>
<th>SUBJECT BALL</th>
<th>AGENT GRAVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY (+ +)</td>
<td></td>
</tr>
<tr>
<td>TRAJECOTORY (*(90.0, 116.6) (90.0, 90.0))</td>
<td></td>
</tr>
<tr>
<td>DIRECTION ((TOWARD GROUND) DOWN WESTWARD))</td>
<td></td>
</tr>
<tr>
<td>START-TIME 1.00</td>
<td></td>
</tr>
<tr>
<td>END-TIME (1.15, 1.00)</td>
<td></td>
</tr>
</tbody>
</table>

The adverbials DOWN and WESTWARD come from the TRAJECOTORY, while the ground is the only object intersecting that trajectory.

At time 1.30 the TRAJECOTORY changes but A.I.2.6 does not cause termination since a curved path is now expected. The static description appears in Figure IV.9. The LEFT-OF and RIGHT-OF relations are broken and these force the SURROUNDED-
BY BALL HAND relation to disappear. The adverbial OUT-OF is applicable, but as there are no CONTACT relation changes, B.3 remains active:

**event B.3**

<table>
<thead>
<tr>
<th>SUBJECT BALL</th>
<th>AGENT GRAVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY</td>
<td>(+ NTL)</td>
</tr>
<tr>
<td>TRAJECTORY</td>
<td>(90.°, 161.5°)</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>((OUT-OF HAND) TOWARD GROUND DOWN WESTWARD)</td>
</tr>
<tr>
<td>START-TIME</td>
<td>1.00</td>
</tr>
<tr>
<td>END-TIME</td>
<td>(1.30 1.15)</td>
</tr>
</tbody>
</table>

Nothing new occurs until time 1.75 when a significant TRAJECTORY change is noted. The DOWN to UP change causes termination of B.3, too (A.1.2.5(2)). We can assert that the TRAJECTORY change was probably caused by contact with the only other object directly in the path of the ball, the ground. To make this assertion note that the ball is NOT-MOTILE and there are no known SUPPORTED-BY relations to any other object. Hence the preconditions for AGAINST are satisfied and we have (OFF-OF GROUND) and (AGAINST GROUND). As this is a CONTACT change, it goes into DIRECTION(B.4), terminating it. Gravity is still the AGENT since the ball is again unsupported.

**event B.4**

<table>
<thead>
<tr>
<th>SUBJECT BALL</th>
<th>AGENT GRAVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>START-TIME</td>
<td>1.60</td>
</tr>
<tr>
<td>END-TIME</td>
<td></td>
</tr>
</tbody>
</table>

**event B.5**

<table>
<thead>
<tr>
<th>SUBJECT BALL</th>
<th>AGENT GRAVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>VELOCITY</td>
<td>(+ NTL)</td>
</tr>
<tr>
<td>TRAJECTORY</td>
<td>(90.°, 161.5°)</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>(UP WESTWARD)</td>
</tr>
<tr>
<td>START-TIME</td>
<td>1.60</td>
</tr>
<tr>
<td>END-TIME</td>
<td>(1.75 1.60)</td>
</tr>
</tbody>
</table>
At time 1.90 things are not much different, but at time 2.05 there is enough trajectory change to terminate B.5 as the ball begins to reach the top of its bounce.

event B.5

SUBJECT BALL
AGENT GRAVITY
VELOCITY (+ NIL)
TRAJECTORY (90.0, 33.70)
DIRECTION ((UP WESTWARD))
START-TIME 1.60
END-TIME 1.90

event B.6

SUBJECT BALL
AGENT GRAVITY
VELOCITY (+ *)
TRAJECTORY (* (90.0, 63.40) (90.0, 33.70))
DIRECTION ((UP WESTWARD))
START-TIME 1.90
END-TIME (* 2.05 1.90)

At time 2.20 the trajectory is horizontal so UP no longer applies, and at 2.35 DOWN is noted, terminating B.6 (A.I.2.5(2)). However, the repetition of the DOWN adverbial causes other things to happen (section IV.2.2) and we will leave the description at 2.05 for now. Rather than finish the bouncing ball, we will return to this point in section IV.2.2.1 to see how the repetitive movement is described.

We will conclude this example with the events for the hand. We assume the hand is PART-OF a MOTILE object PERSON. At times 1.00 and 1.15 the hand is seen to move WESTWARD. After time 1.15 we have:

event H.1

SUBJECT HAND
AGENT PERSON
VELOCITY (0, NIL)
START-TIME NIL
END-TIME 0.85

event H.2

SUBJECT HAND
AGENT PERSON
VELOCITY (* + {STARTS})
TRAJECTORY (* (90.0, 90.0) (90.0, 90.0))
DIRECTION (WESTWARD)
START-TIME 0.85
END-TIME (* 1.15 1.00)

At 1.30 the hand begins to move upwards:
event H.2

SUBJECT HAND
AGENT PERSON
VELOCITY (+ (STARTS))
TRAJECTORY (90.0, 90.0)
DIRECTION (WESTWARD)
START-TIME 0.05
END-TIME 1.15

event H.3

SUBJECT HAND
AGENT PERSON
VELOCITY (+)
TRAJECTORY (90.0, 150.0) (90.0, 90.0)
DIRECTION (UP, WESTWARD)
START-TIME 1.15
END-TIME (* 1.30 1.15)

H.2 is terminated by A.I.2.6.

At time 2.20 the hand becomes NOT-VISIBLE, and by A.I.2.5(3), AWAY stays on H.3 when it terminates.

event H.3

SUBJECT HAND
AGENT PERSON
VELOCITY (+ NIL)
TRAJECTORY (90.0, 150.0)
DIRECTION (AWAY, UP, WESTWARD)
START-TIME 1.15
END-TIME 2.05

Putting these events into English sentences as before and ignoring the ball after 2.05, we obtain:

B.1 The PERSON MOVES the BALL with the HAND FROM the HAND.
B.2 The BALL STARTS MOVING WESTWARD.
B.3 The BALL MOVES DOWN and WESTWARD and TOWARD the GROUND, then OUT-OF the HAND.
B.4 The BALL MOVES AGAINST the GROUND and OFF-OF the GROUND.
B.5 The BALL MOVES UP and WESTWARD.

[Meanwhile]

H.1 The HAND is VISIBLE.
H.2 The PERSON STARTS MOVING the HAND WESTWARD.
H.3 The PERSON MOVES the HAND UP and WESTWARD, then AWAY.
Remember that references to WESTWARD may be augmented with the 
(LEFTWARD OBSERVER) phrase.

**IV.2. Event sequences and repetitions**

Having shown how to build a primitive event description for 
a scenario we begin to put it back together into more condensed 
and higher level events. This process involves three parts:

1. Redundancy elimination.
2. Higher level adverbial recognition.

The second and third parts are closely related because the 
repetitions we will investigate are indicated by BACK-AND-FORTH, 
TO-AND-FROM, and UP-AND-DOWN, all higher level adverbials. 
Conversely, AFTER, AHEAD-OF, TOGETHER(1), and APART depend on 
the notion of repetition. We take up the first part below.

**2.1. Condensing events**

We will consider several demons which shape the primitive 
events into more linguistic descriptions. For example, it is 
quite possible that an event generated by Algorithm I is not 
important in certain contexts, such as a null event of short 
duration or a trajectory change which seemed only to indicate a 
path change rather than a contact relation.

The algorithm for condensing events will be given in terms 
of a set of demons (called Description Demons and given a 
number, thus DD.#) which are meant to be applied whenever they 
match events in the data base. These demons act mostly on 
events and their properties, whereas the adverbials act 
primarily on object properties.

There are two groups of description demons: one whose 
members are always active; the other whose members are activated 
when memory space becomes critical or when specifically invoked 
by some descriptive necessity. Both groups of demons have a 
precondition 1 which is satisfied by the termination of an event 
ode. In addition, the second group senses a boolean variable 
which might be set if, for example, the system were interactive
and were asked to produce a shorter description. These demands begin to approach linguistic transformations in their reorganization or elimination of existing conceptual data.

**Group 1. Mandatory condensations.**

**DD.1.** If a null event \( E \) lasts for less than two frames then split its time interval \( T \) in half and set

\[
\begin{align*}
\text{END-TIME(} \text{PREVIOUS}(E) \text{)} & := \\
\text{END-TIME(} \text{PREVIOUS}(E) \text{)} & + T/2; \\
\text{START-TIME(} \text{NEXT}(E) \text{)} & := \\
\text{START-TIME(} \text{NEXT}(E) \text{)} & - T/2.
\end{align*}
\]

In addition, the null event node is deleted by setting

\[
\text{NEXT(} \text{PREVIOUS}(E) \text{)} := \text{NEXT}(E).
\]

If this node were part of a \textsc{repeat-path} list, it must be deleted from that list as well. Such an event is considered an artifact of the discrete sampling interval, so \textsc{starts} and \textsc{stops} must be modified accordingly.

Assign

\[
\begin{align*}
\text{RVP} & := \text{RATES(VELCocity(} \text{PREVIOUS}(E) \text{))}; \\
\text{RAP} & := \text{RATES(ANGULAR-VELCocity(} \text{PREVIOUS}(E) \text{))}; \\
\text{VPN} & := \text{RATES(VELCocity(} \text{NEXT}(E) \text{))}; \\
\text{RAN} & := \text{RATES(ANGULAR-VELCocity(} \text{NEXT}(E) \text{))}.
\end{align*}
\]

Functions \textsc{first}, \textsc{last}, \textsc{but-first}, and \textsc{but-last} retrieve the first, last, all but the first, and all but the last items of a list.

If \textsc{first}(\textsc{rvp}) = \textsc{stop}s then set

\[
\text{RATES(VELCocity(} \text{PREVIOUS}(E) \text{))} := \text{BUT-FIRST(\textsc{rvp})}.
\]

If \textsc{first}(\textsc{rpp}) = \textsc{stop}s then set

\[
\text{RATES(ANGULAR-VELCocity(} \text{PREVIOUS}(E) \text{))} := \text{BUT-FIRST(\textsc{rpp})}.
\]

If \textsc{last}(\textsc{eun}) = \textsc{start}s then set

\[
\text{RATES(VELCocity(} \text{NEXT}(E) \text{))} := \text{BUT-LAST(\textsc{eun})}.
\]

If \textsc{last}(\textsc{pan}) = \textsc{start}s then set

\[
\text{RATES(ANGULAR-VELCocity(} \text{NEXT}(E) \text{))} := \text{BUT-LAST(\textsc{pan})}.
\]

These merely remove false \textsc{stop}s and \textsc{start}s from the \textsc{rates} lists of the neighboring events. By the event terminations \textsc{a1.2.7} and \textsc{a1.2.9}, these terms will be generated from the null event observed and therefore cannot be attributed to any previous or later movements. Hence the deletions must always be "correct."

**DD.2.** Compress node \( E \) and \( F=\text{NEXT}(E) \) into \( F \) if they satisfy the following. (Assume the \textsc{direction} lists are temporarily converted into single level list structures.)

1. \textsc{subject}(F) = \textsc{subject}(F).
2. \textsc{agent}(F) = \textsc{agent}(F).
3. \textsc{instrument}(F) = \textsc{instrument}(F).
4. \textsc{reference}(F) = \textsc{reference}(F).
5.1 One or both of \textsc{direction}(E) and \textsc{direction}(F) are NIL, or
5.2 The last adverbial in each of \textsc{direction}(E) and \textsc{direction}(F) are the same and are from the set \{'\textsc{forward}', '\textsc{backward}', '\textsc{sideways}'\}, and the next-to-last adverbial in \textsc{direction}(F) is the same as the first adverbial in \textsc{direction}(E), or
5.3 The last adverbial in each of \textsc{direction}(E) and \textsc{direction}(F) are not the same but the last in \textsc{direction}(F) is the same as the first in \textsc{direction}(E).

If so, then set

-136-
RATES(VELOCITY(F)) := APPEND(RATES(VELOCITY(F)), RATES(VELOCITY(E))); 
RATES(ANGULAR VELOCITY(F)) := APPEND(RATES(ANGULAR VELOCITY(F)), RATES(ANGULAR VELOCITY(E))); 
START-TIME(F) := START-TIME(E); 
DIRECTION(F) := APPEND(BUT-LAST(DIRECTION(F)), DIRECTION(E)); 

Or, if case 5.2 was used, the last two adverbials in DIRECTION(F) are deleted:

DIRECTION(F) := APPEND(BUT-LAST(BUT-LAST(DIRECTION(F))), DIRECTION(E)).

Finally, delete F and set

NEXT(PREVIOUS(E)) := F

if PREVIOUS(E) exists. This demon essentially "undoes" a termination for a trajectory change which failed to change any significant relations between the SUBJECT and its environment. DD.2 will almost always delete the initial null event for an object. The ball in Example 2 is an exception since E.1 has a different AGENT than E.2.

DD.3. If the following chain of relations holds between events:

\[
\begin{align*}
\text{SUBJECT}(F_i) &= X \\
\text{INSTRUMENT}(F_i) &= Y.1 \\
\text{AGENT}(F_i) &= \text{NIL} \\
\text{SUBJECT}(F_j) &= Y.1 \\
\text{INSTRUMENT}(F_j) &= Y.2 \\
\text{AGENT}(F_j) &= \text{NIL} \\
\vdots \\
\text{SUBJECT}(F_k) &= Y.(n-1) \\
\text{INSTRUMENT}(F_k) &= Y.n \\
\text{AGENT}(F_k) &= \text{NIL} \\
\end{align*}
\]

and

\[
\begin{align*}
\text{SUBJECT}(E_n) &= Y.n \\
\text{AGENT}(E_n) &= Y.n \\
\end{align*}
\]

for terminated nodes E_j, ..., E_k, then set

\[
\begin{align*}
\text{INSTRUMENT}(E_i) &:= (Y.1 Y.2 ... Y.(n-1)) \\
\text{AGENT}(E_i) &:= Y.n \\
\end{align*}
\]

and delete E_j, ..., E_k from the database entirely. Simply stated, only keep active event nodes for AGENTS and ultimate SUBJECTS. Although DD.3 seems quite drastic, the movements of the intermediate INSTRUMENTS will usually be superfluous.

**Group 2. Optional condensations.**

**DD.4.** Remove all adverbials from DIRECTION which specify directions in the support plane (for example, NORTHWARD, LEFTWARD OBSERVER). This does not include TOWARD or UP OR DOWN. The remaining adverbials specify changes in locative relations; we have eliminated most of the actual path.

**DD.5.** Remove all RATES terms from VELOCITY and ANGULAR VELOCITY except STOPS. Since the onset of movement will usually precipitate some relation changes, STARTS will almost always be redundant. Terms INCREASES and DECREASES,
if used at all, may be replaced in the description by mentioning ACCELERATES or DECELERATES.

DD.6. Replace one or more adverbials to the same object by the most general adverbials (usually FROM or TO).

After DD.4, DD.5 or DD.6 is applied, DD.2 may be used to condense the description further. As a temporary rule in the sense that the changes it produces should not be permanent, we can apply a liberal version of DD.2:

DD.7. This is the same as DD.2 except the condition on DIRECTION is ignored and the new DIRECTION is the concatenation of the two former lists, with redundancies in direction terms removed (for example, eliminate duplicate FORWARD adverbials but not duplicate concepts involving contact).

The non-permanency is necessary should later events create a repeated sequence, and we do not want contrary adverbials (say, UP and DOWN) together on one node.

2.1.1. Condensing the car example

Returning to the events for the car we apply the mandatory condensations and find that the null event C.1 disappears into C.2.

The optional condensations are now applied. DD.4 is very powerful but yields surprisingly good descriptions:

C.2 The car STARTS MOVING TOWARD the OBSERVER, then ONTO the ROAD.

C.3 The car while GOING FORWARD, STARTS TURNING, moves TOWARD the OBSERVER, then FROM the DRIVEWAY and OUT-OF the DRIVEWAY.

C.4 The car, while GOING FORWARD, MOVES AROUND the HOUSE and AWAY-FROM the DRIVEWAY, then AWAY-FROM the HOUSE, then STOPS TURNING.

C.5 The car, while GOING FORWARD, MOVES AWAY.

After applying DD.5 we obtain:

C.2 The car MOVES TOWARD the OBSERVER, then ONTO the ROAD.

C.3 The car, while GOING FORWARD, MOVES TOWARD the OBSERVER, then FROM the DRIVEWAY and OUT-OF the DRIVEWAY.

Note that STARTS TURNING is deleted from C.3 by DD.5. C.4 and C.5 do not change. After applying DD.6 we get:

C.2 The car MOVES TOWARD the OBSERVER, then ONTO the ROAD.

C.3 The car, while GOING FORWARD, MOVES TOWARD the OBSERVER, then FROM the DRIVEWAY.
C.4 The C.I.P., while GOING FORWARD, MOVES AROUND the HOUSE and AWAY-FROM the DRIVEWAY, then AWAY-FROM the HOUSE, then STOPS TURNING.

C.5 The CAR, while GOING FORWARD, MOVES AWAY. Applying DD.7 for a final English-like description, (TOWARD OBSERVER) and FORWARD are redundant (no intervening adverbial contradicts their indications), while (FROM DRIVEWAY) absorbs (AWAY-FROM DRIVEWAY) by re-application of DD.6. This yields:

The CAR MOVES TOWARD the OBSERVER, then ONTO the ROAD, while GOING FORWARD, then FROM the DRIVEWAY, then AROUND the HOUSE, then AWAY-FROM the HOUSE, then STOPS TURNING, then MOVES AWAY.

Note that FROM the DRIVEWAY follows ONTO THE ROAD. This is true given the adverbial definitions and the pictorial configuration: the car is on the road before it leaves the driveway. The position of the "while GOING FORWARD" phrase could be shifted backwards in time to the beginning of the translatory motion, but this may be risky in the general case. We will leave it where it is, since this is primarily a higher level linguistic matter.

2.1.2. Condensing the ball example

Turning to the ball and hand, although we have yet to finish the description, we can apply the mandatory condensations and find DD.2 condenses B.2 into B.3 and H.1, H.2 into H.3.

Applying the optional condensations to the remaining nodes we obtain:

B.1 The BALL MOVES FROM the HAND.
B.3 The BALL MOVES DOWN and TOWARD the GROUND, then OUT-OF the HAND.
B.4 The BALL MOVES AGAINST the GROUND and OFF-OF the GROUND.
B.5 The BALL MOVES UP.
H.3 The HAND MOVES UP, then AWAY.

DD.6 cannot do anything with the relations to HAND in B.1 and B.3 because these nodes are not combinable by DD.2 (they have different AGFMTs).
Algorithm II. Repetition matching.

A.II.1. The preconditions for an adverbial UP-AND-DOWN, BACK-AND-FORTH or TO-AND-FROM are satisfied as described above.

A.II.2. Set

TC := CURRENT-END-TIME(E,j);
E := CURRENT-EVENT-NODE(SUBJECT ...)

{DIRECTION (UP-AND-DOWN)},
{REPEAT-PATH (E,i ... E,j)}
{START-TIME START-TIME(E,i)}
{END-TIME (* TN TC) }

where E,j is the currently active node for this SUBJECT.

A.II.3. Terminate E,j. The repetition will require E,j to become a lower level node. Now E will be the active, higher level event node.

A.II.4. Set

END-TIME(E,i) := CURRENT-END-TIME(E) + START-TIME(E,i) - START-TIME(E,j);
START-TIME(E,i) := CURRENT-END-TIME(E).

This increments the times of the first node in the repeated sequence (E,i) so that it has the same duration, but starts at the end of the first cycle.

A.II.5. E,i is not updated (A.II.4) again until after the contrary term appears.

A.II.6. Nodes between E,i and E,j may be matched by intermediate events. The START-TIMES and END-TIMES are updated as in A.II.4 to maintain the temporal sequence. In any case the stored adverbial (for example UP or DOWN), must match the current adverbial of the same type. Thus the match proceeds though the list of nodes with UP until DOWN is noted again in the input. Then nodes are skipped until the DOWN node is reached; the skipped nodes updated along the way by A.II.4.

A.II.6.1. Data is assumed to match if the orientation adverbial (UP, FORWARD, etc.), SUBJECTS, AGENTS, INSTRUMENTS, and REFERENCEP are matched identically with the assertions on the lower level node. Any other data is added to the active REPEAT-PATH node E. Thus E will be terminated, say by CONTACT changes or trajectory changes, only if these changes generate adverbials which normally cause termination, but do not appear at the lower
level. Thus a raw trajectory change is not likely to terminate F, nor would going by something new.

A.11.6.2 A null event in the input matches the current lower level node, no matter what it is. The REPEAT-PATH may then wait beyond the normal expected end of the cycle. The REPEAT-PATH node will eventually terminate by A.II.8 for certain, so there is no harm in waiting "a little longer" for the repetition to begin again. This would happen to the bouncing ball after several more frames.

A.11.6.3 When a lower level node fails as a pattern, its END-TIME is updated to CURRENT-END-TIME(F) to preserve its time interval, even if this should cause the interval to become zero.

A.11.7 If F is terminated by other than a repetition mismatch, say by an unexpected contact relation, the newly created node inherits REPEAT-PATH(F).

A.11.8 If the observation time for any particular sub-event exceeds, say, three times the duration of the entire repetition, F is terminated. The bound is very subjective and nearly irrelevant, since it depends on how long we want to wait for an event to repeat itself. We would like to terminate repeated events if the repetition appears to have ceased.

2.2.1 The ball again

Now we take up the bouncing ball example again. This is the event sequence constructed so far. Note that B.3 absorbs B.2 by DD.2 as described in section IV.2.1.2.

<table>
<thead>
<tr>
<th>event B.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJECT BALL</td>
</tr>
<tr>
<td>AGENT PERSON</td>
</tr>
<tr>
<td>INSTRUMENT HAND</td>
</tr>
<tr>
<td>VELOCITY (O.1, 0.1)</td>
</tr>
<tr>
<td>DIRECTION (FROM HAND)</td>
</tr>
<tr>
<td>START-TIME 0.0</td>
</tr>
<tr>
<td>END-TIME 0.85</td>
</tr>
</tbody>
</table>

-143-
When we left it the ball was just starting to go DOWN at time 2.35. Because DOWN preceeded UP in B.5, UP-AND-DOWN is invoked by A.II.1. The higher level node B.7 is created by A.II.2 and the times on B.3 are adjusted by A.II.4. Since Algorithm II terminates B.6 (A.II.3), B.5 and B.6 are left with the same DIRECTION list. The condensation DD.2 applies and B.6 absorbs B.5. We now have:
The new DOWN adverbial (as well as (TOWARD GROUND)) match DIRECTION(B.3) hence DIRECTION(B.7) need not contain these terms. WESTWARD is not part of the UP and DOWN motion so it goes onto B.7. Likewise for AGAINST and OFF-OF. These do not terminate B.7 because of A.II.6.1.

At time 2.50 and 2.65 the DOWN movement continues so END-TIME(B.7) is merely updated.

At time 2.80 a trajectory change occurs but as B.7 has a REPEAT-PATH it is not terminated (A.II.6.1) since the generated adverbials (UP and WESTWARD) already appear in B.6 and B.7. We
have finished matching B.3 so its END-TIME is corrected to 2.65 (A.II.6.3).

At time 2.95 B.6 matches since the direction becomes up. B.4 is skipped and its times updated by A.II.6. At time 3.10 we lose the UP adverbial, but as we do not have any contrary indication, B.6 is assumed to match. The events now look like:

**event B.3**

```
SUBJECT BALL
AGENT GRAVITY
VELOCITY (+ NIL)
TRAJECTORY (90.0, 161.5)
DIRECTION (OUT-OF-HAND)
(TOWARD GROUND) DOWN WESTWARD)
START-TIME 2.20
END-TIME 2.65
```

**event B.4**

```
SUBJECT BALL
AGENT GRAVITY
DIRECTION (OFF-OF GROUND) (AGAINST GROUND))
START-TIME 2.65
END-TIME 2.65
```

**event B.6**

```
SUBJECT BALL
AGENT GRAVITY
VELOCITY (+ NIL)
TRAJECTORY (90.0, 63.4)
DIRECTION (UP WESTWARD))
START-TIME 2.65
END-TIME 3.25
```

**event B.1**

**event B.7**

```
SUBJECT BALL
AGENT GRAVITY
VELOCITY (* + *)
TRAJECTORY (* (90.0, 90.0) (90.0, 63.4))
DIRECTION (UP-AND-DOWN WESTWARD)
(OFF-OF GROUND) (AGAINST GROUND))
REPEAT-PATH (B.3 B.4 B.6)
START-TIME 1.00
END-TIME (* 3.10 2.95)
```

At time 3.25 we pick up DOWN again, the match updates END-TIME(B.6) and returns to B.3.

This process continues in the same manner through time 3.70. The final description of the highest level nodes (in canonical English) is:
2.1 The BALL MOVES FROM the HAND.

2.7 The BALL MOVES WESTWARD and AGAINST the GROUND and OFF-OF the GROUND and UP-AND-DOWN.

Assuming the ball eventually stops bouncing, the repetition matching would be terminated by A.II.8: one of the UP or DOWN events would be kept "waiting" longer than the required time. Of course, other events may terminate B.7 earlier.

2.2.2 The man

It should be clear that the same process which allows the duration of an individual event to vary can easily allow for the shorter starting and stopping rotations (TO-AND-FRO) of the signals in Example 1. Similarly, the arm and leg movements of the walking man of Example 4 are accommodated as BACK-AND-FORTH repetitions. We will need to use rule A.II.6.2 there for certain null events when orientation changes become very small, but otherwise the example presents no problems. We assume that there is more data in succeeding (unillustrated) pictures since no actual repetition is shown in Figure 1.6.

We will describe the motion of the man and two of the visible parts, the right leg and the left leg. There would be nothing new in the events for the arms so we will not present them. These limbs show the use of BACK-AND-FORTH, since we recognize the RIGHT-SIDE view of the man and therefore know his FRONT direction. We will use frame numbers as if they were actual times.

The displacements of the man's head and body indicate that he is moving forward. Since we are assuming the sequence continues for the sake of the repetition description, the motion continues at least through frame 10.

<table>
<thead>
<tr>
<th>event M.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJECT MAN</td>
</tr>
<tr>
<td>AGENT MAN</td>
</tr>
<tr>
<td>VELOCITY (0, NIL)</td>
</tr>
<tr>
<td>ANGULAR-VELOCITY (0, NIL)</td>
</tr>
<tr>
<td>START-TIME 1.0</td>
</tr>
<tr>
<td>END-TIME 1.0</td>
</tr>
</tbody>
</table>
event M.2

SUBJECT MAN
AGENT MAN
VELOCITY (* + (STARTS)) (0.0, 90.0)
TRAJECTORY (* (0.0, 90.0) (0.0, 90.0))
DIRECTION (FORWARD)
ANGULAR-VELOCITY (* 0.0)
START-TIME 1.0
END-TIME (* 10.0 9.0)

Condensation DD.2 deletes M.1.
More is happening with his limbs. The AGENT and INSTRUMENT on the initial nodes are supplied by A.I.1.2.

event MFL.1

SUBJECT RIGHT-LEGG
AGENT MAN
INSTRUMENT RIGHT-HIP
VELOCITY (0.0, NIL)
ANGULAR-VELOCITY (0.0, NIL)
START-TIME 1.0
END-TIME 1.0

event MFL.2

SUBJECT RIGHT-LEG
AGENT MAN
INSTRUMENT RIGHT-HIP
VELOCITY (4.7 (STARTS))
ANGULAR-VELOCITY (0.0, 90.0)
DIRECTION ((CLOCKWISE BACKWARD))
START-TIME 1.0
END-TIME 4.0

event MFL.3

SUBJECT RIGHT-LEG
AGENT MAN
INSTRUMENT RIGHT-HIP
ANGULAR-VELOCITY (0.0, NIL)
START-TIME 4.0
END-TIME 5.0

event MFL.4

SUBJECT RIGHT-LEG
AGENT MAN
INSTRUMENT RIGHT-HIP
ANGULAR-VELOCITY (15.0 (STARTS))
DIRECTION ((CLOCKWISE FORWARD))
START-TIME 5.0
END-TIME 9.0

Condensation DD.2 causes the initial event MFL.1 to disappear, while MFL.4 absorbs MFL.3 by DD.1. The RATES adjustment removes STOPs from MFL.4. We now have:
event MPL.2

SUBJECT RIGHT-LEG
AGENT MAN
INSTRUMENT RIGHT-HIP
ANGULAR VELOCITY (4.7 (STARTS))
AXIS (0.0, 90.0)
DIRECTION ([CLOCKWISE BACKWARD])
START-TIME 4.5
END-TIME 4.5

event MPL.4

SUBJECT RIGHT-LEG
AGENT MAN
INSTRUMENT RIGHT-HIP
ANGULAR VELOCITY (15.0 NIL)
AXIS (180.0, 90.0)
DIRECTION ([COUNTERCLOCKWISE FORWARD])
START-TIME 4.5
END-TIME 9.0

At time 10.0 the leg begins to move BACKWARD again so BACK-AND-FORTH is invoked, creating the REPEAT-PATH node MPL.5. Changes to MPL.2 and MPL.4 are straightforward, while MPL.5 looks like:

event MPL.5

SUBJECT RIGHT-LEG
AGENT MAN
INSTRUMENT RIGHT-HIP
ANGULAR VELOCITY (* 7.0 15.0)
AXIS (* 0.0, 90.0) (180.0, 90.0)
DIRECTION (BACK-AND-FORTH)
REPEAT-PATH (MPL.2 MPL.4)
START-TIME 1.0
END-TIME (* 10.0 9.0)

The very same sort of structure evolves for the left leg:

event MIL.1

SUBJECT LEFT-LEG
AGENT MAN
INSTRUMENT LEFT-HIP
VELOCITY (0.0, NIL)
ANGULAR VELOCITY (0.0, NIL)
START-TIME 1.0
END-TIME 1.0

event MIL.2

SUBJECT LEFT-LEG
AGENT MAN
INSTRUMENT LEFT-HIP
ANGULAR VELOCITY (10.9 (STARTS))
AXIS (180.0, 90.0)
DIRECTION ([COUNTERCLOCKWISE FORWARD])
START-TIME 1.0
END-TIME 2.0
event MLL.3

SUBJECT LEFT-LEG
AGENT MAN
INSTRUMENT LEFT-HIP
ANGULAR VELOCITY (5.1 NIL)
AXIS (0.0, 0.0, 0.0)
DIRECTION ((CLOCKWISE BACKWARD))
START-TIME 2.0
END-TIME 7.0

---

event MLL.4

SUBJECT LEFT-LEG
AGENT MAN
INSTRUMENT LEFT-HIP
ANGULAR VELOCITY (0.0, NIL)
START-TIME 7.0
END-TIME 8.0

Condensation DD.1 deletes the null event MLL.1. By time 9.0 the leg begins to move FORWARD again so BACK-AND-FORTH is invoked, generating MLL.5 and terminating MLL.4. Since this makes MLL.4 null, DD.2 divides its time interval between MLL.3 and MLL.2 (which is now NEXT(MLL.4)). MLL.4 is deleted and the events for the left leg look like:

---

event MLL.5

SUBJECT LEFT-LEG
AGENT MAN
INSTRUMENT LEFT-HIP
ANGULAR VELOCITY (* 8.0)
AXIS (* (180.0, 0.0) NIL)
DIRECTION (BACK-AND-FORTH)
REPEAT-PATH (MLL.2 MLL.3)
START-TIME 1.0
END-TIME (* 9.0 8.0)

---

event MLL.2

SUBJECT LEFT-LEG
AGENT MAN
INSTRUMENT LEFT-HIP
ANGULAR VELOCITY (10.9 (STARTS))
AXIS (180.0, 90.0)
DIRECTION (COUNTERCLOCKWISE FORWARD)
START-TIME 8.0
END-TIME 10.0

---

event MLL.3

SUBJECT LEFT-LEG
AGENT MAN
INSTRUMENT LEFT-HIP
ANGULAR VELOCITY (5.1 NIL)
AXIS (0.0, 0.0, 0.0)
DIRECTION ((CLOCKWISE BACKWARD))
START-TIME 2.0
END-TIME 7.5
Comparing MRL.5 and MLL.5 we see that their lower level events are opposed pairs of BACKWARD and FORWARD movements as we should expect.

2.3. General repetitions

The general problem of finding repetitions in the input data is computationally more formidable though conceptually similar to the simpler cases we just described. A repetition is, after all, the realization that new events are already adequately described in the data base by past events. In general these past events may have an arbitrary pattern; for our three adverbials we were able to pre-specify the form as existence of a pair of primitive concepts. For other event structures, each new concept would have to initiate a search backwards in the event graph for matching events. If one is found then a tentative REPEAT-PATH is set up and new events compared with the old while the match succeeds. If it fails we must search back further in the graph for we may have selected the wrong starting point. (Suppose the sequence was a long, repeated string of characters.) Algorithms for finding repetitions in strings are easy to discover. For the multi-dimensional events we are using, the algorithm must be made more flexible (for example, Morris(1972) or, in more generality, Becker(1973)). In our methodology, the preferred approach would be to allow every "significant" event to be its own "demon," looking for repeated instances of itself as its expectation.

Although the syntactic part of repetition recognition is fairly straightforward, especially since we do not restrict the time periods to match, the matching criteria are crucial. The semantics can become very subjective. With BACK-AND-FORTH the semantics are well defined: distinctive orientation changes. But at higher levels, extracting the significant part of the sequence may be difficult even with unlimited time and space:

He hit the bag repeatedly.
She drives to work every morning.
History repeats itself.

All we can say is that it is very similar to a process of learning: we must determine what is or is not significant in
the data. Unfortunately, classical learning schemes offer little help as there is no teacher.

Returning to the adverbials, we have previously noted that AHEAD-OP and AFFECT are invoked when two objects traverse a similar path though separated in time. The path of one object is used as a pattern for that of the second; the semantics of the match focusing on the changing location relations rather than on contact or orientation change. It does not matter whether one man comes AFTER another walking backwards or hopping. It does matter that he follow the same path and thus experience the same location relations (to FIXED objects).

As in the general repetition case, we must be concerned with identifying the AFFECT or AHEAD-OP adverbial from arbitrary events. There is no path we can point to and say "look for this to be repeated by another object." Heuristics can be imagined for this situation, such as constant comparison of paths between different moving objects, but this seems to be too enthusiastic. After all, the people walking along a crowded sidewalk are not all AFTER or AHEAD-OP one another. Rather we would be more apt to use these concepts to compare two particular objects, selected for other reasons (perhaps during an interactive mode), or to relate one object of obvious importance (the observer) to others.

Like AFFECT and AHEAD-OP, TOGETHER and WITH look for repetitions in events between objects, but now there is no temporal separation. We must allow looseness in the path, however, since the local relations between adjacent objects may vary.

He was on one side of the seat, she on the other, though they drove to the movie together.

In fact, so many minor locational variations are possible that we should look only for adjacency of the two objects and similar trajectories. Despite the availability of this information, the most reliable source is the high level event data contained in AGENT and INSTRUMENT. Two SUBJECTs are combined when they have the same AGENTS and INSTRUMENTs, two AGENTS are combined when they have the same INSTRUMENTs and SUBJECTs, and similarly for the remaining case. For pairs of ANIMATE AGENTS, this will not
apply — the common AGENT does not exist. But then we may presume that a common path is not fortuitous and resort to matching trajectories.

We have already noted that WITH describes the relation between the objects in a list of AGENTS, INSTRUMENTS or SUBJECTs. The inverse of TOGETHER is APART, describing the mismatch of previously matched events.

The final high level adverbial is ALONG, a rather different concept than the previous. There is a repetition involved here, but one of a temporal structure against a static structure. The path of the SUBJECT must be repeated in a line or boundary of a proximal object. Here a trajectory direction is compared with an orientation direction for the match. Success instantiates ALONG on the DIRECTION list, a valuable anchor on transient paths to permanent structures in the world.

IV.3. Higher level descriptions

Throughout this chapter we have been discussing the construction of an event graph for a scenario of moving objects. Implicit in the examples were certain inferences made from the data to supply higher level concepts, for example, the AGENT GRAVITY property for the ball in Example 2. We will not be concerned with the inference mechanism itself, only the availability of the necessary knowledge. Additional descriptive power comes from the recognition of motion verbs from the event graph. We will use demons to recognize motion verbs and to solve certain mechanical problems.
3.1. **Motion verbs**

The final group of concepts necessary to arrive at realistic conceptual descriptions are the motion verbs. Appendix II contains a list of motion verbs compiled by Miller (1972), augmented with related verbs derived from this investigation. These share the same demon structure as the other concepts: all the verbs watch for events to match their individual case constraints. However, we are not supplying any intrinsic structure for the storage of verb names. These will be output as a description and only saved in the canonical form. (See, for example, Rumelhart et al. (1973) and Schank et al. (1973).)

We have anticipated this usage by providing the event representation with properties close enough to surface structures to make verb generation possible. What remains to be done is to list patterns for all the motion verbs the system can distinguish. We will not, primarily because the process is only one of giving case definitions of these verbs; constructing a lexicon of verbs giving admissible values for properties our representation allows. A verb describing an active event node could use the current static description of the scenario. The event node in isolation, however, will usually yield enough information since the adverbials retain the objects to which relationships have changed.

We do not claim that the system can distinguish all the verbs in Appendix II, since many of them depend on cases not covered by our representation (section III.3.5). Additional higher level knowledge about object capabilities should allow most of this set of motion verbs to be adequately defined.

3.1.1. **Sample motion verb definitions**

Going back to the discussion in section I.3, the definition of SWINGS requires a repetition adverbial TO-AND-FROM or BACK-AND-FORTH, while CLOCKWISE and COUNTERCLOCKWISE appear on the lower level repeated nodes. Clearly SWINGS applies to the man's leg movements in events MRL.5 and MLL.5. Thus, literally,
The MAN SWINGS his RIGHT-LEG [or LEFT-LEG] BACK-AND-FORTH.

The difference between SWINGS and WAVES would be the replacement of TO-AND-PRO or BACK-AND-FORTH by UP-AND-DOWN in the latter.

As mentioned in section 1.3 as well, the lack of rotational terms would imply the verb PFCIPROOCATES or perhaps OSCILLATES. If CONTACT relations were noted in one or both events by the presence of AGAINST, then BOUNCES would be more appropriate (see below).

As for the movement of the entire person, we consider a definition for WALKS. If the following conditions are satisfied then the event node F will be described by the concept WALKS.

WALKS
1. MOBILITY(AGENT(F))=ANIMATE.
2. VELOCITY(F) ≠ NIL.
3. SUBJECT(FL) is related to SUBJECT(F) via parallel CONNECTED-TO and PART-OF chains of edges. Likewise for SUBJECT(FR) and SUBJECT(F).
4. DIRECTION(FL) and DIRECTION(FR) include the adverbial BACK-AND-FORTH.
5. The interval when lower level nodes in one REPEAT-PATH contain BACKWARD overlaps the interval when lower level nodes in the other REPEAT-PATH contain FORWARD, or vice versa.

This is in fact the case with the event nodes for Example 4. There are some interesting things to notice about this definition. By (2) the person cannot walk in one spot. By (3) the person must have two moving parts joined with the whole. No specification of type is made, so this definition is also satisfied with TYPE(SUBJECT(FL))=HAND, for example. Certainly a person can walk on his hands. There is no contact restriction either, so it does not matter if he walks on stilts or even on air. The concept WALKS applies perfectly in these cases in English. By (4) the two parts must exhibit a certain kind of motion: BACK-AND-FORTH. The lower level event structure must exist before the verb is recognized. Moving the legs UP-AND-DOWN is not appropriate by itself. By (5) the BACK-AND-FORTH movements must be opposed, so as to distinguish WALKS from HOPS. Further distinctions, say WALKS from RUNS or MARCHES, would require additional conditions on the low level nodes. Just as the recognition of repetitions depended on learning the
significant events from the ongoing input stream, so too is the
definition of verbs a learning process.

It may often occur that several verbs apply to the same
event at the same time. In certain cases, the verbs will be
conceptually related on a hierarchy of motion verbs. We have
already shown that the most general concept is the composite
CHANGES-LOCATION-AND/OR-CHANGES-ORIENTATION. A description at
this level is good but hardly demonstrative of the capabilities
a verb lexicon with our case representation would provide. The
examples have used the generic concepts of MOVES and TURNS as
paraphrases. To improve this situation we should construct a
hierarchy of verb patterns, one containing another if the class
of concepts matched by the latter are always matched by the
former. However, from the observation of the patterns of many
motion verbs it is clear that the hierarchy is not very deep,
that is, few motion verbs are direct sub-concepts of other
motion verbs. They all seem to be rather specific, selecting
one case or another to quantify and characterize. As Miller
points out, the hierarchy that does arise is based upon the
particular cases a verb requires.

For the problem of several applicable verbs, the choice
will probably not be between different shades of meaning (if the
data is accurate enough, any movement distinctions should be
quantifiable), but between different case constructions.
Consider definitions for FALLS, DROPS, THROWS and BOUNCES:

FALLS
1. AGENT(F) = GRAVITY.
2. DIRECTION(F) includes DOWN but not UP.

DROPS
1. MOBILITY(AGENT(F)) = ANIMATE.
2. SUBJECT(F) has a part S related by SUB-PART chains
   such that DIRECTION(F) includes (FROM S), (OUT-OF S)
   or (OFF-OF S).
3. DIRECTION(F) or DIRECTION(NEXT(F)) includes DOWN
   but not UP.

THROWS
1. MOBILITY(AGENT(F)) = MOTILE.
2. (Same as DROPS 2.)
For example, the ball in Example 2 clearly left the hand in a downward movement. An ambiguity arises in attempting to establish whether or not the unseen person was the AGENT of the motion, that is, was the ball DROPPED or or did it FALL. Whether it was DROPPED or THROWN seems much less important. The patterns for these verbs indicate that DROPS and THROWS are appropriate to B.1. FALLS applies to B.3 but not to B.7 because of UP-AND-DOWN. BOUNCES does apply to B.7 because of AGAINST and OFF-OF. The description (following the application of the description demons in section IV.2.1) is therefore:

B.1 The BALL (DROPS/IS THROWN) FROM the HAND.
B.7 The BALL BOUNCES AGAINST the GROUND and OFF-OF the GROUND and UP-AND-DOWN.

Verb patterns may remove adverbials subsumed into the verb definition. For example, BOUNCES makes the specification of AGAINST and OFF-OF unnecessary (although the object GROUND would be lost). B.7 would then read:

B.7 The BALL BOUNCES UP-AND-DOWN.

We could only directly reconstruct that the ball bounced AGAINST and OFF-OF something. However, considerations at this level seem more appropriate to conceptual understanding rather than event construction.

For the car example we can note that any event with the car as AGENT can use DRIVES instead of MOVES. Moreover, DRIVES implies going FORWARD, otherwise we would say BACKS. TURNS is still appropriate. Moving TOWARD the OBSERVER is the same as APPROACHES. We could describe MOVES FROM as LEAVES. Applying these definitions to the event graph for the car (before DD.7 was applied) we obtain:

C.2 The CAR APPROACHES, then MOVES ONTO the ROAD.
C.3 The CAR [DRIVES], APPROACHES, then LEAVES the DRIVEWAY.
C.4 The CAR DRIVES AWAY-FROM the DRIVEWAY and TURNS AROUND the HOUSE, then DRIVES AWAY-FROM the HOUSE, then STOPS TURNING.
C.5 The CAR DRIVES AWAY.

Or, after condensation by DD.7:
The CAR APPROACHES, then MOVES ONTO the ROAD, then LEAVES the DRIVEWAY, then TURNS AROUND the HOUSE, then DRIVES AWAY-FROM the HOUSE, then STOPS TURNING, then DRIVES AWAY.

The major awkwardness with this last description is that it relates the car to every other object in the scene. Normally one object or another would be the focus of attention and statements would be made regarding its role. Such manipulations of the descriptions are yet unclear.

3.1.2. Comparison with human descriptions

Looking at the human descriptions of a similar scene (Appendix I), we find the popular use of the phrase PULLS OUT OF to describe leaving the driveway. PULLS emphasizes the agentive role of the car here, especially as no one has been observed DRIVING it. Similarly, the use of DOWN to indicate motion away from the observer can be considered special phrasiology associated with cars travelling on roads. Our description will still make sense but uses a stricter interpretation of the adverbials. Most of the other verbs used are quite generic: MOVES, COMES, TRAVELS and TURNS. The interesting comparison comes in the correspondence between our events and the division evident in the composite description. (The bracketed numbers refer to those descriptions in Appendix I.)

C.2 The CAR APPROACHES, then MOVES ONTO the ROAD.  |  (no comparison due to different location of observers.)

C.3 The CAR [DRIVES], APPROACHES, then LEAVES the DRIVEWAY.  |  [1] The car pulls out of the driveway.

C.4 The CAR DRIVES AWAY-FROM the DRIVEWAY and TURNS AROUND the HOUSE, then DRIVES AWAY-FROM the HOUSE, then STOPS TURNING.  |  [2] The car turns left.  

C.5 The CAR DRIVES AWAY.  |  (the car does not drive AWAY in our defined sense in Example I.)

The third described event [3] gives the relation of the car to the viewer in the film, rather than our event C.2. Likewise, the remainder of the composite description is irrelevant to the segment we have analyzed. Note that although these descriptions were generated from two different views of the same event, they
match in the essential points. Our description is, of course, more burdened with facts as we just pointed out. Some of the descriptions provided by humans are quite high level, especially since a short description was solicited.

3.2. Event prediction

We can use demons for the prediction as well as for the description of events. In this section we will suggest some inference rule demons (named Prediction Demon. #, or simply PD. #) which will allow the system to demonstrate its understanding of Example 3 by predicting the results of a certain movement of the man's hand. These demons are not complete in the sense that any diagram would be described correctly; we would need too much knowledge about mechanics and geometry for that. Instead they attempt to capture our intuitive notions about the actions of simple machines in stereotyped situations and perform reasoning at the event level. Nowhere will we appeal to low level information on rotations or translations, but this leaves the results more open to error in unusual configurations.

Contrast this to Baker's (1973) approach where the movement of a lever is simulated pictorially rather than descriptively, that is, in terms immediately accessible to conceptual descriptions. We would disagree with Baker that:

A possible explanation of the intuitive "seeing" of solutions to visual [diagram] problems is that closely associated with the human visual centres are highly-parallel processors; these processors would operate at very high speed in response to very simple commands, so that the solution is presented apparently instantaneously. (page 40)

Our own protocol of the pulley and lever system to be analyzed in section IV.3.2.2 follows the demon executions. The solution did not appear to us "instantaneously," but had to be analyzed sequentially, part by part. The "processors" and their "commands" seem better represented by the demons and their preconditions which are matched against the diagram data base or the hypothesized movement.

Significantly, our scheme apparently satisfies three of Slocan's (1971) statements regarding the description of mechanical diagrams, although Baker makes a similar claim.
Figure IV.10. Levers and a pulley (after Sloman)

Figure IV.11. Intermediate arrows
It is impossible for any situation to be represented by [Figure IV.10a] unless it is also represented by [Figure IV.10b]. ... (page 212)

The invocation of a demon from the data base assertions of other demons (according to section III.1.3) makes (b) follow "simultaneously" from (a).

Anyone who does not find this immediately obvious may be helped by being shown figures with arrows in intermediate positions, as in [Figure IV.11]. (page 213)

The demons, of course, insert such intermediate information in the data base as each is invoked. Moreover this answers the question posed in the final statement:

Problem: how do we know where to insert the intermediate arrows? (page 213)

The intermediate arrows correspond precisely to the intermediate directional relation (adverbial) assertions produced by executed demons. Let us follow several examples.

3.2.1. Pulleys

Figure IV.13 shows the data base information for a single pulley illustrated in Figure IV.12. We will assume that the recognition of the pulley configuration instantiates the appropriate CONTACT-type relations. The variable OUTPUT points to an object (not of TYPE ROPE) which contacts one part of the rope.

Let E(object-name) denote the current event node having the indicated object as its SUBJECT. The pulley rule is

**PD.1.** Given an arrangement like Figure IV.13, apply the motion

\[
\text{DIRECTION}(E(\text{rope-a-node})) := (\text{NOT-IN-THE-DIRECTION-OF pulley-node}).
\]

The result is

\[
\text{DIRECTION}(E(\text{rope-b-node})) := (\text{TOWARD pulley-node}),
\]

but more importantly:

1. If OUTPUT is of TYPE PULLEY then

\[
\text{DIRECTION}(E(\text{rope-b-node})) := (\text{AWAY-FROM OUTPUT}).
\]

2. If OUTPUT is not of TYPE PULLEY we must ascertain whether or not the OUTPUT object moves or the pulley moves. The heuristic is:

\[
\text{If OUTPUT is currently NOT-MOBILE and axle is CONNECTED-TO a MOBILE object then}
\]
Figure IV.12. Simple pulley

Figure IV.13. Simple pulley representation
else \[ \text{DIRECTION}(F(\text{pulley-node})) := (\text{TOWARD OUTPUT}) \]
\[ \text{DIRECTION}(F(\text{OUTPUT})) := (\text{TOWARD pulley-node}). \]

Admittedly, the problem of deciding the actual MOBILITY of the axle attachment and the OUTPUT object may be difficult, but we will use the simple heuristic that:

Any object CONNECTED-TO [or through a chain of such edges to] a FIXED object is NOT-MOBILE.

Note that the rotation of the pulley is ignored, and reasonably so, since the rope could just as well slip over the pulley.

Consider a simple case first (Figures IV.14 and IV.15).

Set
\[ \text{DIRECTION}(F(\text{ROPE-A})) := (\text{AWAY-FROM PULLEY}). \]

Since OUTPUT=CEILING (not a PULLEY) and is FIXED, we get
\[ \text{DIRECTION}(F(\text{PULLEY})) := (\text{TOWARD CEILING}). \]

This rule correctly handles the various arrangements in Figure IV.16, as well as extensions of these, assuming the axle attachment is appropriately fixed.

For compound pulley systems the events created by PD.1 cause successive applications of PD.1. The data base (Figure IV.18) for Example 3 (Figure IV.17) has such a situation. Suppose we assert

\[ \text{DIRECTION}(F(\text{ROPE-1})) := (\text{NOT-IN-THE-DIRECTION-OF PULLEY-1}). \]

By PD.1 we insert
\[ \text{DIRECTION}(F(\text{ROPE-2})) := (\text{TOWARD PULLEY-1}), \]

and since OUTPUT=PULLEY-2 deduce
\[ \text{DIRECTION}(F(\text{ROPE-2})) := (\text{TOWARD PULLEY-1}) \]

and
\[ \text{DIRECTION}(F(\text{ROPE-2})) := (\text{AWAY-FROM PULLEY-2}). \]

By straightforward hierarchic concept generalization
\[ \text{DIRECTION}(F(\text{ROPE-2})) := (\text{NOT-IN-THE-DIRECTION-OF PULLEY-2}). \]

Applying PD.1 again,
\[ \text{DIRECTION}(F(\text{ROPE-3})) := (\text{TOWARD PULLEY-2}). \]

Since OUTPUT=CEILING and the ceiling is CONNECTED-TO-RIGID (via a chain of such edges) to the FIXED ground, we get
\[ \text{DIRECTION}(F(\text{PULLEY-2})) := (\text{TOWARD CEILING}). \]
Figure IV.14. Another pulley system

Figure IV.15. Representation of Figure IV.14
3.2.2. The mechanical diagram

To start at the beginning of the analysis of Example 3 (Figure IV.17), suppose we are told to assume

\[
\text{DIRECTION}(E(\text{HAND})) := (\text{TOWARD FLOOR}).
\]

If one end is pulled AWAY-FROM the other end, it eventually follows:

\[
\begin{align*}
\text{PD. 2.} & \quad \text{if } \text{GUIDED-BY}(\text{rope-node object-1-node}) \quad \text{and} \\
& \quad \text{GUIDED-BY}(\text{rope-node object-2-node}) \quad \text{and} \\
& \quad (\text{NOT-IN-THE-DIRECTION-OF object-2-node}) \\
& \quad \text{then conclude} \\
& \quad \text{DIRECTION}(E(\text{object-1-node})) := \\
& \quad (\text{NOT-IN-THE-DIRECTION-OF object-2-node}).
\end{align*}
\]

This is a special case of a general rule regarding "pulled" objects. It does not take into account local rotational affects.
that might occur. The non-rigidity of the rope is incorporated into the directional constraint.

A class of rules dictates the relationship between locative relations and directional relations. Others are possible, but we will need:

PD-3. Let \texttt{rel} \in \texttt{SUPPORT-PLANE U ELEVATION} relations. If \texttt{rel}(object-a-node object-b-node) and \texttt{rel}(object-b-node object-c-node) then \texttt{DIRECTION} (P(object-b-node)) := (TOWARD object-a-node) implies \texttt{DIRECTION} (P(object-b-node)) := (AWAY-FROM object-c-node)
and vice versa. In addition if \texttt{rel} = \texttt{ABOVE} or \texttt{BELOW}, then
\[
\text{\texttt{DIRECTION}}(F(\texttt{object-b-node})) := (\texttt{UP}) \text{ or } (\texttt{DOWN})
\]
respectively.

Since \texttt{ABOVE} (\texttt{PULLEY-1 \textsc{hand}}) and \texttt{ABOVE} (\texttt{HAND FLOOR}) hold in
the static description, \texttt{DIRECTION} (\texttt{E(HAND)}) = (\texttt{TOWARD FLOOR}) and
\texttt{PD.3} imply
\[
\text{\texttt{DIRECTION}}(F(\texttt{HAND})) := (\texttt{AWAY-FROM PULLEY-1})
\]
and an easy generalization gives
\[
\text{\texttt{DIRECTION}}(F(\texttt{HAND})) := (\texttt{NOT-IN-THE-DIRECTION-OF PULLEY-1})
\]
Now \texttt{PD.2} applies, yielding
\[
\text{\texttt{DIRECTION}}(F(\texttt{ROPE-1})) := (\texttt{NOT-IN-THE-DIRECTION-OF PULLEY-1}).
\]
The pulley rule \texttt{PD.1} is now applicable, as previously noted.

A very liberal heuristic for \texttt{CONNECTED-TO} relations
describes the transmission of the movement.

\texttt{PD.4: If CONNECTED-TO (object-1-node object-2-node) and \texttt{DIRECTION}}(F(\texttt{object-1-node})) includes some of
\{\texttt{UP, DOWN, TOWARD, AWAY-FROM, IN-THE-DIRECTION-OF}\}

then, assert the same for \texttt{DIRECTION} (\texttt{E(object-2-node)}),
provided object-2 does not connect to other objects in
such a way that it is \texttt{NOT-MOBILE}.

The looseness comes from the assumption that the joints do not
rotate significantly.

From the result of the pulleys,
\[
\text{\texttt{DIRECTION}}(F(\texttt{PULLEY-2})) := (\texttt{TOWARD CEILING})
\]
and applying \texttt{PD.4} three times, we get

\[
\text{\texttt{DIRECTION}}(F(\texttt{CONTACT-POINT})) := (\texttt{TOWARD-CEILING}).
\]

To finish this example we must predict what happens to the
lever. There are three rules, one for each class of levers, but
they all have the same basic pattern (Figure IV.19). The
variable \texttt{PIVOT} will point to the \texttt{CONNECTED-TO-POINT}, \texttt{CONNECTED-TO-LINE} or \texttt{SUPPORTED-BY} lever subpart. The relation \texttt{rel} is from
the set of \texttt{SUPPORT-PLANE} and \texttt{ELEVATION} relations. The \texttt{PIVOT}
part is assumed to be the axis of the lever's rotation, so the
lever rule is:

\texttt{PD.5: (1) If PIVOT=CONTACT-POINT-1 or -3:}
CONTACT-POINT-1  CONTACT-POINT-2  CONTACT-POINT-3

CONNECTED-TO-RIGID  rel

CONNECTED-TO-POINT
or
CONNECTED-TO-LINE
or, if neither of these,
SUPPORTED-BY

(Only one such edge of the three shown must be present.)

rel is any SUPPORT-PLANE or ELEVATION relation.

Figure IV.19. Lever patterns
or vice versa.

(2) If PIVOT=CONTACT-POINT-2:

\[
\text{DIRECTION}(E(\text{CONTACT-POINT-1})) := \text{opposite DIRECTION}(E(\text{CONTACT-POINT-3}))
\]

or vice versa.

In either case the DIRECTION relations used come from the set

\{UP, DOWN, TOWARD, AWAY-FROM, IN-THE-DIRECTION-OF\}.

Since we presently have

\[
\text{DIRECTION}(E(\text{CONTACT-POINT})) := (\text{TOWARD CEILING}),
\]

rule PD.5 applies to produce

\[
\text{DIRECTION}(E(\text{LEFT-END})) := (\text{TOWARD CEILING}).
\]

By a rule PD.6 analogous to PD.4 which asserts that an object SUPPORTED-BY another single object shares its DIRECTION, we obtain from SUPPORTED-BY (BLOCK LEFT-END)

\[
\text{DIRECTION}(E(\text{BLOCK})) := ((\text{UP (OFF-OF FLOOR) (TOWARD CEILING)})).
\]

From PD.3, (TOWARD CEILING) implies (AWAY-FROM FLOOR) for the block. Therefore the SUPPORTED-BY relation with the floor is broken (since the floor is not moving), triggering the UP adverbial because of ABOVE(CEILING FLOOR). OFF-OF comes from the breaking of the relation itself.

At this point the scenario has been analyzed correctly from the given hypothetical movement. By applying DD.3 to condense the SUBJECT-Instrument chain (we assume that all the nodes are terminated), we get:

\[
\begin{align*}
\text{AGENT PERSON} \\
\text{INSTRUMENT (HAND ROPE-1 ROPE-2 ROPE-3 PULLEY-2 AXLE-2 SUPPORT-2 CONTACT-POINT LEFT-END))} \\
\text{SUBJECT BLOCK} \\
\text{DIRECTION ((UP (OFF-OF FLOOR) (TOWARD CEILING)))}
\end{align*}
\]

Paraphrased, this might be:

The PERSON MOVES his HAND, MOVING the ROPE-1, MOVING the ROPE-2, ..., MOVING the BLOCK UP and OFF-OF the FLOOR and TOWARD the CEILING.

The reasoning embodied in the rules seems to capture the essence of the mental reasoning we might use to solve the problem. We make several assumptions about the mobility of the various parts in the configuration, but we would be surprised if things went differently. For example, if the application of
PD.5 had instead selected the left end of the board as the pivot, it would have predicted the right end going towards the ceiling. This is a plausible occurrence if we do not assume that CONNECTED-TO-POINT(RIGHT-END TRIANGLE) held, but only SUPPORTED-BY. The system also lacks any information on the relative strength and weight of the various parts so that snapping ropes or bending levers are not considered.

3.2.1. Some other simple machines

While on the subject of machines, it is interesting to note that the seemingly complex movement of a bolt and a nut are describable by the same techniques. Just to illustrate, suppose we know both the bolt and the nut have the same thread bandedness, say right, and the bolt is currently stationary. If SURROUNDED-BY(BOLT NUT) and if

\[
\text{if } \text{SURROUNDED-BY}(\text{BOLT NUT}) \text{ and if}
\]

\[
\text{then } \text{DIRECTION}(\text{F(NUT)}) \equiv \text{CLOCKWISE}
\]

\[
\text{else if } \text{DIRECTION}(\text{F(NUT)}) \equiv \text{(ONTO BOLT) CLOCKWISE)}
\]

\[
\text{then } \text{DIRECTION}(\text{F(NUT)}) \equiv \text{(COUNTERCLOCKWISE)}
\]

\[
\text{DIRECTION}(\text{F(NUT)}) \equiv \text{(OFF-OF BOLT) COUNTERCLOCKWISE)}.
\]

If the nut is stationary the terms change slightly. If

\[
\text{if } \text{DIRECTION}(\text{F(BOLT)}) \equiv \text{CLOCKWISE}
\]

\[
\text{then } \text{DIRECTION}(\text{F(BOLT)}) \equiv \text{(INTO NUT) CLOCKWISE)}
\]

\[
\text{else if } \text{DIRECTION}(\text{F(BOLT)}) \equiv \text{(OUT-OF NUT) COUNTERCLOCKWISE)}.
\]

It is a simple but tedious exercise to derive all possible event relations for non-stationary bolts and nuts and for differing rotational velocities if lower level information can be taken into account. Likewise, gears are a simple matter; we can even use AXIS and ANGULAR VELOCITY coupled with SIZE as a radius to compute the actual direction and velocity of the transmitted motion.
Chapter V. Recognition of fundamental concepts

The first part of this chapter continues a presentation begun in section II.1 of a coordinate system for two-dimensional images of three-dimensional space as projected onto a spherical "eye." As a model of idealized wide angle vision it is not tied to a specific camera configuration but is a theoretical approach to the perception of events in space. Properties of the model are discussed in terms of algorithms for plane slant of surfaces, movement of the observer, trajectories and rotation of objects. A unified treatment of certain spatial relations also results from the model, even when size information on objects is lacking.

The second part discusses a methodology for the recognition of objects using the object model graph. The method involves top-down, model-driven analysis to efficiently process the sequence of static images.

V.1. Properties of objects and events

Recall the definitions of the mappings E, S and F, the sphere S, and the successive frames Im(V,t) from a viewpoint V at time t described in section II.1. There we presented the relationship between points in space and a spherical projection of those points. Here we will continue the investigation of the projective properties of this two-dimensional image.

In the following sections the only additional assumption made about the visual system is that it has a sufficiently fine sampling grid so that numerical errors in calculations are, for theoretical concern, of little consequence. Practical considerations will be mentioned where applicable or necessary. In addition, although Im is defined everywhere it need not be observed all at once.

Since there can be no straight lines contained in the image sphere's surface S, all three-dimensional curves must project into spherical curves. This is not so bad as it might seem. The most interesting of these possible curves are great circles, the intersection G with S of a plane C through V. A line,
curve, closed curve, or set of any of these lying entirely within C will project into (a set of) arcs of the great circle G. These arcs have no apparent curvature to the observer at V; the image "looks like" sets of collinear straight line segments. We would not expect this to be very informative because of the high degree of ambiguity involved in the inverse perspective transformation.

With the spherical system in standard observer orientation, vertical lines are distinguished by the simple constraint that the $\theta$ coordinate of all its points is a constant. Given two successive observer positions (slightly separated) the verticality of a line (sufficiently close to the observer) may be easily determined if in each of the two images the endpoints of the line satisfy the above condition. It is not true that constant $\phi$ indicates horizontality in the same manner.

Any set of parallel lines in space has a useful projective image. (See Figure V.1.) For such a set \{k(i)\} there is one and only one line N such that N passes through V. The planes $P(i)$ containing $k(i)$ and passing through V must intersect along N. This means that the family of great circles which are the intersections of these $P(i)$ with S intersect in the two points common to N and S. Since N is parallel to each of the $k(i)$, the spherical coordinates of the intersections \{N', N\} give precisely the angular orientation of the $k(i)$ relative to the coordinate system. In planar projections one of the points $N'$ or $N$ would be called the vanishing point; the other would have no projective correlate since it would lie behind the focal plane. The situation in the spherical projection is different since both points have real coordinates.

We will use the intersection of great circle arcs to compute object orientations and trajectories as well as observer movement. But before proceeding further it should be noted that there is nothing intrinsically complex about the computation of the intersection of two great circle arcs. The following algorithm takes two pairs of spherical coordinate points

\[
\text{\textit{v}}_0 = (\theta(1), \phi(1))^0, \quad \text{\textit{v}}_1 = (\theta(2), \phi(2))^0
\]

\[
\phi_0 = (\theta(3), \phi(3))^0, \quad \phi_1 = (\theta(4), \phi(4))^0
\]
and computes $N^+$ and $N^-$.

1. Convert the spherical coordinates to Euclidean coordinates:

   $$(x(i), y(i), z(i)) = \mathbb{P}(\theta(i), \phi(i), 1)^0 \quad \text{for } i=1,2,3,4.\)$$

2. The normal to the plane through the origin and containing $(x(1), y(1), z(1))$ and $(x(2), y(2), z(2))$ is found from the cross product of those two vectors:

   $$(a,b,c) = \left( (y(1)z(2) - z(1)y(2)), (x(2)z(1) - z(2)x(1)), (x(1)y(2) - y(1)x(2)) \right).$$

   Similarly, for $(e,f,g)$ as the cross product of $(x(3), y(3), z(3))$ and $(x(4), y(4), z(4))$. (In general the normals $(a,b,c)$ and $(e,f,g)$ are not unit vectors.)

3. The intersection of these two planes is the cross product of their normals, that is, the line through $(0,0,0)$ and $N=(bg-ce, ce-ag, af-be)$. (In general $N$ does not have unit length.)

4. Finding an intersection of the line containing $N$ with $S$ is equivalent to finding the spherical coordinates of $N$ projected onto $S$:

   $$N^+ = (\theta^+, \phi^+)^0 = \mathbb{P}(N).$$

   Then $N^+$ is one solution point while $N^- = (\theta^+, n \mod \pi, n - \phi^+)^0$ is the diametrically opposite solution point.
5. The solution point in the direction of $P_0$ to $P_1$ and of $G_0$ to $G_1$ is simply the one of $\{N^+, N^-\}$ closest in arclength to $P_1$ (and therefore to $G_1$ as well).

Costwise, if $f$ represents a call to a mathematical function (sine, cosine, arctangent, arccosine, or square root) and $m$ a multiplication or division, the algorithm uses

$$10f + 31m$$

operations per two pairs of points, a constant.

In order to minimize numerical errors each of the points $(\Theta(i), \Phi(i))^0$, $i=1, 2, 3, 4$, should be well separated from the other three and the configuration of all four should enclose some non-trivial spherical surface area. If either of the pairs contain two identical points step (2) will yield a zero length normal vector $(a, b, c)$ or $(e, f, g)$. If the two pairs are effectively coplanar step (3) will yield a zero length normal vector $N$. The algorithm itself can therefore detect these degenerate cases and respond accordingly.

If the two pairs generate paths that intersect somewhere between the endpoints then a spurious normal in the direction of the intersection will be generated. To the algorithm the pair has "gone to infinity" and then returned (see Figure V.2). This can be avoided by an appropriate line intersection test. Note that the spherical coordinates of these points define two lines in a plane, since the intersection point $N^0$ will still lie on the same normal vector. Another alternative is to guide point pair selection by high level considerations of orientation and trajectory.

Given more than two great circles the computation of the intersection should proceed pair by pair. In general this can lead to difficulties in specifying what the actual intersection point is, since numerical errors will insure that even quite precise data will yield a variation in computed values. A simple mean value for the set usually gives a satisfactory result.
1.1. Plane-slat

Suppose a visible face of an object is planar and contains a uniform rectangular grid of feature points. The angular orientation of this face with respect to the spherical coordinate system can be determined by finding the spherical intersection points of the great circle arcs generated by the grid of features. For a uniform rectangular grid, the arc families in two different directions suffice to determine the orientation in two orthogonal directions. The normal to the orthogonal pair is the orientation direction; it is easily found as the cross product of the two orientation vectors. If the face grid is not rectangular the computation will still make sense but will yield the wrong slant (perspective illusion).

If the face features do not form a grid (or an unknown grid angle) but there are rather large numbers of them and they have a statistically uniform distribution over the face, information can be extracted from the texture. The lines of maximum textural density change, that is, the directions of the density gradient, can provide one family of arcs while the lines of minimum gradient can provide the other. These two sets of lines are orthogonal on the object surface. The computation of the face orientation then proceeds as before. A planar projective case has been described by Flock (1964); this scheme subsumes
his. A low level texture gradient operator is used in Bajcsy and Lieberman (1974). The psychological literature shows the general usefulness of gradient analysis, see for example Gibson (1950).

Without appreciable or regular visual texture but with planar faces, face orientation depends only on edge or vertex features. Without knowledge of the expected angles and relationships of the various features it is unlikely that any face orientations can be confidently computed. With knowledge based on expected forms, quite good results may be had. This situation is the normal paradigm for computer vision (robot) research projects, where polyhedral or geometrically simple objects abound. For example, rectangular faces have orthogonal feature points, so that face orientation may be easily computed.

The orientations of the two visible faces of the house in Figure IV.2 are determined by this slant algorithm. The results appear in Table V.1 and have been normalized so that $0 \leq \theta \leq 180^\circ$.

<table>
<thead>
<tr>
<th>Table V.1.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FRONT WALL</strong></td>
</tr>
<tr>
<td>left edge</td>
</tr>
<tr>
<td>right edge</td>
</tr>
<tr>
<td>top edge</td>
</tr>
<tr>
<td>bottom edge</td>
</tr>
<tr>
<td>orientation</td>
</tr>
<tr>
<td>right-left</td>
</tr>
<tr>
<td>top-bottom</td>
</tr>
<tr>
<td>normal direction</td>
</tr>
</tbody>
</table>

| **ROOF** |
| left edge   | (1.62, 1.36) | (1.63, 1.45) |
| right edge  | (0.98, 1.39) | (0.85, 1.48) |
| top edge    | (1.62, 1.36) | (0.98, 1.39) |
| bottom edge | (1.63, 1.45) | (0.85, 1.48) |
| orientation | (90.8°, 61.4°) |
| right-left  | (175.0°, 89.3°) |
| top-bottom  | (85.9°, 151.0°) |

The computed orientations of the house wall and roof are within 5% of the actual values. The slope of the roof was designed to be 150°. Although three significant digits were used for
picture coordinates, the actual range (approximately 1. to 2.) makes the data accuracy only about 1%.

1.2 Observer movement

The observer is an object in the world and has an object node showing its relationships to other objects in the scene. It is always NOT-VISIBLE, has no SUB-PARTS, and has ORIENTATION values fixed by the initial assumptions for the interpretation of spherical coordinates given in section II.1. In addition, it may be given an eye height in SIZE. The observer is a passive viewer if MOBILE, or an autonomous robot if MOTILE.

An event node may have the observer as a SUBJECT if he moves. All the properties of an event apply, in particular TRAJECTORY, AXIS, and DIRECTION. The first two are determined from the successive images of (FIXED) objects in the scenario.

There are six parameters of observer movement and each parameter describes the relationship of the observer's spherical vision system reference frame to the three-dimensional Euclidean coordinate frame of the environment. The first three are translations that relate the origin of the sphere S to the origin of the Euclidean system O. The other three relate to the orientation of the spherical coordinates: pan is rotation about the ray \((0,0,1)\)°, UP; tilt is rotation about the ray \((\pi/2,\pi/2,1)\)°, the LEFT direction; and rotate is rotation about the ray \((0,\pi/2,1)\)°, the FRONT. We could derive the appropriate transformation equations for new spherical coordinates given the old, similar to the expressions derived in Duda and Hart (1973) for planar systems, but another approach will be taken here to exploit the spherical model.

Pan, tilt and rotate coordinate transforms are unnecessary since all spatial points retain their real image coordinates if the viewpoint is not moved. We can leave actual system imaging problems to further mathematical techniques [Sobel (1973)]. But if we use a hemispherical view of the world where 0 ≤ z ≤ π, any object within visual range can be observed simply by panning, that is, by rotation about the vertical axis until the desired scene is obtained. The restriction to panning yields a
computational reward. Suppose a pan of \( \varphi \) radians is executed, then the new spherical coordinates of the old image point \((\theta^*, \varphi^*)^o\) become simply \((\theta^*-\varphi \mod n, \varphi^*)^o\). In a plane projection the simplest transformation is rotate (one 2x2 matrix multiplication), but rotate is the least useful vision capability and panning requires the full three-dimensional rotation equation. It follows that observer rotation creates at worst a linear transformation \((\theta-\varphi, \varphi)^o\) of the entire image and therefore rotation cannot change any of the relative relations between objects and their images. If the observer is able to correlate fixed points in successive (panned) images then there will be no \( \theta \) transformation at all.

For observer translation along some path it is easy to show that the new visual position of a given point is mathematically dependent on the distance between observation points and the distance from the observer to the actual point. If both these quantities are known then the \((x, y, z)\) position converts readily into spherical coordinates from the new spherical center. Normally, at least one of the two crucial distances will be unknown. If a point has not moved in space (perhaps it is a feature related to a FIXED object by a CONNECTED-TO-RIGID relation) then its new image point must lie on a great circle arc passing through the old image point and the translation heading. (The heading is easily obtained by the trajectory algorithm given in section V.1.3.1, provided a pair of FIXED feature points are known.) This provides a test for whether a new image point can possibly be related to the old. (See Figure V.3.) If \( H=(\theta^*, \varphi^*)^o \) is the translation heading and \( Q(1)=(\theta(1), \varphi(1))^o \) was the last observed image of the point \( Q=(u, v, w) \), then in the new observer position \( Q \) must still be on the plane containing \( V, H, \) and \( Q(1) \):

\[
N \cdot (x, y, z) = 0
\]

where "\( \cdot \)" denotes vector dot product and

\[
N = \{(y*z(1)-z*y(1)), (x(1)*z-x(1)*x), (x*y-y*x(1))\},
\]

and

\[
(x^*, y^*, z^*) = \mathcal{E}((\theta^*, \varphi^*, 1)^o)
\]

\[
(x(1), y(1), z(1)) = \mathcal{E}((\theta(1), \varphi(1), 1)^o)
\].
Therefore the image point $Q^{(2)} = (\theta^{(2)}, \varphi^{(2)})^0$ is on this great circle arc if and only if

$$0 = \mathbf{N} \cdot (\cos \theta^{(2)} \sin \varphi^{(2)}, \sin \theta^{(2)} \sin \varphi^{(2)}, \cos \varphi^{(2)}).$$

Thus without knowing the actual distance from the observer to an object feature nor the distance moved by the observer, a constraint may be placed on the expected new image location of the feature. The constraint may be used with the beginning movement where no prior velocity is available. In section V.1.3.1 we will give an algorithm for actually determining the new position when the observer or object trajectory is known and the velocity is assumed constant.

For the most part, observer movement complicates the frame-to-frame picture correlation but does not significantly affect the description of the scene. Because motion descriptions are based on locative relation changes, events for other objects will be nearly independent of the observer's own motion events.
1.3. Primitive motions

Rigid objects in space have two primitive motions available to them: a move from one point to another so that object features travel in parallel paths, or a rotation about an axis. Both the path and the axis may vary continuously, but they are higher level temporal concatenations and/or superimpositions of the primitive motions. The next two sections examine the determination of three-dimensional trajectories and rotations from object feature displacements in the image sequence.

1.3.1. Trajectory

Let $O\text{b}$ be a rigid object with a set of feature points $\{F(i)\}$, $i=1, \ldots, n(\text{Ob})$. Assume that object recognition demons have associated the projected points of the same feature in successive temporal images, $\text{Im}(V, t^{(1)})$ and $\text{Im}(V, t^{(2)})$.

Now suppose $O\text{b}$ maintains a constant orientation (with respect to the environment) and translates in an arbitrary linear trajectory during the time interval $t^{(2)}-t^{(1)}$. If $P_{S}(F^{(1)}(i))$ were the spherical coordinates (if visible) of the features $F(i)$ at time $t^{(1)}$ and if $P_{S}(F^{(2)}(i))$ were those of the $F(i)$ at time $t^{(2)}$, then there is a set of pairs $\{(P_{S}(F^{(1)}(i)), P_{S}(F^{(2)}(i)))\}$ giving the successive image coordinates of each (totally visible) feature point. Since each feature point describes a linear trajectory and since object rigidity forces all these feature trajectories to be parallel, the set of image paths of feature points generates a family of great circle arcs which intersect the projection sphere at the direction coordinates of the trajectory. Thus the direction of movement of a linearly translating object is found by applying the algorithm in the beginning of section 5.1 to all feature point pairs and averaging the results.

In practice the computation of the intersection of the whole family is too costly and prone to numerical errors, however two feature points nearly orthogonal to the trajectory are sufficient to determine two maximally angularly separated arcs. This is not as circular as it sounds because the
movements are tracked by higher-level descriptions of expected motion. Thus only when an expected motion fails to materialize or when unexpected movement occurs will "blind" computation of the trajectory be necessary. Even changes in trajectory are not likely to involve drastic angular changes (except during collision), and those feature point pairs orthogonal before the change are likely to be nearly so afterwards, at least long enough to select a new pair from higher level considerations. In fact, if most observed trajectories occur in planes roughly parallel to the ground, then feature pairs on vertical lines will usually satisfy the orthogonality criterion.

An illusion would arise if objects were not fixed in size and allowed to expand or contract. Fortunately the effect is rare in natural environments except for an occasional balloon. Without other information an expanding object will be perceived as approaching the observer and a contracting object as receding. Perceptual expansion or contraction of a rigid object need not be considered a special type of perceived movement as suggested by Gibson (1957). Translations across the field of view as well as towards or away from the observer are determined by identical computations.

The two most important sources of error in the trajectory computation come from point separations or temporal displacements which are on the order of magnitude of the visual resolution of the system. For the first case the object itself may be too small to yield a reasonably separated point pair. The bouncing ball of Example 2 falls in this class. The solution is to consider the mean displacement of the object in \((\theta, \phi)\) coordinates as if it were a spatial trajectory. Such an object appears to have only two-dimensional motion. Indeed, from Figure 7.3 it is difficult to ascertain the heading of the ball except for left, down, and up.

The second case arises when the frame-to-frame displacement of a perhaps well-separated pair of points is slight. The translation of the man in Example 4 nearly falls into this class. There is an exact analogy with the problem of line-fitting discrete data points in a single image: a temporal
sequence of points requires an arc fit to ascertain when a new point departs significantly from the current path. An algorithm for line-fitting by Brice and Pennema (1970) will work in this situation because points are processed sequentially, that is, temporally. The sliding mask can fit a straight line rather than an arc since small angular displacements look like straight lines anyway. By the time they do not, the cumulative displacements will have been quite sufficient for the trajectory calculation. These displacements will only be needed at this very low level and can be queued in the IMAGE property of an object.

If the object is known to be SUPPORTED-BY or GUIDED-BY a horizontal object then the pair of points is no longer necessary and a single point displacement will suffice. The other point is assumed to travel in the plane and the normal to its movement is just (0, 0, 1). In general, if the normal is known, then movement in a plane of any orientation is best described by this device. If these assumptions hold the results will be more accurate since one cross product is not computed. Compare the trajectories of the car in Figure IV.2, where the support is assumed, with the same sequence without this assumption [Badler (1974)]. The difference is most apparent when the displacements are small (for example, between times 3.00 and 4.00).

The trajectory serves in an active as well as a passive role. Consider a non-rotating object Ob moving along a linear trajectory with heading (θ⁺, Φ⁺). Let A be a point of Ob and let A(1), A(2), and A(3) be the actual locations of A during observation times t(1), t(2), and t(3), respectively. Knowing the locations of the images PS(A(1)) and PS(A(2)) of the point A, the image location PS(A(3)) may be predicted under assumptions of constant velocity and adherence to the same trajectory. No knowledge of the three-dimensional positions of A(1), A(2), and A(3) is necessary, not even of distances between the points. The following procedure will show how to do this. (Refer to Figure V.4.) The goal is to compute the expected projection point PS(A(3)).
Since the observed positions $PS(A^{(1)})$ and $PS(A^{(2)})$ make the angular displacement $a(1)$ computable and since the known heading $H$ makes the angle $b(1)$ computable, the angle $c(1)$ is easily found as $\pi - a(1) - b(1)$. Letting
$$d = ||A^{(2)} - A^{(1)}||$$
the distance
$$||A^{(3)} - A^{(1)}||$$
can be expressed as a multiple of $d$, say $kd$ for some constant $k$. With constant velocity and equal time sampling intervals $k=2$. In general the distance
\[(k - 1)\delta = \|A(3) - A(2)\|\]
can be directly related to the ratio of the sampling intervals
\[i_{1-2} = t(2) - t(1)\]
and
\[i_{2-3} = t(3) - t(2)\]
namely,
\[
\frac{i_{1-2}}{\delta} = \frac{i_{2-3}}{(k-1)\delta}
\]
that is,
\[k - 1 = \frac{i_{2-3}}{i_{1-2}}. \quad \text{(Eqn. 1)}\]

From the law of sines in triangle \(VA(1)A(2)\) we get
\[
\frac{\sin a(1)}{\delta} = \frac{\sin c(1)}{w(1)}
\]
and in triangle \(VA(1)A(3)\),
\[
\frac{\sin a}{k\delta} = \frac{\sin c(2)}{w(1)}. \quad \text{(Eqn. 2)}
\]
Rearranging these gives
\[
\frac{\sin c(1)}{\sin a(1)} = \frac{w(1)}{\delta} = \frac{k \sin c(2)}{\sin a}
\]
An easy substitution \(c(2) = \pi - a - b(1)\) gives an equation which can be solved for \(a\):
\[
\sin a = \frac{k \sin a(1)}{\sin c(1)} \sin(a + b(1))
\]
After some work the solution is found to be
\[a = \arcsin\left(\frac{m}{\sqrt{n^2 + m^2}}\right)\]
where
\[m = k \sin a(1) \sin b(1)\]
and
\[n = \sin c(1) - k \sin a(1) \cos b(1)\]
By inspection of the figure, the angle \(a\) can never exceed \(\pi\), hence only the positive solution for \(a\) is necessary.

The expression for the denominator is defined unless it is 0; solving for \(\sin a(1)\) (or \(\sin c(1)\), since it is symmetric in these terms) gives, after simplification,
\[\sin a(1) = \sin c(1)[\cos b(1) \pm \sqrt{\cos^2 b(1) - 1}] / k\]
This is 0 if and only if \(\sqrt{\cos^2 b(1) - 1}\) is real, that is, \(\cos^2 b(1) = 1\). This implies \(\cos b(1) = \pm 1\) which implies \(b(1) \equiv 0 \pmod{\pi}\), or else if \(\sin c(1) = 0\) then \(c(1) \equiv 0 \pmod{\pi}\). But then either of these force \(a(1) = 0\); hence \(A(1), A(2)\) and \(V\) are collinear. In this case the prediction is simply
\[PS(A(3)) = PS(A(2))\]
and the remainder of the algorithm below may be skipped.

The position \( PS(A^{(3)}) \) must now be computed. Notice that the whole point of this algorithm is that the actual distances \( w(1), w(2), w(3), \) and \( d \) are irrelevant, only their ratios are important. Thus any arbitrary assignment of length to one will force values on the other three. Since \( PS(A^{(3)}) \) is the point of interest, it would be convenient if \( w(3) = 1 \). Then values for \( w(1), w(2), \) and \( d \) would provide three-dimensional coordinates for "pseudo-points" \( A'^{(1)} \) and \( A'^{(2)} \). From these a simple vector formula for a line determines the pseudo-point \( A'^{(3)} \) and by choice of \( w(3) = 1 \), \( A'^{(3)} \) lies precisely in \( S \) at \( PS(A^{(3)}) \).

To see this let

\[
\text{Let } w(3) = 1
\]

so that the law of sines gives

\[
w(1) = \frac{\sin c(2)}{\sin b(1)}
\]

and

\[
w(2) = \frac{\sin c(2)}{\sin c(1)}
\]

Let

\[
A'^{(1)} = w(1) PS(A^{(1)})
\]

and

\[
A'^{(2)} = w(2) PS(A^{(2)})
\]

Remember that \( PS(A^{(1)}) \) and \( PS(A^{(2)}) \) are known visual coordinates so only \( \Xi \) need be computed. Then

\[
A'^{(3)} = A'^{(1)} + k(A'^{(2)} - A'^{(1)})
\]

The point \( A'^{(3)} \) should now satisfy

\[
||A'^{(3)}|| = 1
\]

hence

\[
PS(A^{(3)}) = PS(A'^{(3)})
\]

The last equation is clearly a parametrization by \( k \) of the image of a great circle arc defined by two distinct points.

One perhaps useful side effect of this algorithm is that the point of closest approach may be estimated. In Figure \( V.4., A* \) indicates the point where the trajectory of \( A \) is perpendicular to a vision line from \( V \). Let \( A^{(3)} \) correspond to \( A* \) so that \( c(2) = \pi/2 \). Since \( b(1) \) is known, \( a \) and \( a(2) \) are also easily computed. Choosing \( w(3) = 1, w(1), w(2) \) and \( d \) are again determined, as are \( A'^{(1)} \) and \( A'^{(2)} \), from the previous algorithm. Now only \( k \) is unknown for the final step and this can be found from (Eqn. 2) as

\[
k = (w(1) / d) \sin a
\]
since \( \sin c(2) = 1 \). Thus
\[ A^1(3) = P^5(A^*) \]
is easily obtained. Now that \( k \) is known, (Eqn. 1) gives
\[ i^2 - 3 = i^1 - 2(k - 1) \]
the time interval between the observation of \( A^2(2) \) and the expected passage at \( A^* \).

If the observer is the point "A" and the stationary observed point is "Y", then similar arguments will provide an algorithm for the successive spherical image points of "Y" as "A" moves. Another variation is to predict the appearance time of an object point occluded in its trajectory. Here \( A(3) \) would be the known intersection of the trajectory arc with the far edge of the occluding object. The unknown \( t(3) \) can then be computed.

As an important corollary to the basic prediction algorithm, changes in velocity may be detected if the trajectory does not change too rapidly. If the object feature is not found at the predicted point, then its discovery to one side indicates whether the velocity has decreased (\( A(3) \) between \( A(2) \) and \( A'(3) \)) or increased.

1.3.2. Rotation

In the trajectory section the assumption was made that translating objects are not rotating. To alleviate this situation we need to recognize rotations and know when a rotation and a trajectory are composed. The rotational component must be found first, then the trajectory computation is applied to generic object features such as topmost, frontmost, rearmost, and bottommost.

To find rotational motion alone the piecewise linear trajectory computation might still be applicable since the trajectories of object features lie in parallel planes through the axis of rotation. With an appropriately selected set of feature path pairs satisfying the further condition that their paths are approximated well by parallel lines, a set of trajectories of these features may be computed. These will not all exhibit a common direction, but instead will "fan out" in
several directions. Under optimal circumstances these vectors will lie in a plane and the perpendicular to this plane will be parallel to the axis of rotation.

Where an object has planes that permit the computation of plane slant, rotation is determined from changes in the direction of the slant. Then the parallel feature path assumption can be ignored. The axis of rotation of the car in Example 1 is computed in this manner from the orientation changes of its planes.

Finding rotation from orientation changes avoids the problem of "apparent rotation": the effect presented by a passing object where first one end, then a side, then the other end is visible. It appears to have turned around, but of course this is an artifact of its path past the observer. It is not rotating as long as the orientation of its features with respect to the environment does not change.

Rotation may also be determined visually, by feature appearance or disappearance, where plane surface orientations are not available. This involves recognizing the changing two-dimensional views of an object and relating the observed sequence of views to orientation directions of the observer. An approximate axis direction for the rotation may be obtained as the continually-present or continually-non-present view and its opposite. For example, we always see a TOP view of the turning car in Figure IV.2 and never see it in Figure I.2. In both cases we could deduce rotation about the TCP-BOTTOM axis, provided rotation was otherwise perceived. The actual direction chosen of course depends on the sense of the rotation.

If an axis direction is known, the orientation of a point with respect to that axis can be found as the intersection of the plane perpendicular to the axis direction vector and the great circle arc through the desired point and a point on the axis. For example, if the direction of the shoulder axis is known, then the orientation of the arm is found from the arc through the "fixed point," the shoulder, to the hand or the elbow. Even though the axis is known approximately, this is a good estimate for spatial orientation. If the object points do
not lie in a perpendicular to the axis, this will still suffice to detect changes in the orientation angle. If the axis direction is unknown the vector from the viewpoint to the approximate axis line intersection with the object can be used. The rotation is then assumed to be perpendicular to the line of sight.

Changes in the angular velocity of a rotating object are easily computed from the arclength of the orientation change over time. The precision of the computation depends mostly on the consistency of the axis direction in successive views.

1.4. Compound motions

In the event graph there must be descriptions for low level movements, such as the swinging motions of a person’s legs and feet during walking, as well as more generic or higher level movements, such as the concepts of WALKS and CRAWLS. These are required because low level information is too prone to pictorial error or even simple non-existence. For example, if a walking person passes behind a box and his legs are temporarily occluded, we would not want to conclude that he stopped walking. A single event WALKS must characterize the action. Since WALKS is quite a high level concept (section IV.3.1.1), we cannot expect it to match pictorial information directly. An organization of the event description in terms of other events must be exploited.

For the description of compound movements of hierarchically connected (that is, tree-structured) objects an analysis proceeds "top-down." Knowledge of the movements of the highest level part (namely the translation and rotation of the entire object) is required before the lower level movements can be so characterized. For example, the motion of a walking person (consisting mostly of translation) can be accounted for so that the swinging arm describes (say) a simple rotation about the shoulder joint rather than a complex motion including the body translation. Similarly the path of a point on the wheel of a car describes a relatively complex spatial curve (a cycloid) while actually only rotating with respect to the translation of
the car. Thus the description of the events at the highest level for a compound object is not only useful, but actually essential in order to understand the lower level movements in context. The object recognition algorithm works in a top-down fashion as well (section V.2.2). It is worthwhile noting that graphical animation languages [Badler (1972), Baecker (1969)] require precisely the same kind of top-down processing for motion generation, using simple movement primitives (translation and rotation) or path description techniques as developed by Baecker. There is an analogous situation here, where a higher level path may not have any simple description, so that it is essential to recognize the path of the higher level object before the movement of its parts.

Unfortunately the process of accounting for the motion of higher level parts of an object in the observed motion of the lower level ones is not completely foolproof. In the case of translation of the higher level, the lower level movements may be isolated by considering the orientation of the part. Since the objects we are allowing have joints if they have movable parts, the lower level motion must be rotational. This can be used to determine the motions of legs and arms on a walking person as FORWARD or BACKWARD. (See, for example, its use in section IV.2.2.2.)

In the case of rotation of the higher level part things become stickier. Begin by assuming that the rotational axis is vertical. If we imagine a person spinning, say on skates, then we know the arms and legs are turning about a certain definite axis. However this information does not directly compensate for the rotation, as knowledge of translation did, because there is no simple visual relationship between the rotational path of a point in space and its visual path. We could expect that movement in planes parallel to the axis (see Figure V.5a) would be more noticeable than movements in planes perpendicular to the axis (see Figure V.5b) because the latter would essentially be angular velocity variations in the global rotation. Thus the raising or lowering of the skater's arms would be more noticeable than, say, the non-uniform rotation of the head, because’ the
former can be seen as movements in a virtual plane facing the observer. Spatial movements of a point are "projected" into a two-dimensional plane in which the observer can track rotations around the attachment area. It should be obvious that he can track the translation of the axis line, if necessary, shifting the rotation origin.

So far the rotating parts essentially had a fixed point on the axis, namely arms, legs and head about a body axis. The spatial orientation of such a part relative to the (known) axis is easily found (section V.1.3.2). If the objects of interest are relatively distant from the axis, as for example wrist or fingers, we would have to carefully chain rotations through two or three joints to analyze the movement of the hand. People have considerable difficulty performing this operation and it seems to require conscious effort to isolate such distant events (in the sense of depth of compound movements). It is doubtful that the movements of hands and feet during rotation of the body, or even translation for that matter, are automatically extracted from the visual data. Rather they are assumed to inherit the rotational motion of the arm (for hands) or the leg.

![Diagram](attachment:figure_v_5_compound_rotation_vertical_axis.png)

**Figure V.5. Compound rotation, vertical axis**
(for feet). This is not to claim that such movements cannot be described, only that description is rarely performed and usually only when pressed by conscious effort or need.

When the rotation is not about a vertical axis, the movements can be "mentally" re-projected onto a vertical plane. As the axis direction approaches the horizontal plane this projection is useless. Concomittantly, however, the rotation itself becomes visually characterizable as a rotation in two dimensions (see Figure V.6a). The axis, now a point, allows a description of movements in the plane of the rotation with the axis as the relative origin. Less well described are movements in planes parallel to the axis (see Figure V.6b). Fortunately the objects considered here have few, if any, motions associated with this appearance of movement. A toy cart with overlong axles yields an example where translatory (sliding) motion of the dominating object (the cart) may be present independently of the rotational motion of the wheels. Translation may be forced by a dependence situation however, witness the motion of a nut along a bolt where the rotation of either one may force movement along the axis.

The restriction to objects with tree-structured connections is essential so that linkage analysis or mechanical statics are not needed. Otherwise the movement of one part of the object cannot be isolated from the movements of others. The system does not attempt such an analysis because it is beyond the realm of the heuristic and into that of the mechanical. Analytical tools may be used at this point.
1.5. Spatial Layout

The context of objects in a scene is basic for many of the directional adverbials for describing motion. The objects in the vicinity of the observer are perceived in a "ground plan" or plane map created by the location relations between object features in the ground plane. As a corollary, distance determination is based on angular measures in the plan, rather than absolute distances. When object SIZE values are not known, distances may be estimated relative to the visual appearance of objects, leading to a simple heuristic for metric ADJACENCY relations.
1.5.1. Location Relations

Let A and B be two points on the same horizontal plane. A
and B might, for example, belong to the bottom edges of two
objects Ch(A) and Ch(B) lying on the same support plane. The
SUPPORT-PLANE relations between A and B may be determined from
information available in the spherical projection picture.
Define $R(A, B, C)$ to mean that A is related to B by the relation R
(relative to C). This may be read with the appropriate
substitutions as "A is R B relative to C". The set of relations
from which $R$ may be obtained is based on combinations of
SUPPORT-PLANE relations:

$$\{IN-FRONT-OF, IN-BACK-OF, RIGHT-OF, LEFT-OF\}$$

Thus $R$ may be $IN-FRONT-OF$, $IN-BACK-OF$-$AND$-$RIGHT-OF$, $LEFT-OF$, and
so on.

The relations in SUPPORT-PLANE are found by reference to
the horizontal plane. The direction of an arc AB in a plane P
is precisely its intersection with the great circle arc formed
by a plane parallel to the containing plane P. To compute this
direction requires only the points A and B and the normal to the
containing plane. For a horizontal plane this normal is just
$(0,0,1)$. The intersection point chosen is the one in the R
direction from A.

For the first case consider $R(A, B, W)$ where $R \neq W$. Divide the
circle $\mathcal{O}=\pi/2$ about $W$ into eight sectors

$$R(i) = (\phi \in \mathbb{E}[-\pi/8 + i\pi/4, i\pi/8 + i\pi/4]) \pmod{2\pi}$$

for $i=0,\ldots,7$. Then

- $R(0) \Rightarrow \text{IN-FRONT-OF}$
- $R(1) \Rightarrow \text{IN-FRONT-OF-AND LEFT-OF}$
- $R(2) \Rightarrow \text{LEFT-OF}$
- $R(3) \Rightarrow \text{IN-BACK-OF-AND LEFT-OF}$
- $R(4) \Rightarrow \text{IN-BACK-OF}$
- $R(5) \Rightarrow \text{IN-BACK-OF-AND RIGHT-OF}$
- $R(6) \Rightarrow \text{RIGHT-OF}$
- $R(7) \Rightarrow \text{IN-FRONT-OF-AND RIGHT-OF}$

After the direction $\Theta(AB)$ from $A$ to $B$ is computed, select that
$R(i)$ for which

$$\Theta(A) - \Theta(AB) \pmod{2\pi} \in R(i).$$

Consider the case $R(A, B, R)$ where R has some intrinsic
orientation such that a direction from R can be called the
FRONT. (This assumes that $\phi$ of the direction is near $\pi/2$.) The
direction may be obtained by choosing some visible baseline as points $A'$ and $B'$ and computing their direction $\theta(A'B')$ relative to $V$. Alternatively the FRONT may be estimated from the visible view of the object $\text{Ob}(R)$. Let $D$ be this FRONT direction. After the direction $\theta(AB)$ from $A$ to $B$ is computed, select that $P(i)$ for which

$$\theta(AB) - \theta(D) + n \mod 2n \in P(i).$$

Note that this is compatible with the case $R(A,V,V)$: here $\theta(D) = 0$ and $\theta(AV) = \theta(A) + n \mod 2n$. Hence $\theta(AV) - \theta(D) + n$ simply reduces to $\theta(A)$, yielding the convention of section II.1.

A similar process to discriminate ELEVATION relations is unnecessary since any constraint forcing consideration of points $A$ and $B$ in the same vertical plane immediately yields an ABOVE or BELOW distinction from a comparison of their $\phi$ coordinates.

In actual use the computation of location relations must be tempered by higher level considerations since the relation obviously depends strongly on the choices for $A$ and $B$. One possible choice is the midpoints of the visible bases of $\text{Ob}(A)$ and $\text{Ob}(B)$. This often works, but does not quite give the desired result in cases where the extents of the two objects differ greatly. For example, a situation illustrated in Winston(1970) (see Figure V.7) generates the relation $\text{Ob}(A)$ is IN-BACK-OF-AND-RIGHT-OF $\text{Ob}(B)$. The IN-BACK-OF part does not seem appropriate; the difficulty is due of course to the choice of $A^1$. A better heuristic strategy would be to choose $A$ and $B$ so as to minimize the distance between $A$ and $B$. Since at least one of these points may frequently be occluded, a second approximation would be to choose a point on the longest baseline closest to the other baseline. This would result in the choice of $A^2$ in Figure V.7, and the response now would be that $\text{Ob}(B)$ is RIGHT-OF $\text{Ob}(A)$.

As another example, again from Winston(1970), consider Figure V.8. Calling the direction along the road the FRONT of the arch (assume that this is part of the program's knowledge about arches in the object model), and choosing $A^1$ as the closest point of $\text{Ob}(A)$ (the arch) to base point $E^1$ of $\text{Ob}(B)$ (the box), one immediately obtains that $B$ is IN-FRONT-OF-AND-RIGHT-OF
Figure V.7. Two blocks (after Winston)

Figure V.8. Block and arch (after Winston)
A relative to A. One should note that neither of these situations is satisfactorily handled by Winston (1970) nor in later efforts by Finin (1971).

In general other tests will have to be applied, especially when the objects touch or share a common edge. Often relations will conflict when computed for various choices of point pairs, but this can lead to decisions about the more subtle comparisons between objects. Thus in Figure V.9, Ob(B) is IN-BACK-OF Ob(A) according to A\textsuperscript{1} and B\textsuperscript{1}, but IN-FRONT-OF according to A\textsuperscript{2} and B\textsuperscript{2}. These are valid relations between parts of objects, not the object as a whole. Such problems are beyond the scope of the present investigation.

If it has not already become apparent, three-dimensional information such as depth or actual dimensions of objects were unnecessary for these computations of LEFT-OF, RIGHT-OF, IN-FRONT-OF and IN-BACK-OF. The only assumption was that the points considered lay in the same horizontal support plane. In fact the plane need only be horizontal within some very generous tolerance, say of ±30°, because the small vertical error introduced will not noticeably affect the θ coordinate of the computed direction.

![Diagram](image)

*Figure V.9. Ambiguous relations*
The division of the circle into relations by equal sectors is arbitrary and other segmentations are possible. Psychological experimentation could possibly refine the boundaries. The angles of the "pure" relations (LEFT-OF, RIGHT-OF, IN-FRONT-OF, IN-BACK-OF) could be decreased and new sectors added for shades of meaning (SLIGHTLY-LEFT-OF, SLIGHTLY-RIGHT-OF, SLIGHTLY-IN-FRONT-OF, SLIGHTLY-IN-BACK-OF). We could imagine a spectrum of semantic combinations:

<table>
<thead>
<tr>
<th>1 &amp; VERY SLIGHTLY</th>
<th>1 &amp; SLIGHTLY</th>
<th>2 BUT MORF 1</th>
<th>1 &amp; 2 BUT SLIGHTLY</th>
<th>2 &amp; SLIGHTLY</th>
<th>2 &amp; VERY SLIGHTLY</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For example, if 1 is LEFT-OF and 2 is IN-BACK-OF, the resulting spectrum of concepts is

- LEFT-OF
- LEFT-OF-AND-VERY-SLIGHTLY-IN-BACK-OF
- LEFT-OF-AND-SLIGHTLY-IN-BACK-OF
- IN-BACK-OF-BUT-MORE-LEFT-OF
- IN-BACK-OF-AND-LEFT-OF
- LEFT-OF-BUT-MORE-IN-BACK-OF
- IN-BACK-OF-AND-SLIGHTLY-LEFT-OF
- IN-BACK-OF-AND-VERY-SLIGHTLY-LEFT-OF
- IN-BACK-OF

The angular locations of these divisions might be based on the first four standard deviations of a normal distribution about the mean. An encoding using a real value (for example, LEFT-OF(0.33), IN-BACK-OF(0.67)) is possible but only seems to delay a decision which must be made eventually anyway.

1.5.2. Plane Maps

Leaving the problem of placing labels on directions, let us see what further information may be obtained concerning the static arrangement of the scene. Again assume that all objects are supported by a single approximately horizontal plane (the ground, for example). (See Figure V.10.) Choose some point \( V^i \) at the intersection of a known fixed object and the ground. For greater accuracy \( V^i \) should be close to the viewpoint \( V \); this also reduces the chance that \( V^i \) will become occluded by moving objects. \( W \) is the point in the ground plane directly below \( V \). By applying the algorithm for \( R(V^i, R(i), V) \) to (all) points \( R(i) \) which are on baselines of objects in view, we get a triangle where all three angles are known: for \( i=1 \), say,
Choose \[ ||W'V'|| \] arbitrarily to be 1. Then every point \( B(i) \) is uniquely located in the horizontal plane in terms of the base length \[ ||W'V'|| \]. If this length is known in terms of other units (feet, meters, etc.) then all distances \[ ||WB(i)|| \] and \[ ||V'B(i)|| \] are similarly determined. More generally, since the angles from \( W \) and \( V' \) to all the \( B(i) \) are known, any single known dimension determines all other dimensions. The result is that this implicit triangulation from \( W \) and \( V' \) provides an immediate ground plan for the objects in the vicinity of the viewpoint. This ground plan is obtained in a size-independent manner (in terms of angles only) and may be scaled to convenient dimensions if a reference length is available. There seems to be no real need for this scaling to take place, however, as plans of
miniature circuit boards or maps of the earth are understandable
with a different-sized, though angle-preserving, representation.

The advantage of this scheme for depth mapping is that no
 correlation operation between stereo views of a scene need be
 performed. The implicit viewpoint \( V' \) assures that angles will
 always be measured relative to a fixed reference point; there is
 no chance of mismatching points in a stereo correlation. It
 must be emphasized that only a single image is required.
 Although these computations require an approximately horizontal
 plane, they can be used in any situation where support planes
 not in the ground plane are themselves supported. Therefore
 this scheme may be applied recursively to every supporting plane
 once its supports are themselves located. In some visual
 situations the ends of the supports might not be visible (as in
 a table with the legs in from the edges and viewed from above)
 but then we are more likely to satisfy ourselves with the
 position of an object on the table rather than its position with
 respect to the floor.

We believe that this simple monocular mechanism can account
 for one of the famous illusions relevant to the psychology of
 perception. To illustrate, the famous Ponzo illusion (1928)
 (see Figure V.11), where the top line appears longer than the
 equally-long bottom line, is often explained by size constancy
 resulting from a perception of depth [Murch (1973), page 236].
 However, the reasons for judging depth in the converging diagonal
 lines are not perfectly clear, nor does this theory explain why
 the top line should appear to lie in the same plane (that is, it
 could appear to float above the other line independently of the
 depth judgement on the diagonal lines). The proposed algorithm
 says simply that the observer assumes that the lines all lie in
 a ground plane (since there are no contrary indications) and
 that from any point in the picture as \( V' \) a layout such as Figure
 V.12 is proposed. The greater length of the further line is
 then immediately apparent. That this directional scheme is
 stronger than size constancy arguments is supported by the fact
 that at no time were any assumptions made regarding size or even
 depth, for that matter, of any of the lines.
Figure V.11. The Ponzo illusion

Figure V.12. Predicted map for Ponzo illusion
1.5.3. Distance determination

There are several sources for reference distances in the scene. The most valuable is the height of the observer (the distance \( h \)) which determines the distances from \( W \) to base points of objects (Figure V.10). This has been used by Roberts (1965). Other sources are the objects themselves, for if they have size properties and the features referred to are visible, a simple calculation gives their distance from the observer in any direction. From the angle \( c \) subtended by those features in the image and the known size \( ||AB|| \),

\[ d = ||AB|| \left(2 \tan\left(\frac{c}{2}\right)\right)^{-1}. \]

Finally, we can use the temporal analog to binocular vision, motion parallax, to decide the depth of an arbitrary point from two images. The details will not be necessary here.

The locative relation angle \( p=\theta(\text{Ob}(A),\text{Ob}(B)) \) between \( \text{Ob}(A) \) and \( \text{Ob}(B) \) and \( b=\theta(W,\text{Ob}(A)) \) between \( W \) and \( \text{Ob}(A) \) in a dimensionless plane map provide a simple algorithm for the estimation of \( \text{FAR-FROM}, \text{NEAR-TO} \) and \( \text{AT ADJACENCY} \) relations. (See Figure V.13.) Define the extent \( d(\text{Ob}) \) of an object as the minimum arclength over points in its current image. (See Figure V.14.) Using this as a reference in a local plane map we can take

\[ d = \left(2 \tan\left(\frac{d(\text{Ob})}{2}\right)\right)^{-1} \]

as a heuristic measure of the relative distance of \( \text{Ob} \) from \( W \). Taking the distance from \( W \) to point \( A \) of \( \text{Ob}(A) \) as \( d \), the distance between \( \text{Ob}(A) \) and \( \text{Ob}(B) \) along the relation angle is just

\[ d(AB) = d \sin p / \sin b. \]

If \( \sin b \) is very close to zero, then \( d(AB) \) may be obtained by a simple proportion in \( \phi \):

\[ d(AB) = d \tan \phi(B) / \tan \phi(A). \]

This supplies the recognition algorithm for \( \text{FAR-FROM}, \text{NEAR-TO} \) and \( \text{AT} \):

- \( \text{Ob}(A) \) is \( \text{FAR-FROM} \) \( \text{Ob}(B) \) if \( d(AB) > 1 \),
- \( \text{Ob}(A) \) is \( \text{NEAR-TO} \) \( \text{Ob}(B) \) if \( 1 \geq d(AB) > 0.2 \),
- \( \text{Ob}(A) \) is \( \text{AT} \) \( \text{Ob}(B) \) if \( 0.2 \geq d(AB) > 0 \).

-202-
Figure V.13. Adjacency relations

Figure V.14. Extent of an object
Obviously the choice of the actual ranges is quite subjective, but the concepts we are dealing with are also quite "fuzzy" [Freeman (1973)]. In addition, since the above choices depend on the extent of only one of Ob(A) or Ob(B), the necessary asymmetry in these relations is introduced as section III.2.5.1 required.

V.2. Recognizing objects

The final but non-trivial problem is recognizing instances of objects in the individual static frames that constitute the scenario. By our initial assumptions, we are ignoring the real-life case of continuous perception of the world, substituting instead the motion picture paradigm of discrete images. This at once frees us from a tangential investigation of temporal picture processing in the low level sense [for example, see Baumgart (1973)], but also constrains us to utilize current static scene analysis methodology in an efficient manner. The goal of this section is to present an object recognition scheme which basically relies on a top-down, model-driven, static analysis. The temporal efficiency is a result of the top-down approach, though no claims are made regarding actual recognition times for any particular scene or image sequence.

2.1. Structural models

In section III.2.2 we discussed SUB-PART edges which defined a structural model of an object. Each model node represents a certain named or otherwise significant part of an object. In the object model, the IMAGE properties (section III.2.5.4) define whatever pictorial (visual) properties that part of the object exhibits, if this is meaningful. For example, the parent object model node for a PERSON might have no IMAGE property associated with it because its structure is a composite of lower level SUB-PART appearances.

Each part of an object may have several intrinsic ORIENTATION values (section III.2.6). A FOOT, for example, may have a TOP, a BOTTOM, a FRONT and so on. These may be

-204-
independent of the ORIENTATION of higher level parts, especially if joints intervene. In a particular position, for example, the BOTTOM of the foot may face the same direction as the FRONT of the body. The ORIENTATION indicates the appropriate IMAGE features that characterize each view, thus not all features need be observed to recognize the part.

Because of the hierarchic nature of the structural model, recognition of an instance of an object may proceed in a top-down fashion, higher level parts requesting lower level parts and so on until specific IMAGE features are observed and the lowest level parts recognized. At the bottom, the IMAGE features invoke specific low level operators to detect points, lines, curves or regions in the actual image.

Bottom-up traversal of the structural model could be used, too. In this case we would attempt to recognize an object from observation of low level parts which eventually fit together into a whole. Such a scheme has been proposed by Uhr (1973). However, it is difficult to say that something is, for example, a leg, unless we are in fact looking for a part of an object which has a leg. A "leg-shape" may characterize many other objects as was pointed out by Guzman (1971). So we must be careful in using bottom-up techniques.

The obvious situations for bottom-up recognition are the initial view of a sequence and the appearance of previously unseen objects later in the sequence. We will talk about the former for simplicity, though there is actually no difference in the recognition process which we will describe later. At the beginning of the sequence nothing has been seen, so there is no developed object description. Applying bottom-up recognition to the first frame would produce a description of the scene which could be used thereafter in a top-down manner. However, a model-driven analysis always proceeds top-down because no object will be recognized unless an object model exists for it. That is, to recognize even the initial instance of an object, that object must be in the repertoire.

Once the initial instance is discovered from the object model the recognition follows the object description. By using
the same format for the structural description as for the structural model, the matching process can be uniformly defined for both initial instance and later verification steps.

This last claim in no way predicts that recognition times for first frames and later frames will be comparable. On the contrary, it localizes the source of a greater recognition time for the first frame in the multiplicity of object models which are attempting to find instances of themselves in the scene. If the object model universe is small, then few models will be competing. In a large universe, however, contextual clues as to what object models are likely to be required must be provided to make the system feasible. (Very general notions of what the scene is about may suffice, for example, a room or outdoors. Significantly, research in low level scene analysis has had success with these more general features. See, for example, Brice and Fennema (1971) for rooms and Bajcsy and Lieberman (1974) for outdoor scenes.) Efficiency and accuracy is known to be enhanced in single images by model-driven analysis [Shirai (1972), Grape (1973)].

As has been pointed out by Barrow et al. (1972), Grape (1973), and Masumi (1973), model-driven analysis allows the possibility of recognizing partially occluded objects as belonging to one or more object models. A related problem arises in the temporal scenario where the initial image may present an ambiguously occluded object. The scene description must be built up frame-by-frame unless other tactics can provide an object identity (inference rules, for example). We have assumed here that the initial frame by itself supplies an unambiguous scene to the viewer, but there is no need to so restrict the sequence. Underwood and Coates (1972) use several images to construct a description of a polyhedron. Another approach to description generation from consecutive images is given by Baumgart (1973). Much more work needs to be done in this area.

Once the object has been observed in two successive images, the IMAGE property of each part will show the changing pictorial locations of object features. This must follow from the proper recognition of those features from the model. The displacements
of unoccluded features are the basis for computing trajectories, rotations and observer movement. We will say the object images are **correlated** if some subset of object features have explicit image coordinates in two frames. Obviously the movements of objects cannot be described unless the exact appearance an object presents in each frame is known.

2.2. **The recognition algorithm**

The object recognition algorithm is a slight modification of a technique called "hierarchical synthesis" which was proposed by Barrow et al. (1972). The hierarchy consists of a set of modules corresponding to object parts. Modules tell other modules that they have or require certain recognized parts. The correspondence between Barrow's modules and our system comes by associating with each object part model node a demon [module] which communicates with other demons along SUB-PART edges. The difference arises from the fact that the network of modules is only activated from the top as described in the previous section. Barrow's network can be used in a strict bottom-up manner as well.

For each object model part node \( M \) there is a demon which looks like:

**precondition 1**

```plaintext
A PART-OF\( (M) \) node demon activates this demon.
```

In response to activation, all demons associated with SUB-PART\( (M) \) nodes are activated.

**precondition 2**

```plaintext
All or certain allowed combinations of the object nodes SUB-PART\( (M) \) are instantiated, provided they satisfy other constraints which may be required.
```

**postcondition**

```plaintext
The object node associated with this demon is instantiated with SUB-PART edges to the set of nodes appropriate as above.
```

By specifically activating the demons associated with the highest level object model node for an object, the recognition message is transmitted down the SUB-PART graph. Activation of
the top level is triggered by the successful analysis of the current frame and the request for a new one.

The constraints and combinations allowed in precondition 2 are determined by other properties on the model node: CONNECTED-TO for connectivity and ABOVE, LEFT-OF, IN-FRONT-OF and so on for spatial relations between parts. The necessity of considering valid combinations arises from the probable recognition of several instances of simple generic features at the lowest level. The economy of this recognition process is discussed in the Barrow reference.

Note that a demon at any level may find more parts than it needs in order to be satisfied. The demon is free to instantiate more than one node representing its part. This would happen normally in a scene with several objects of the same type.

The algorithm is defined so that demons are effectively activated in parallel. This is somewhat undesirable for practical purposes and also from considerations of contextual recognition. For example, we mentioned in section III.2.5 the use of location relations in the object model as guidelines for object recognition. To utilize these relations the demons must "know" what other demons have found. (Bottom-up processing does not completely solve the problem either, since object identities may not be known when that knowledge is important.) Beyond some simpleminded heuristics we do not have any sound ideas for a solution. Goal-directed formalisms may provide a convenient programming vehicle [Hewitt et al (1973)].
2.3. **Object correlation**

From the recognition algorithm we can see that previously instantiated object nodes indicate to higher level part demons that lower level structures have been seen. In the analysis of the next frame, this knowledge can be used to produce a more efficient recognition of the same object. The object model is still the source of the demon communication structure (as parts of the object may have been occluded), but now nodes may already be instantiated in the data base from prior observations. Demons which created these nodes can use the locational information to define a reduced search space for the new image location of the part.

If the object did not move from the previous two frames, then the saved location values of its features are used to verify the location in the new frame. Otherwise the differing location values (if significant) may be used in conjunction with trajectory prediction to hypothesize a new location for the feature. This would only be used for search localization, as there are numerous reasons why the prediction may fail to be accurate or fail completely (occlusion, certain rotations, radical trajectory changes, and so on). Prediction has only been discussed in the context of a uniform trajectory (section V.1.3.1).

Individual lower level parts may change pictorial location significantly, while the whole object may not. (We are using our assumption that objects in the image sequence appear to be moving smoothly to a human observer.) For example, the swinging arms and legs of the man in Example 4 are not likely to intersect their own outlines in successive frames. The man as a whole, however, would be more likely to. The attachment points for the arms and legs could then be located from the recognized body.

It is therefore desirable to compute trajectory predictions with respect to the largest (and therefore highest level) object part for which the prediction is obtainable. The search space for the object in the next image is therefore reduced to the predicted outline (or else simply the old outline) and then
augmented to account for reasonable changes. It may be possible
to define the additional area if other parameters of the object
being observed are known, for example, mass (for momentum) or
its ability to maneuver. A system which does this for
automobiles in traffic flow movies is described by
Wolferts (1974). He uses a more sophisticated locational model
than just an outline for object correlation.

There is an important consequence to this last discussion.
It argues against using low level picture differencing as an
effective method of tracking moving objects, unless higher level
information regarding what is an object and what are its
possible motions are expressed in the data base. Picture
differencing, while useful in detecting changes in pictorial
data [Quam (1971), Quam et al (1972), Leese et al (1970)], is much
too sensitive to actual lighting conditions. The detection and
description of motion should depend on the successive locations
of object features, or their appearance and disappearance. The
multiplicity of "features" in a raw image (look at the output of
any low level picture operator) forces the further abstraction
to the object level. Here we can finally rely on the
recognition of a large group of features as an object and regard
the movement of its features as indicative of its own movement.
This cannot be said of changes in pictorial qualities. We only
need to think of a single white block on a table while an
experimenter moves a spotlight around the room. The image
sequence would show radical illumination changes, hence
significant between-frame picture differences, yet the block is
stationary.

Working from the line or curve level is not sufficient
either. The illustrations in this thesis indicate that line
differencing alone (that is, looking for lines that have changed
position) is useless. The lines must be associated with an
object. Similarly the analysis of changing regions alone would
suffer from the same problem. In addition, line- and region-
finders operate on low level data and would (especially for
regions) inherit the illumination problem.
Although illumination was not a consideration in the movie sequences illustrated, any system which recognizes objects in successive images will have to use higher level information such as presented here. It does not matter so much what picture processing is actually performed as long as an object classification results. In fact, this argument implies that what is important for a low level operator (for example, a line- or vertex-finder) is not that it find every such feature in the picture, but that once one is found by some criteria, then that feature can be recognized in context again in spite of illumination variations. Programming such operators is a task for future research in temporal picture processing.
Chapter VI. Conclusions

Looking back at the initial goals we set for temporal scene analysis in the Introduction, we would have to claim that the goals are possible to achieve and that the methodology we presented shows how to achieve them. The input scenarios have been dynamic, illustrating a variety of movement types including translation (the car, ball and man), rotation in two different orientations on top of translation (the car while advancing and the legs while walking), rotation alone (the signals) and repetition (the bouncing ball). The analysis of motion of rigid or jointed objects as determined from two-dimensional picture sequences was elaborated to the point at which simple linguistic concepts of the movements began to fail, such as the case of multiply-jointed parts or non-tree connected objects.

The spherical projection model provided a neat conceptual formalism for characterizing object properties in dynamic scenes, covering plane slant, trajectories, rotations and spatial relations. The interpretation placed on the world as a terrestrial environment with a "human-like" observer is essential to the concepts involved. In an actual system the spherical model could be replaced by any camera model taking into account a spherical projective view of the world, but this would have only cluttered the basic geometric simplicity of the situation.

We are less sanguine, however, about the capability of computer vision systems as they exist today to extract the necessary accuracy in data from the actual images. The resolution and brightness discrimination in the human eye far exceeds contemporary capabilities in picture processing due to time and space problems. Our analysis has shown, however, that treating the visual process in the abstract yields a powerful conceptualization of the relations between objects in the vicinity of the observer and of movements of objects in space which might be lost under pictorial errors in a coarsely quantized image.

To properly evaluate the event description algorithms as they purport to satisfy the goals, we should keep in mind, not the possible raggedness of the picture data, but rather the
results of the algorithms which build descriptions. Thus if the input consisted of the data structure of a three-dimensional graphical animation system (before the visual projection and hidden line elimination) then the event algorithms would build a high level description of the movements generated by the animation system. Explicit three-dimensional models of objects admit mathematically precise algorithms for, say, support-plane and elevation relations, as well as improved contact determination. Exercising the system from this level allows a concentration on the changing relations between objects—avoiding concern about whether or not particular three-dimensional relations are present in the (often ambiguous) two-dimensional image. Of course we need to store more information about an object than current animation systems allow; the object model node indicates the additional knowledge while the structural model mimics the usual graphical object structure.

The descriptions which are derived from low level data are, nonetheless, quite valid from a conceptual point of view. We have not claimed to put the description into good surface English, but will be content to provide high level data to semantic network manipulating routines for the English generation process. Such transformations would be free to re-organize direction lists, for example.

The methodology employed for the event generation process has consequences for artificial intelligence research on the representation of knowledge. We have discussed the difference between the state approach to storing temporal data and our approach which describes the changes in the data. The model arises naturally from the types of events we sought to describe, where changes in relations were more important than the relations themselves. The path to abstraction is thus built on a foundation of low level changes, described by intermediate level changes (adverbials), themselves yielding to higher level adverbials or canonical descriptions (motion verbs).

The algorithms and examples have pointed out that the places where ambiguity in the description arise can be localized if the input data is not faulty. The case of CONTACT or non-
CONTACT is the only one at the intermediate level. Higher level ambiguity results from the problem of inferring AGENTS and INSTRUMENTS, and hence in deciding among various motion verbs.

By using a demon methodology for the recognition of concepts at every level, we are assured of building the description as high and as quickly as possible. Because every demon leaves its name (or a distinctive mark) in the new relationship which it discovers (for example, in DIRECTION), the process is somewhat reversible. Old static descriptions are certainly discardable, in fact, the system maintains only the current description plus a few minimal queues of values for derivatives and for image point location. Old (terminated) event descriptions may also be deleted, but we would have to go beyond the level of motion verbs alone to describe the changing events, for example, as EXCHANGES or JOURNEYS.

From a psychological viewpoint the analysis of motion and its relation to linguistic concepts could serve as a framework in which to further the study of dynamic perceptual processes. The works of Gibson (1950, 1966) and Piaget (1937, 1946) seem to fall at the beginning and end of the present investigation. It is quite conceivable that this computational methodology for transforming sequences of images into a semantic relational structure can help unify some of the diverse explorations in the psychology of events.

There are many problems that need to be investigated or at least confronted in an actual implementation of this system, even without worrying about live TV data. Some possibilities for future work follow.

- It should be possible to extend the ground plan algorithms in a consistent manner in order to build a three-dimensional model of the surrounding environment. Determining spatial relations would become less heuristic and algorithms for contact and support would become more accurate.

- Object recognition cannot always be expected to yield an object identity if, for example, the object is distant, in a visually-cluttered region, or partially occluded. It would be useful to infer object identity from observed function or usage.

- We need more inference rules for the understanding of motion at the conceptual level, that is, how we infer causality from action.
• A lexicon for motion verbs must be developed. At this level it seems that some sort of "learning algorithm" for the definitions is essential.

• The restriction to rigid objects should be removable by an investigation into object characteristics and their relation to visual events.

At least two independent implementations (at the University of Toronto and at the University of Pennsylvania) of the ideas in this thesis have been projected and it is expected that there will be some changes, adjustments, and new problems. We are convinced, however, that the basic design is methodologically sound and meets the requirements we set in the beginning.


Appendix II. Sample linguistic descriptions of events

The following are extractions from descriptions of the movie illustrated in Figure I.2 provided by fifteen observers instructed to briefly describe what they saw. The phrases have been slightly altered to unify tense and object names. The descriptions are organized by object, there being four independently moving objects in the scenario: the signals (as a unit), the train (again as a unit), the car and the viewer. The viewer was considered by some to be an object in the scene, that is, as a car passenger or a person on foot. Others thought of themselves as a passive viewer of a movie.

Within each object's motion description the phrases used in any description are ordered temporally. That this was possible to do at all was somewhat surprising considering the number of possible interpretations of the various movements. Temporally coincident (or overlapping) groups of events are numbered and succession is indicated by "[then]."

A superscript integer before a phrase indicates the number of repetitions of that phrase (or a tense or reference variant) in the whole sample.

SIGNALS

1. *swinging
   1swinging to and fro
   1swinging about seven times
   1moving
   1start waving
   1start flashing
   1announcing train
   1indicate the approaching train

[then] 2. 1stop swinging

TRAIN

1. *approaches a crossing
   1approaches from the left
   1enters from the left
   1appears from left
   1comes from left
   1moves to right

[then] 2. 3passes
   1passes by
   1passes over road
   1passes in front of scene
   2crosses the road
   1goes across
   1goes across the road on the tracks
   1flashes by
   1goes by
   1moves across the scene from left to right
moves in from left to right

[then] 3. disappears
disappears to the right
goes off to the right
is gone

CAR
1. pulls out of the driveway
pulls out from the house
started to pull out from the house
comes out of the driveway
comes down the driveway
leaves the driveway
moves out
picks up in front of viewer
pulls out in front of the house

[then] 2. turns left down the road
turns left

[then] 3. continues in front of viewer
moves in front of viewer
comes out in front of viewer

[then] 4. drives away
drives away along the road towards the store
travels down road away from viewer
pulls away from viewer
continues to drive down the road to the corner
starts moving along the road
drives down the road
moves away
speeds up down the road
moves on the road

[then] 5. approaches the corner
reaches a turn

[then] 6. stops at the corner

[then] 7. turns right at the corner
turns right towards the store

[then] 8. disappears from sight

VIEWER (in car or walking)
1. stopped at the crossing
stopped
motionless viewer

[then] 2. moves forward
moves over the tracks
moves directly ahead across the tracks
crosses the tracks
moves across the tracks

[then] 3. moves along the road
moves down the road
continues along the road

[then] 4. follows the car a short way
moves closer to the car

[then] 5. bears right

[then] 6. moves across to the store
heads towards the store
approaches the store

[then] 7. pulls into the parking lot

-222-
VIEWER (passive)

1. pan in along road
   focus on house

[then] 2. zoom in close to store
   pan over to store
   focus changes to store
   close up of store
   focuses on store
   the scene was zoomed
Appendix II. English motion verbs

The following list of motion verbs is adapted from Miller (1972) with a few additions arising from concepts described by our event representation.

- absorbs
- accelerates
- accompanies
- admits
- advances
- ambles
- approaches
- arrives
- ascends
- assembles
- attends
- attracts
- bears
- bounces
- bounds
- brings
- broadens
- canters
- carries
- charges
- clammers
- climbs
- closes
- collects
- collides
- comes
- continues
- crawls
- creeps
- croissants
- dances
- decelerates
- deepens
- departs
- depresses
- descends
- diffuses
- dives
- drags
- drifts
- drives
- drops
- ejects
- elevates
- embarks
- emigrates
- empties
- enters
- escapes
- exits
- expands
- extends
- falls
- fills
- flaps
- flees
- flexes
- files
- flings
- floats
- follows
- gallups
- gathers
- goes
- grows
- halts
- hands
- hikes
- hops
- hurls
- hurries
- inches
- injects
- inserts
- interposes
- invades
- jerks
- jogs
- journeys
- jumps
- kicks
- lands
- launches
- lays
- leads
- leaves
- lengthens
- lifts
- limps
- lowers
- lumber
- marches
- meanders
- minces
- motors
- mounts
- moves
- moves
- nears
- nods
- opens
- oscillates
- overruns
- passes
- parallels
- passes
- penetrates
- pivots
- places
- plunges
- pounces
- precedes
- proceeds
- progresses
- projects
- propels
- pulls
- pursues
- pulls
- puts
- races
- raises
- rambles
- recedes
- releases
- removes
- replaces
- retreats
- returns
- reverses
- revolves
- rides
- rises
- rolls
- rotates
- runs
- sails
- saunters
- scatters
- scrambles
- scurries
- sends
- separates
- sets
- shakes
- shifts
- shoves
- shrinks
- shuts
- sinks
- skis
- slides
- slacks
- slides
- slinks
- slips
- slithers
- soaks
- soars
- speeds
- spins
- spirals
- spreads
- springs
- spraints
- squirms
- stagger
- starts
- steps
- widens
- stops
- straggles
- withdraws
- strays
- wobbles
- worms
- wriggles
- stumbles
- zigzags
- zooms

swerves
swims
swings
swoops
takes
throws
thrusts
tilts
tiptoes
toddlers
tosses
totters
tours
tows
travels
traverses
treads
trips
trots
trundles
tucks
tumbles
turns
twirls
twists
undulates
vacates
vauls
veers
visits
voyages
wades
walks
wanders
waves
weaves
whirls
widens
wings
withdraw
wobbles
worms
wriggles
zigzags
zooms
<table>
<thead>
<tr>
<th>Number</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>*1</td>
<td>T.E. Hull</td>
<td>The Numerical Integration of Ordinary Differential Equations</td>
</tr>
<tr>
<td>*2</td>
<td>J.C. Mason</td>
<td>Chebyshev Methods for Separable Partial Differential Equations</td>
</tr>
<tr>
<td>*3</td>
<td>Neil Frederick Stewart</td>
<td>The Comparison of Numerical Methods for Ordinary Differential Equations</td>
</tr>
<tr>
<td>*5</td>
<td>G.F. Gabel</td>
<td>A Predictor-Corrector Method Using Divided Differences</td>
</tr>
<tr>
<td>*6</td>
<td>Chandler Davis and W.M. Kahan</td>
<td>The Rotation of Eigenvectors by a Perturbation - III</td>
</tr>
<tr>
<td>*7</td>
<td>D.J. Cohen and C.C. Gotlieb</td>
<td>The Syntax Graph: A List Structure for Representing Grammars</td>
</tr>
<tr>
<td>*8</td>
<td>D. Tsichritzis</td>
<td>Measures on Countable Sets</td>
</tr>
<tr>
<td>9</td>
<td>Brian Smith</td>
<td>Error Bounds, Based Upon Gerschgorin's Theorems, for the Zeros of a Polynomial</td>
</tr>
<tr>
<td>*10</td>
<td>M. Puzin</td>
<td>Simulation of Cellular Patterns by Computer Graphics</td>
</tr>
<tr>
<td>*11</td>
<td>J.C. Mason</td>
<td>Orthogonal Polynomial Approximation Methods in Numerical Analysis</td>
</tr>
<tr>
<td>*12</td>
<td>Donald M. Kaplan</td>
<td>Recursion Induction Applied to Generalized Flowcharts</td>
</tr>
</tbody>
</table>

* Not available
TECHNICAL REPORTS
(continued)

<table>
<thead>
<tr>
<th>Number</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>*13</td>
<td>Sunil K. Pal</td>
<td>Numerical Solution of First-Order Hyperbolic Systems of Partial Differential Equations</td>
</tr>
<tr>
<td>*14</td>
<td>Pat Conroy</td>
<td>Simulation of Texture by Computer Graphics</td>
</tr>
<tr>
<td>*15</td>
<td>J.C. Mason and I. Barrodale</td>
<td>Two Simple Algorithms for Discrete Rational Approximation</td>
</tr>
<tr>
<td>*16</td>
<td>S. Gerschgorin and Paul Turan</td>
<td>Two Translations in Numerical Analysis</td>
</tr>
<tr>
<td>*17</td>
<td>George Olshevsky Jr.</td>
<td>An Automated Aid to Visualizing Convex Hypersolids</td>
</tr>
<tr>
<td>*18</td>
<td>Derek Gordon (2nd ed.) Corniel</td>
<td>Graph Isomorphism</td>
</tr>
<tr>
<td>*19</td>
<td>Alan Borodin</td>
<td>Computational Complexity and the Existence of Complexity Gaps</td>
</tr>
<tr>
<td>*20</td>
<td>John David Duffin</td>
<td>A Language for Line Drawing</td>
</tr>
<tr>
<td>*21</td>
<td>A. Ballard</td>
<td>Transformations on Programs</td>
</tr>
<tr>
<td>*22</td>
<td>S.A. Cook</td>
<td>Linear Time Simulation of Deterministic Two-Way Pushdown Automata</td>
</tr>
<tr>
<td>*23</td>
<td>D. Tsichritzis (2nd ed. Editor)</td>
<td>Topics in Operating Systems</td>
</tr>
<tr>
<td>*24</td>
<td>R. Bunt (2nd ed.) and D. Tsichritzis</td>
<td>A Selective Annotated Bibliography for Operating Systems</td>
</tr>
<tr>
<td>*25</td>
<td>J.C. Mason and I. Farkas</td>
<td>Continuous Methods for Free Boundary Problems</td>
</tr>
</tbody>
</table>

* Not available
<table>
<thead>
<tr>
<th>Number</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>*26</td>
<td>W.A. Walker and C.C. Gotlieb</td>
<td>A Top Down Algorithm for Constructing Nearly-Optimal Lexographic Trees</td>
</tr>
<tr>
<td>*27</td>
<td>D. Tsichritzis</td>
<td>Iff Programs</td>
</tr>
<tr>
<td>*28</td>
<td>L.W. Jackson</td>
<td>Automatic Error Analysis for the Solution of Ordinary Differential Equations</td>
</tr>
<tr>
<td>*30</td>
<td>Stanley Cabay</td>
<td>Second-Order Hyperbolic Equations with Data on Two Intersecting Boundaries</td>
</tr>
<tr>
<td>*31</td>
<td>Douglas Dale Olesky</td>
<td>Inclusion Regions for Partitioned Matrices</td>
</tr>
<tr>
<td>*32</td>
<td>J. Ian Munro</td>
<td>Some Results in the Study of Algorithms</td>
</tr>
<tr>
<td>*33</td>
<td>D. Tsichritzis</td>
<td>Modular System Description</td>
</tr>
<tr>
<td>*34</td>
<td>J. Mylopoulos</td>
<td>On the Relation of Graph Automata and Graph Grammars</td>
</tr>
<tr>
<td>*35</td>
<td>P.H. Roosen-Runge</td>
<td>The Algebra of Distributionally Defined Classifications</td>
</tr>
</tbody>
</table>

* Not available
<table>
<thead>
<tr>
<th>Number</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>Erold W. Hinds</td>
<td>Square Roots of an Orthogonal Matrix</td>
</tr>
<tr>
<td>*38</td>
<td>D. Tsichritzis</td>
<td>Topics in Operating Systems Revisited</td>
</tr>
<tr>
<td>*39</td>
<td>David G. Kirkpatrick</td>
<td>On the Additions Necessary to Compute Certain Functions</td>
</tr>
<tr>
<td>*40</td>
<td>P. Keast</td>
<td>Multi-Dimensional Quadrature Formulae</td>
</tr>
<tr>
<td>*41</td>
<td>Gordon D. Mulligan</td>
<td>Algorithms for Finding Cliques of a Graph</td>
</tr>
<tr>
<td>42</td>
<td>Stephen A. Cook and Robert A. Reckhow</td>
<td>Diagonal Theorems for Random Access Machines</td>
</tr>
<tr>
<td>*43</td>
<td>C.J.C. Lee</td>
<td>Translation of Context-Free Programming Languages Using Semantic Trees</td>
</tr>
<tr>
<td>*44</td>
<td>D. Tsichritzis</td>
<td>Lecture Notes on Operating Systems</td>
</tr>
<tr>
<td>*45</td>
<td>David Blair Coldrick</td>
<td>Methods for the Numerical Solution of Integral Equations of the Second Kind</td>
</tr>
<tr>
<td>*46</td>
<td>Wayne Enright</td>
<td>Studies in the Numerical Solution of Stiff Ordinary Differential Equations</td>
</tr>
<tr>
<td>*47</td>
<td>Barry Graham</td>
<td>An Algorithm to Determine the Chromatic Number of a Graph</td>
</tr>
<tr>
<td>*48</td>
<td>Philip A. Bernstein</td>
<td>Description Problems in the Modeling of Asynchronous Computer Systems</td>
</tr>
<tr>
<td>*49</td>
<td>Rudolf Mathon</td>
<td>On the Approximation of Elliptic Boundary Value Problems by Fundamental Solutions</td>
</tr>
</tbody>
</table>

* Not available
<table>
<thead>
<tr>
<th>Number</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>George Tourlakis</td>
<td>Some Results in Computational Topology</td>
</tr>
<tr>
<td>51</td>
<td>R. Michael Wharton</td>
<td>Grammatical Inference and Approximation</td>
</tr>
<tr>
<td>52</td>
<td>John Mylopoulos, Norm Badler, Lou Melli and Nicholas Roussopoulos</td>
<td>An Introduction to l.pak, A Programming Language for AI Applications</td>
</tr>
<tr>
<td>53</td>
<td>Arthur Sedgwick</td>
<td>An Effective Variable Order Variable Step Adams Method</td>
</tr>
<tr>
<td>54</td>
<td>Frank Wm. Tompa and C.C. Gotlieb</td>
<td>Choosing a Storage Schema</td>
</tr>
<tr>
<td>56</td>
<td>Keith O. Geddes</td>
<td>Algorithms for Analytic Approximation</td>
</tr>
<tr>
<td>57</td>
<td>Robert Moenck</td>
<td>Studies in Fast Algebraic Algorithms</td>
</tr>
<tr>
<td>58</td>
<td>Daniel Brand</td>
<td>Resolution and Equality in Theorem Proving</td>
</tr>
<tr>
<td>59</td>
<td>R.D. Tennent</td>
<td>Mathematical Semantics and Design of Programming Languages</td>
</tr>
</tbody>
</table>

* Not available
<table>
<thead>
<tr>
<th>Number</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>*61</td>
<td>C.C. Gotlieb</td>
<td>Data Types and Structures, A Synthetic Approach</td>
</tr>
<tr>
<td>62</td>
<td>C.A. Steele A.E. Sedgwick</td>
<td>DEPT: A Disciplined Extension of Fortran</td>
</tr>
<tr>
<td>*63</td>
<td>T.E. Hull J.J. Hofbauer</td>
<td>Language Facilities for Multiple Precision Floating Point Computation, with Examples and the Description of a Preprocessor</td>
</tr>
<tr>
<td>64</td>
<td>Philip R. Cohen</td>
<td>A Prototype Natural Language Understanding System</td>
</tr>
<tr>
<td>*65</td>
<td>D.G. Corneil</td>
<td>The Analysis of Graph Theoretical Algorithms</td>
</tr>
<tr>
<td>66</td>
<td>T.E. Hull W.H. Enright</td>
<td>A Structure for Programs that Solve Ordinary Differential Equations</td>
</tr>
<tr>
<td>67</td>
<td>Bengt Lindberg</td>
<td>Optimal Stepsize Sequences and Requirements for the Local Error for Methods for (Stiff) Differential Equations</td>
</tr>
<tr>
<td>70</td>
<td>C.C. Gotlieb A.L. Furtado</td>
<td>Data Schemata Based on Directed Graphs</td>
</tr>
<tr>
<td>71</td>
<td>T.L. de Carvalho</td>
<td>Some Results in Automatic Theorem-Proving with Applications in Elementary Set Theory and Topology.</td>
</tr>
<tr>
<td>72</td>
<td>Yiu-Sang Moon</td>
<td>Some Numerical Experiments on Number Theoretic Methods in the Approximation of Multi-dimensional Integrals.</td>
</tr>
</tbody>
</table>

* Not available
<table>
<thead>
<tr>
<th>Number</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>Lucio F. Melli</td>
<td>The 2.pak Language: Primitives for AI Applications.</td>
</tr>
<tr>
<td>74</td>
<td>David G. Kirkpatrick</td>
<td>Topics in the Complexity of Combinatorial Algorithms.</td>
</tr>
<tr>
<td>75</td>
<td>Gerald A. Gorelick</td>
<td>A Complete Axiomatic System for Proving Assertions about Recursive and Non-Recursive Programs</td>
</tr>
<tr>
<td>76</td>
<td>Richard B. Bunt</td>
<td>Self-Regulating Schedulers for Operating Systems</td>
</tr>
<tr>
<td>77</td>
<td>James F. Allen</td>
<td>A Prototype Speech Understanding System</td>
</tr>
<tr>
<td>78</td>
<td>Alexander T. Borgida</td>
<td>Topics in the Understanding of English Sentences by Computer</td>
</tr>
<tr>
<td>79</td>
<td>Stephen A. Cook</td>
<td>Axiomatic and Interpretive Semantics for an Algol Fragment</td>
</tr>
<tr>
<td>80</td>
<td>Norman I. Badler</td>
<td>Temporal Scene Analysis: Conceptual Descriptions of Object Movements</td>
</tr>
</tbody>
</table>