Printing Requisition / Graphic Services

15072

1. Please complete unshaded areas on form as applicable.
2. Distribute copies as follows: White and Yellow to Graphic Services. Retain Pink Copies for your records.
3. On completion of order the Yellow copy will be returned with the printed material.
4. Please direct enquiries, quoting requisition number and account number, to extension 3491.

TITLE OR DESCRIPTION: CS-87-67

DATE REQUISITIONED: Aug 16/88 | DATE REQUIRED: ASAP | ACCOUNT NO.: 1 2 6 6 1 4 4 1

REQUISITIONER - PRINT | PHONE: 2112 | SIGNING AUTHORITY: Sue Ledingham

MAILING INFO - NAME: DENGELIS | DEPT.: C.S. | BLDG. & ROOM NO.: DC 3114 | ☐ DELIVER ☐ PICK-UP

Copyright: I hereby agree to assume all responsibility and liability for any infringement of copyrights and/or patent rights which may arise from the processing of, and reproduction of, any of the materials herein requested. I further agree to indemnify and hold blameless the University of Waterloo from any liability which may arise from said processing or reproducing. I also acknowledge that materials processed as a result of this requisition are for educational use only.

NUMBER OF PAGES: 95 | NUMBER OF COPIES: 30

TYPE OF PAPER STOCK: ☑ BOND ☐ NCR ____ PT. ☑ COVER ☐ BRISTOL ☑ SUPPLIED ☐ _____

PAPER SIZE: ☑ 8½ x 11 ☐ 8½ x 14 ☐ 11 x 17 ☐ _____

PAPER COLOUR: ☑ WHITE ☐ _____ | INK: ☑ BLACK ☐ _____

PRINTING: ☐ 1 SIDE ____ PGS. ☑ 2 SIDES ____ PGS. | NUMBERING: FROM ____ TO ____

BINDING/FINISHING: 3 Hole punched inside. ☑ COLLATING ☑ STAPLING ☐ ____ HOLE PUNCHED ☐ PLASTIC RING

FOLDING/PADDING | CUTTING SIZE

Special Instructions

Next master
Copies required.

COPY CENTRE — OPER. NO. | BLDG. | MACH. NO.

DESIGN & PASTE-UP — OPER. NO. | TIME | LABOUR CODE

			D 0 1
			D 0 1
			D 0 1

TYPESETTING — QUANTITY

P A P 0 0 0 0 0				T 0 1
P A P 0 0 0 0 0				T 0 1
P A P 0 0 0 0 0				T 0 1

PROOF

P R F				
P R F				
P R F				

NEGATIVES — QUANTITY | OPER. NO. | TIME | LABOUR CODE

F L M				C 0 1
F L M				C 0 1
F L M				C 0 1
F L M				C 0 1
F L M				C 0 1

PMT

P M T				C 0 1
P M T				C 0 1
P M T				C 0 1

PLATES

P L T				P 0 1
P L T				P 0 1
P L T				P 0 1

STOCK

				0 0 1
				0 0 1
				0 0 1
				0 0 1

BINDERY

R N G				B 0 1
R N G				B 0 1
R N G				B 0 1
M I S 0 0 0 0 0				B 0 1

OUTSIDE SERVICES

$ _____ COST

TAXES - PROVINCIAL ☐ FEDERAL ☐ GRAPHIC SERV. OCT.88 4M-2

Printing Requisition/Graphic Services

4183

1. Please complete unshaded areas on form as applicable.
2. Distribute copies as follows: White and Yellow to Graphic Services. Retain Pink Copies for your records.
3. On completion of order the Yellow copy will be returned with the printed material.
4. Please direct enquiries, quoting requisition number and account number, to extension 3451.

TITLE OR DESCRIPTION: CS-87-67

DATE REQUISITIONED: Dec. 4, 1987 | DATE REQUIRED: ASAP | ACCOUNT NO.: 1 2 6 6 6 1 4 4 1

REQUISITIONER- PRINT: S. DeAngelis | PHONE: 2192 | SIGNING AUTHORITY: [signature]

MAILING INFO – NAME: S. DeAngelis | DEPT.: CS | BLDG. & ROOM NO.: MC 6081E | DELIVER / PICK UP

Copyright: I hereby agree to assume all responsibility and liability for any infringement of copyrights and/or patent rights which may arise from the processing of, and reproduction of, any of the materials herein requested. I further agree to indemnify and hold blameless the University of Waterloo from any liability which may arise from said processing or reproducing. I also acknowledge that materials processed as a result of this requisition are for educational use only.

NUMBER OF PAGES: 73 | NUMBER OF COPIES: 50

TYPE OF PAPER STOCK: [x] BOND [ ] NCR ___ PT. [x] COVER [ ] BRISTOL [x] SUPPLIED [ ]

PAPER SIZE: [x] 8½ x 11 [ ] 8½ x 14 [ ] 11 x 17 [ ]

PAPER COLOUR: [x] WHITE [ ] | INK: [x] BLACK [ ]

PRINTING: [ ] 1 SIDE ___ PGS. [x] 2 SIDES ___ PGS. | NUMBERING FROM ___ TO ___

BINDING/FINISHING: [x] COLLATING [x] STAPLING [ ] ___ HOLE PUNCHED [ ] PLASTIC RING

FOLDING/PADDING | CUTTING SIZE

Special Instructions

Math fronts & backs enclosed.

COPY CENTRE — OPER. NO. | BLDG. | MACH. NO.

DESIGN & PASTE-UP — OPER. NO. | TIME | LABOUR CODE: D01, D01, D01

TYPESETTING — QUANTITY

	QUANTITY	OPER. NO.	TIME	CODE
PAP000000				T01
PAP000000				T01
PAP000000				T01

PROOF

PRF				
PRF				
PRF				

NEGATIVES

	QUANTITY	OPER. NO.	TIME	LABOUR CODE
FLM				C01
FLM				C01
FLM				C01
FLM				C01
FLM				C01

PMT

PMT				C01
PMT				C01
PMT				C01

PLATES

PLT				P01
PLT				P01
PLT				P01

STOCK

				001
				001
				001
				001

BINDERY

RNG				B01
RNG				B01
RNG				B01
MIS00000				B01

OUTSIDE SERVICES

$ ______ COST

TAXES — PROVINCIAL [ ] FEDERAL [ ]

GRAPHIC SERV. OCT. 85 482-2

Perceptual Reasoning: A Logical Foundation for Computer Vision

J. D. Denis Gagné
Department of Computer Science

Research Report CS-87-67
December 1987

Perceptual Reasoning: A Logical Foundation for Computer Vision

by

J. D. Denis Gagné

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, 1987

Abstract

The role of model-based computer vision is to provide an interpretation of an image based on symbolic knowledge of a domain of interest. We clarify the meaning of this notion of interpretation by viewing visual recognition by a computer as theory formation.

We provide a declarative semantics which defines what constitute valid interpretations of an image, and provides a basis for ranking these interpretations. This meaning of an interpretation is independent of implementation, but we also show how this semantics can be combined with the typical representation used for visual knowledge to define a model-based vision system. More specifically, we show how valid interpretations can be constructed from a simple knowledge representation that uses a particular form of composition and specialization hierarchies.

Our specification of an image interpretation clarifies several aspects of model-based visual recognition. We compare it to related work in which a cycle of hypothesize-test-revise is used to iterate toward a preferred interpretation.

Contents

1 Computer Vision: A Brief Introduction **1**

1.1 Human Perception and Computer Vision 2

1.1.1 What do Computers See? 3

1.1.2 Low Level and High Level Computer Vision 4

1.2 Visual Recognition by Computers 5

1.3 Motivation 6

1.4 Outline of the Dissertation 9

2 Control Structures of Computational Vision **10**

2.1 Existing Computer Vision Control Structures 10

2.1.1 Three Paradigms 11

2.1.2 Segmentation and Interpretation 11

2.2 Model-based Approaches to Computational Vision 13

2.3 The Cycle of Perception 14

2.4 Theory Formation: An Intuitive Abstraction 14

3 A Semantics for Visual Recognition **18**

3.1 High Level Scene Analysis 18

3.2 Perceptual Reasoning: Visual Recognition as Theory Formation . . . 19

3.2.1 Using Logic for a Specification 20

3.2.2 The *symbolize* Relation 21

3.2.3 Visual Knowledge Representation 27

3.2.4 Coherent Interpretations 28

3.2.5 Preferring Interpretations 33

3.3 Summary 37

4 An Instance of Perceptual Reasoning: Exploiting Composition and Specialization **39**

4.1 Hierarchical Representation of the Visual Knowledge 40

4.2 A Theory of Schema Labelling 41

4.3 Axiomatization of a Domain 42

4.3.1 Axiomatization Methodology 43

4.3.2 Composition Axioms 44

4.3.3 Specialization Axioms 49

4.4 A Recognition Process 55

4.4.1 Exploiting the Visual Knowledge 57

4.4.2 The Generalization Phase 59

4.4.3 The Composition Phase 60

4.4.4 The Specialization Phase 63

4.5 Summary and Example 64

4.6 Correctness of the Recognition Process 72

5 Conclusion **75**

5.1 Summary and Contribution 76

5.1.1 Summary 76

5.1.2 Contribution 78

5.2 Future Research 79

List of Figures

2.1 The Cycle of Perception 15

3.1 Primitives 24

3.2 Graphical abstract 26

3.3 **T** junctions as evidence of occlusion 30

3.4 An abstraction of Perceptual Reasoning 38

4.1 Composition and Specialization Axes 45

4.2 Composition Hierarchies of the Train Domain 47

4.3 Specialization Hierarchies of the Train Domain 51

4.4 The $\rhd_{train}$ Relation for the Train Domain 56

4.5 Abstract View of a Recognition Process 65

Chapter 1

Computer Vision: A Brief Introduction

Once the main stream of a field of research is laid down, and researchers start following the current, long awaited answers rapidly arise and solutions to central issues become clearer as more complex concepts are made comprehensible. Computational vision has yet to have its main stream formally defined, but as our study of computational vision matures and the pieces of the puzzle start falling into place, central concepts that will become corner stones of further development are arising from a consensus of researchers of the field.

Our aim in this dissertation is to formalize, into a logical framework, the governing paradigms of up-to-date research in model-based approaches to computational vision. This formalism provides a precise and clear semantics for the different concepts used in computational vision. The semantics naturally follows from what we call *Perceptual Reasoning*[1] which introduces the idea of using "theory formation"

[1]The term is borrowed from Hayes [Hay81].

as an abstraction of the visual recognition process in a computational setting.

In this introductory chapter, we briefly introduce the reader to the kind of formats the sensed data take in a computational vision environment, and lay down some common goals as general basis for research in the field of computational vision. An overview of the rest of the dissertation is provided at the end of the chapter.

1.1 Human Perception and Computer Vision

Visual perception serves as a window on the world; it is a link between what is in the world, and the concepts we reason with. Due to the ease with which we achieve visual perception, our ability to visually perceive the surrounding world is something that we take for granted in our daily life.

Philosophers since ancient Greece, and more recently, contemporary psychologists, have tried to explain this complex phenomenon of visual perception. The quest for an explanation of our visual abilities is still on today. Numerous schools of thought have come about to explain visual perception, each having their own version of the "*explanation*" for human perception. Amongst the noticeable ones are those of Helmholtz, Gibson, the sensationalists, and the Gestaltists [Gar86].

With the introduction of computers came a new generation of scientists concerned with visual perception. These scientists belonging to new fields of research, such as *Artificial Intelligence* (**AI**) and *Cognitive Science* (**Cog-Sci**) started to investigate the possibility of defining computational models to achieve visual perception.

Computers provide us with an environment for verifying our theories about perception, and serve as a development tool for new ones. It is hoped by the

Cog-Sci community, that by investigating how we can get the machine to perform different tasks, we will gain insight on how we, ourselves, perform those same tasks. The AI community, on the other hand, is interested in computational models that are based on formal theories, but these theories are not necessarily representative of the way human achieve visual perception.

1.1.1 What do Computers See?

The stimuli (included in images) that are made available to computers for visual perception come in many different forms. Usually images used by computer vision systems come in the form of what we call digital (discrete) images. A *digital image* is an image sampled into digital values that approximates the brightness and location of the sensed data.

In monochrome digital images, the gray level value of the partitioned (sampled) image is assigned to a small cell called a pixel[2] . A pixel is the smallest unit of the image, and can be thought of as abstracting the *receptors*, sensory nerve cells that compose the *retina* in the human apparatus. These images can be represented by an image function, $\mathbf{f}$, where $\mathbf{f}(x, y)$ is the brightness of the gray level of the image at a spatial coordinate (x, y) [BB82]. Note that $\mathbf{f}$, x, and y only take on discrete values. It is this discrete information that the computer has to process to perform visual perception.

Even though this format is widely used, there exist other techniques relevant to computer vision that are used to encode the sensed data of the surrounding environments:

- Colour images

[2] "Pixel" stands for "picture element."

- Stereo images
- Range images

⋮

Colour Images are multispectral images where, for example, the intensity of the three wavelength: red, blue, and green are registered. In this case, the image function f is a vector-valued function with components $\{f_{red}, f_{blue}, f_{green}\}$ [BB82].

The image data in *range images* is obtained, for example, through the use of a laser range finder[3] , the depth information of the scanned surfaces of the scene is then registered (e.g., [LWR85,HYI86]). We can extract the depth information, and other spatial relationships in *stereo images* by examining the disparity between the two image planes (e.g., [Hof86,CF82]).

1.1.2 Low Level and High Level Computer Vision

Computer vision can be roughly separated into two levels of processing: low level and high level. Even though the two levels are often used to classify different tasks in computer vision, the terms remain ill-defined in terms of their scope.

We believe that these terms (low level and high level vision) refer only to the extremities of a "visual processing continuum", we will therefore define them here very generally, avoiding the overlapping section of the two levels.

Low level vision consists of the early processing of the data (e.g., filtering, edge enhancement, range finding). Through the early processing, we try to expand the information compressed in the pixels using their spatial relationships, and other properties found amongst them. For example, the goal of *filtering* is to enhance the

[3]Other devices using light or sonar are also used.

features included in the image by changing the registered gray levels of the image. *Edge enhancement* exploits the relation between local discontinuities in the gray level intensity and object boundaries to mark edges.

At the other end of this visual processing continuum, *high level vision* is more concerned with the aspect of the cognitive use of some encoded knowledge about objects and relations. The different tasks of high level vision are highly dependent on the computational model used. The general goal is to produce a certain description of the *scene* from the *image*; the description required depends on the desired application.

1.2 Visual Recognition by Computers

In order to achieve visual recognition by a computer, a very large amount of information must be processed, and unfortunately, it is very poorly structured. Pixels, by themselves, give rise to highly ambiguous information. The ambiguity arises because the value of each pixel of a digital image could have been generated by many different physical entities[4] , and because the depth dimension is collapsed by the projection of three dimensional objects into a two dimensional image[5] [CF82].

We said earlier that "visual perception serves as a window on the world;" in a computational setting, it should provide a correspondence between the world and the model the computer has of the world. Providing this correspondence is a major goal of computational vision that we refer to as *computational visual recognition.* A very general definition of computational visual recognition can be stated in the following terms:

[4]Many factors contribute to the value of a pixel -light, surface material, texture, etc.

[5]Using the focal length and the properties of similar triangles, we can define a multitude of cartesian coordinates that could have produce the same value for any given pixel.

Computational visual recognition *is the process of finding a mapping between the world, as depicted in an image, and an internal model of that world which can be depicted by the same image.*

The above mapping can be obtained by processing the intrinsic information included in the image, and providing an "*interpretation*" of the image which makes explicit the description of the scene depicted. This explicit description of the scene becomes a prerequisite, in a multi-purpose system, for further tasks such as the manipulation of objects and further reasoning about the objects [BB82]. Note that, in accordance with this definition, the mapping process is independent of the domain specific knowledge of the system. Only the terms used in an interpretation depend on the system's knowledge of a particular domain.

1.3 Motivation

An image can be considered as a collection of appearances of objects from the world. It is generally accepted that going from the object to its appearance is a stable mapping, but that going from an appearance to the object admits many exceptions, simply because the image underconstrains the scene depicted. Simple computational brute force to find this mapping is pointless unless guided by a theory. What we seek may be thought of as a theory for inverting the physical process of image formation [CM84].

In order to define a sensible theory of computational visual recognition, we must be able to define what counts as an acceptable description of an image content. This explicit description of an image content is usually referred to, in the computer vision community, as an "interpretation" of the image. The formal foundations of

what constitutes an interpretation of an image and how it can be obtained have not received much attention in the computer vision community. We consider these formal foundations to be crucial to an understanding of visual recognition. We view as a major focus of computational vision to define clearly and precisely *what constitutes a valid interpretation of an image, and how it can be obtained.*

In Marr's view [Mar78,Mar82], when designing a computational model one should distinguish between

- Computational Theory
- Algorithmic Considerations
- Implementation

The *computational theory* is concerned with the "*what*" and "*why*" of things being computed. *Algorithmic considerations* describe "*how*" computation is to be carried at an abstract level, and the *implementation* level specifies "*how truly,*" in terms of a concrete implementation, the specifications of the previous level can be carried out, or closely approximated.

In accordance with Marr's view, our goal in this dissertation is to define a "computational theory" for *Model-Based* approaches to computational vision and to present "algorithmic considerations" for a particular instance of that theory. Our emphasis is on higher levels of the computational recognition process, sometimes called "*high level scene analysis.*"

There has been much fundamental research done in the lower level of computational vision, which we will refer to only when indicating how higher levels of processing can be used to expedite lower level tasks.

The thesis we support is that the meaning of symbolic interpretation of an image is clarified by viewing computational visual recognition as "theory formation." We henceforth view the computational process of obtaining an interpretation of an image as a theory formation process. The goal of this theory formation process will be to explain, with theories built from a *fixed* set of possible hypotheses, the the observations made in the images, and to use these theories to make further predictions about the image. The fixed set of hypotheses contain the building blocks of the internal model of the world.

By viewing computational visual recognition as theory formation, we obtain a declarative semantics which clearly and precisely defines the concept of an image interpretation. This meaning of an interpretation for an image is independent of the implementation of this process.

The need for having a clear and precise definition of what constitutes a valid interpretation of an image, and a specification of how such interpretations can be obtained, is further justified by the following two points.

First, this semantics provides a framework to evaluate the *correctness* of implementations of model-based systems in computational vision. With a clear and precise definition of what constitutes a valid interpretation of an image, we can evaluate a particular implementation with respect to the criteria of *completeness* and *soundness.*

Second, the meta-theory of how such a valid interpretation is obtained can serve as a specification for new and possibly more efficient implementations. The theory does not equal the implementation as it is the case in many other approaches

Note that this dissertation makes no claims about human perception and is really only concerned with defining formalisms for model-based approaches to com-

putational vision.

1.4 Outline of the Dissertation

In the following chapter, we examine existing control structures used in computational vision. We are particularly interested in abstracting the current trends in model-based approaches to computational vision and synthesizing them into a theory formation process that we call Perceptual Reasoning. We explain informally how computing interpretations of images naturally follow from this abstract process.

In chapter 3, we provide a declarative semantics which defines a valid interpretation, and provides basis for ranking interpretations. This semantics (meaning) of an interpretation is independent of implementation, and can therefore be used as a reference for the correctness evaluation of model-based systems or as a guide for new implementations.

Chapter 4 demonstrates how Perceptual Reasoning can be used to define a computational vision system. We first discuss the use of hierarchies of abstraction for the representation of the visual knowledge as it is used in most model-based approaches, and then present a particular instance of Perceptual Reasoning that exploits composition and specialization hierarchies for its visual knowledge representation. Full examples of the recognition process, and arguments of the correctness of this particular instance of Perceptual Reasoning are provided.

Finally, we conclude in chapter 5 with a summary of our formalism and our contribution, and indicate directions for further research.

Chapter 2

Control Structures of Computational Vision

The aim of this chapter is to examine existing control structures used in computational vision. In particular, we abstract the current trends in model-based computational vision and synthesize them into one coherent framework. We introduce *Perceptual Reasoning* as this abstraction of the visual recognition process in a model-based computational setting. Perceptual reasoning is based on the idea of using "theory formation" as the basis for providing interpretations of images.

2.1 Existing Computer Vision Control Structures

Despite all the past research in computer vision, there is still disagreement on very central issues of what constitutes an appropriate approach to visual recognition by a computer. One such issue is the control structure used for computational vision, which is at the very heart of any vision system. The different control structures

that exist can, at a high level of abstraction, be classified into three very general paradigms: the bottom-up approach, the top-down approach, and the hypothesize-and-test approach.

2.1.1 Three Paradigms

The bottom-up approach, also known as the data driven approach, can be recognized in the pioneering work of Marr [Mar78,Mar82], and Barrow and Tenenbaum [BT78,BT80]. This approach is characterized by a data driven process; the initial level is computed directly from the image and recognition is achieved by incrementally processing the available information upward. Its counterpart, the top-down approach, usually relies on "surface shape" or "wire frame" models and proceeds by trying to identify a restricted set of objects in the given image by going from the model to the image. This approach, however restrictive because of the many appearances the objects can take from different viewpoints, is behind several successful industrial vision systems.

The third control structure alternates between bottom-up and top-down processes; it is the hypothesize-and-test approach. This is the approach taken in so-called *Model-Based* vision systems (e.g., *ACRONYM* [Bro81]). We believe all these systems include the basis of the ideal general solution to the control issue, that is, that they alternate between data- and model-driven processes. In the next section, we redefine this notion in terms of segmentation and interpretation.

2.1.2 Segmentation and Interpretation

Segmentation and interpretation are often seen as the major steps in computer vision recognition. By *segmentation*, we mean the location and extraction of coherent

spatial groups in the image, such as edges, surfaces, background/foreground, etc., and by *interpretation*, we refer to the transformation of the image description, in terms of the segmentation, into a description of the *scene* which conveys a meaningful organization of what is really out there.

To apply these two steps sequentially assumes that a perfect segmentation can be achieved. It is known that only partial segmentation can be achieved at first, unless you make very strong and unverifiable assumptions. We believe, like Mackworth [Mac78], that these two problems (segmentation and interpretation) are not independent, and further believe that contextual knowledge is the key to an adequate segmentation.

In a visual framework, it can easily be shown that contextual information from the complete image plays an important role in how we interpret what we see. A piece of image out of its context carries little information about its surroundings. To convince yourself, simply look through a tube at the scene around you, or place a piece of paper with a narrow slit cut in it over a picture [Pen86,Roc83]. It then seems evident that exploiting contextual information from an image is essential for visual recognition by a computer. Although contextual information plays an important role in visual recognition by a computer, one must realize that the entire picture itself serves as a tube or slit on the surrounding world, meaning that context is always partial. There can be more or less of the contextual information present but never all. Therefore the potential for ambiguity is always present, and a maximal use of the "available" contextual information is what we should aim for.

In order to exploit contextual knowledge during segmentation, we believe it is desirable to alternate between careful segmentation and interpretation, as was suggested by Mackworth [Mac78].

2.2 Model-based Approaches to Computational Vision

Contextual knowledge of an image arises when the recognition of one object (or feature) in the image leads us to expect other certain objects (or features). It is generally accepted that image understanding is almost impossible without such expectations, but some formalisms do not explicitly exploit that information. We believe it is important to use such information as it will reduce the search space when interpreting the image. We further believe that the use of this contextual information should be part of the specification of a computational vision framework, as opposed to being in the form of embedded heuristics, because this will help clarify the early processing.

This idea of using such contextual information is the underlying concept behind model-based approaches to computational vision. This explains the particular attention given to model-based approaches of computational vision in this dissertation.

In our effort to abstract and synthesize the visual recognition process used in model-based approaches to computational vision, we examined many different systems, trying to develop insight into how the analysis of an image is controlled. Amongst the many systems studied were the *Aerial Photographs Analysis System* [MNI78,MNI79], *VISIONS* [HR78], *ACRONYM* [Bro81,Bro84,Bro86], *MAPSEE* [Mac77,HM83], *ALVEN* [Tso85], the *Model-Based System* of Goad [Goa86], and others.

These systems' control structures share many similarities. Mackworth [Mac78] already noticed that a paradigm for the control structure of vision programs can be

characterized in what he defined as the *Cycle of Perception.* He further explained how different vision programs can be characterized by the way they treat the Cycle of Perception.

2.3 The Cycle of Perception

The Cycle of Perception, as proposed by Mackworth [Mac78], consists of four processes: cue discovery, model invocation, model verification, and model elaboration. See figure 2.1.

Mulder [Mul85] gives an elaboration of the respective roles of these processes. Briefly, the *cue discovery* process is equivalent to segmentation, the *model invocation* process associates possible interpretations with the elements of the cue discovery process. *Model verification* is the process that tests whether the description of the model associated with the cues is consistent with the image, finally, the *model elaboration* process ensures, using a constraint satisfaction algorithm, that the constraints produced by the set of all models are jointly satisfied and do not contradict one another.

What is not explicit in Mackworth's Cycle of Perception is a precise specification of *what constitutes a valid interpretation*, i.e. when does the cycle end.

2.4 Theory Formation: An Intuitive Abstraction

The basic concept underlying all these model-based systems that Mackworth characterized with his Cycle of Perception is this idea of hypothesizing the presence of

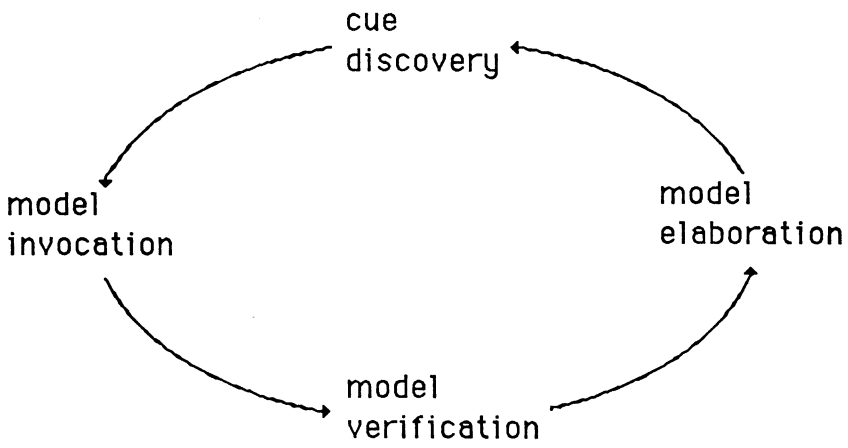


Figure 2.1: The Cycle of Perception

some objects based on cues extracted from the image, and then verifying that these objects are indeed present in the image.

Intuitively, this way of proceeding with the analysis of images is very similar to the way scientists build theories to explain natural and physical phenomena. Informally, having observations of certain phenomenon, a scientist proposes a theory to explain this phenomenon based on related hypothesis and facts. He will then use this theory to explain new observations and to make predictions of further phenomena. In the event that the predictions from his theory are not observable or even contradicted, the scientist will revise his hypotheses, and modify his theory so that it is more representative of the phenomena.

Analogically, we can view the features (cues) extracted from the image as a set of observations to be explained by a theory. We want to use as theories, only consistent sets of objects that explain the presence of the features observed in the image. This criterion for forming theories already considerably reduces the search space for computational visual recognition, yet, many such theories can be found for a given set of observed features. The process of finding theories that explain the observed features characterize what are called "hypotheses activation schemes" in the computer vision community.

Given the set of theories that explain the observations, we want to first consider those that are most likely to be coherent with the image. An attempt to rank the theories is then in order to "focus" our attention on a preferred theory. We then want to verify the "expectations" of this theory. If the expectations of the present theory are coherent with the image, we then use this theory as an "interpretation" of the image. If it turns out that the expectations of the present theory are not coherent with the image, we then focus on the next preferred theory and so on until an interpretation is found or we run out of theories, in which case, it is safe to assert

that no interpretation of the image can be found given the knowledge of the system.

This "theory formation" abstraction of reasoning was inspired by Popper's view of scientific discovery [Pop58], and is basically an abductive form of reasoning [Hem65]. Quine and Ullian [QU78] also favour this kind of theory formation approach as an explanation of human reasoning.

The foundations of the approach we present in this dissertation were inspired by the *THEORIST* project at the University of Waterloo [PGA86]. Theory formation from a fixed set of possible hypotheses has also been shown to include part of human common sense way of doing reasoning in many tasks, e.g., default reasoning [PGA86,GFP86,Poo86], analogical reasoning [Jac86], and planning [Goo87,GG87].

The following chapter defines Perceptual Reasoning, which is a formal characterization of "theory formation" appropriate for describing the way model-based vision systems obtain interpretations of images. From the formal treatment of these ideas arises a semantics for image interpretations.

Chapter 3

A Semantics for Visual Recognition

In this chapter, we view computational visual recognition as theory formation, provide a declarative semantics which defines a valid interpretation, and provide a basis for ranking interpretations. We clarify the meaning of a symbolic interpretation by formalizing model-based computational vision in terms of a theory formation characterization founded on first order logic. This formalism, which we call Perceptual Reasoning, provides logically founded criteria to discriminate between valid and non-valid interpretations, and for preferring one interpretation to another.

3.1 High Level Scene Analysis

We restrict ourselves to the the problem of *High Level Scene Analysis*[1] of static images. We define the problem of high level scene analysis as the process of mapping

[1]The term "image understanding" is also often used.

the *image domain* to the *scene domain.* This distinction between the image and scene domains in the cycle of perception is due to Kanade [Kan78]. We view the image domain as a projection, depicting the scene domain from one fixed viewpoint.

When referring to the image domain, we speak of cues or features that can be extracted from the image (e.g., lines, regions, etc), or of derived cues and features, (e.g., sets of lines and regions), which are directly derived from the image. In the scene domain, we refer to sets of symbols[2] denoting scene objects (e.g., car, wheel, etc.), and to relationships amongst those symbols.

The mapping process between the image domain and the scene domain for static images correspond to a restricted form of computational visual recognition as defined earlier.

3.2 Perceptual Reasoning: Visual Recognition as Theory Formation

A description of a scene depicted by an image is obtained by providing an interpretation of the image in the scene domain. As we observed in the previous chapter, this notion of interpretation is intuitively similar to a scientific theory: a possible explanation of the observations made in the image domain.

Informally, in Perceptual Reasoning we propose theories of scene domain objects that could have generated the appearances that we observe in the image domain. For a set of initial observations from the image, hypotheses are selected from a fixed

[2]These symbols are names of scene domain objects. We use the word symbol here, because of the way these symbols are used to categorize the features from the image. The term "label" is also often used.

set of possible hypotheses to try to explain the observations made[3] . An ordering of the selected hypotheses is then attempted, and the hypotheses are verified, one by one, for coherence with the image. As soon as an hypothesis with all the desired properties of an interpretation is found, we *commit* to this theory as being the interpretation of the given image which provides a description of the scene depicted.

The rest of this chapter will formalize this theory formation characterization of model-based approaches to computational vision.

3.2.1 Using Logic for a Specification

First order logic is well known for its clear and precise semantics and has been shown to be descriptively adequate for many tasks [Hay77,Moo82]. We chose first order logic as the language for the specification of our characterization of computational vision, because of the rigor it provides.

Traditional logic as been perceived as inappropriate for the kind of reasoning we are interested in, because of its monotonicity. That is, a common use of logic is to state our knowledge by making assertions of what is true in the world we are modelling, and then inferring what logically follows from our knowledge. When analyzing images, we do not only want to use deduction, we in addition want to test hypotheses, and to change our hypotheses of the content of an image as new information is obtained. Israel [Isr80] argued that rather then being a problem with logic, the problem was with the way we use logic. Poole [Poo86] further argued that by viewing reasoning as theory formation, we are able to overcome these apparent shortcomings of ordinary logic.

[3]The observations are cues and features of image domain and the hypotheses are scene domain objects, i.e. the possible things that could be in the scene.

The syntax we will use for our characterization is an extension of the syntax of first order predicate calculus. The following normal first order predicate calculus symbols will be used: variables start with upper case letters, constants, functions and predicates are in lower case. The connectives are the standard first order connectives: $\neg$ for negation, $\vee$ for disjunction, $\wedge$ for conjunction, $\supset$ for implication. Parentheses are used as well, and Prolog's convention is used for lists (e.g., $[a, b, c]$ is a 3 item list). All new additions to this syntax will be explain as we introduce them.

Before going into the details of Perceptual Reasoning, we have to formalize and provide definitions for different concepts that will help put Perceptual Reasoning into perspective.

3.2.2 The *symbolize* Relation

We first need a formal way to relate the cues or features of the image domain to the scene domain objects they can depict. Intuitively, we would like to preserve flexibility *vis a vis* the image analysis/synthesis paradigms [Won86] because of the close relationship that exists between these two paradigms. Computer vision has been concerned with the *analysis* of images, that is, to produce a description of what can be "*seen*" in the image. On the other hand *synthesis* belongs to the field of computer graphics. It refers to "*illustrating*" graphically, in an image, the description of a scene.

We provide the desired relation and flexibility by using a two place predicate, *symbolize*[4] , as the basis of our formalism. Intuitively, we want *symbolize* to relate names of objects to the actual things (instances) that can be observed in the image.

[4]**symbolize** is defined in [Oxf79] as a *verb* (i): to be the symbol of (ii): to represent by mean of a symbol.

To achieve this, two domains are defined. The first one consists of the individual, or composite individual (i.e. individuals regrouped into one) things that are actually in the image (e.g., line, region, etc.). We refer to these individuals and composite individuals as the "*cues*" from the image. The second domain is a set of "*symbols.*" These symbols are the names of objects that could be in the scene depicted by the image (e.g., car, bicycle, etc.). For Perceptual Reasoning, *symbolize* is a relation between these two domains. Therefore $symbolize(X, Y)$ is true whenever the *Symbol* X symbolizes (names) a *Cue* Y of an image.

The *symbolize* relation, in Perceptual Reasoning, is used as a basis for a kind of rewriting framework such that a mapping from the image domain to the scene domain is obtained. During this rewriting, the individual cues are regrouped to form composite individual cues. The rewriting is done according to axioms that are written about the *symbolize* relation. The grouping of cues (features) to form a higher level cue, can be perceive as a form of segmentation, as it reflexes the association of cues (features) from the image into coherent spatial groups.

Definition 1: *The* **primitives** *are a distinguished set of symbols from the domain of symbols.*

Primitives denote the finest grains of Perceptual Reasoning and are used in both observations and predictions. We assume that a procedural attachment exists to define the *symbolize* relation between a primitive and a cue from the image.

Definition 2: *An* **instance of a primitive** *is a ground instance of the symbolize predicate where the first argument is a primitive and the second argument refer to a particular cue in the image.*

The symbols used in the initial observations will form the start set of the recognition process. In general, these symbols are going to be primitives. One possible exception to this general rule is image-specific observations directly inputed by the user.

We present a simple example to clarify these concepts in figure 3.1. We first list the primitives for the example. We then define the *symbolize* relation for the primitives (we used small icons for the second arguments of the *symbolize* relation to illustrate the fact that a procedural attachment would be used to define the relation). An example of a particular image is then given along with the instances of primitives that form the observations for this image.

One can recognize in the *symbolize* relation the preservation of the analysis/synthesis flexibility mentioned above. In the synthesis context, procedures for "rendering" the cues or feature would be used to define the *symbolize* relation for the primitives as opposed to procedures for detecting them.

We now introduce the concept of an *Intermediate Image.*

Definition 3: *An* **intermediate image** *is a collection of assertions made from and about the digital image. It consists of instances of* $symbolize(x, y)$*, where* x *is a primitive, and of other detected relations between the cues.*

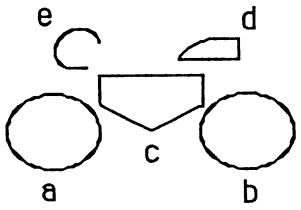
An intermediate image consists of two parts; there is an explicit and implicit part. The *explicit* part of an intermediate image contains the actual *observations*; cues that are found initially in the image. Whenever we attempt to establish the *symbolize* relation in the image to verify a *prediction*, we are dealing with the *implicit* part of an intermediate image.

For perceptual reasoning purposes, the intermediate image is initially going to provide observations to be explained by a theory, as instances of the *symbolize*

primitives = { wheel, frame, seat, handles }

Definition of symbolize for primitives
(procedural attachment)

symbolize(wheel, ○)
symbolize(frame, ⌓)
symbolize(handles, ⊂)
symbolize(seat, ▱)



Instances of Primitives

symbolize(wheel, a)
symbolize(wheel, b)
symbolize(frame, c)
symbolize(seat, d)
symbolize(handles, e)

Figure 3.1: Primitives

predicate and other known relations. These "other relations" are relationships which may hold between cues (e.g., junction types between lines). The intermediate image, through the use of the *symbolize* predicate, is a powerful media because of the flexibility and abstraction it provides with respect to the digital image[5] .

The intermediate image can be seen as a communication node between the lower levels and the higher levels of recognition processing. Through the intermediate image, the lower levels provide observations to be explained by higher levels, and the higher levels make predictions to be verified by lower levels.

The intermediate image is independent of the knowledge source; the observations that form the intermediate image can come from many different sources[6] (e.g., stereo vision disparities, texture analysis, range data from sonar), even from the user (e.g., domain specific knowledge). All these sources can cooperate to build the intermediate image.

Two things are important to notice at this point. First, that the procedural attachment that defines the *symbolize* relation for a primitive can relate to different kind of cues or features depending on the implementation for a particular desired application (e.g., line drawings, volumes, etc.). Second, that no fixed point on the recognition continuum is specified by the formalism for the intermediate image, meaning that the level of abstraction at which the observations are made can vary from implementation to implementation (e.g., $symbolize(edge, a)$ or $symbolize(wheel, b)$). The location on the visual recognition continuum at which the intermediate image should be fixed will depend on efficiency and performance criteria of the overall system. The availability of powerful and highly efficient algorithms

[5]The concept of the intermediate image is similar to the primal and 2.5D sketches of Marr [Mar78,Mar82] or the intrinsic image set of Barrow and Tenenbaum [BT78,BT80] in being an intermediate version of the image, but is still different since symbols are related to the cues.

[6]See Glicksman [Gli82a,Gli82b,Gli83,Gli84], and Rubin [Rub81] for more on multiple sources of information in computational vision.

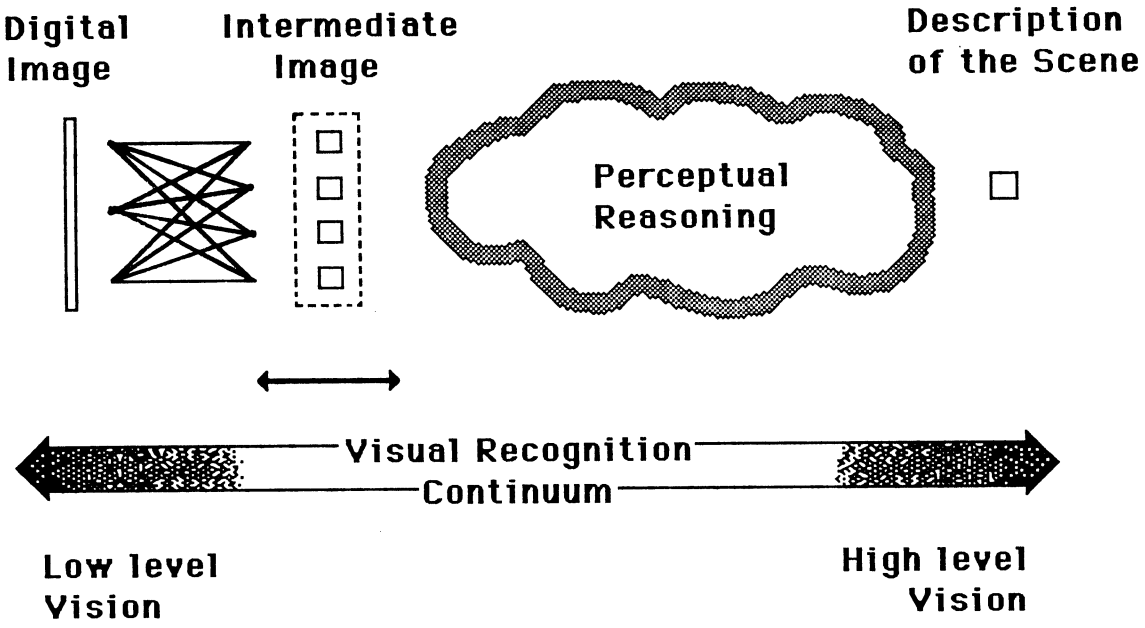


Figure 3.2: Graphical abstract

rithms for lower level tasks will, for example, influence the position of the intermediate image on the continuum.

Figure 3.2 presents a graphical abstraction of some of the ideas discussed to this point. Note that Perceptual Reasoning refers to the process of going from the intermediate image to a description of the scene.

3.2.3 Visual Knowledge Representation

In the proposed formalism, the visual knowledge representation is expressed in a logical language used to explicitly declare, through axioms, the knowledge we have about the "domain of interest" or "intended world". It has been argued that at least first order predicate calculus is needed in a representation that requires manipulation of individuals and relations [Hay77,Moo82].

Most model-based vision systems exploit hierarchical abstractions of prototypical objects in their representation of visual knowledge. This practice doesn't influence the formal meaning of an interpretation of an image, so it will be left out of this chapter, to be taken up in the following chapter when we discuss a particular instance of Perceptual Reasoning.

Within Perceptual Reasoning we distinguish three sets of formulae:

Γ : **The set of facts.** This set contains axioms about the *symbolize* predicate, and other facts. In particular, it contains assertions about the relations known to be always true of the domain of interest, between the scene domain and the image domain.

Δ : **The set of possible hypotheses.** The possible hypotheses are instances of the *symbolize* predicate denoting scene domain objects that we are ready to accept as explanations for the observations, and therefore as part of interpretations. They refer to the possible things that could be in the scene, the building blocks of the internal model of the world.

Obs : **The set of image observations.** It contains all the observations made from the image (e.g., instances of primitives). This set is the explicit part of the intermediate image.

3.2.4 Coherent Interpretations

We now need a formal definition of what it means to have a valid interpretation[7] for an image. With this formal definition it is possible to discard invalid interpretations and to tell when we found a valid one.

Before going into more details, let us look at some general intuitive properties that are desired of an interpretation for an image, independently of how this interpretation is acquired:

- We want an interpretation to somehow account for the observations made in the image.
- We want an interpretation to be sensible given the knowledge the system has of the domain of interest.
- We want an interpretation to predict only these things that are coherent with the image.

A more formal rewording of the above desired properties in terms of logic could be listed as:

1. An interpretation should logically imply the observations made from the image.
2. An interpretation should be consistent with the particular knowledge the system has of the domain of interest and the observations made from the image.
3. All the primitives entailed by an interpretation should be coherent with the image, meaning that for an entailed primitive either:

[7]Note that our use of the term "interpretation" is in accordance with its use in the computer vision community and should not be confused with its use in logic.

(a) we have already observed it in the image or

(b) it can be verified in the image or

(c) there is supporting evidence for it not being observable.

Items (a) and (b) above refer to the intermediate image in the following manner: item (a) -if the primitive has already been observed then it is in the explicit part of the intermediate image, item (b) -another possibility is that the primitive is in the implicit part of the intermediate image, in which case we can verify this prediction by trying to establish the *symbolize* relation for that primitive. Lets consider some of the possible reasons why some features predicted (entailed) by the interpretation may not be verifiable (observable) in the image, and the possible evidence supporting these reasons (item 3(c) above).

First, we may be dealing with objects that are partially occluded in the image. An object is *partially occluded* if it is partially hidden behind another object. If an object is occluded by another, then not all of its predicted parts and relations amongst them will be observable in the image. Evidence of partial occlusion may come from various hints; one of the best known is the presence of **T** junctions in the image [Low85]. See figure 3.3. This idea of using junction types as evidence of possible associations of regions into objects was first observed by Guzman [CF82].

Another possible reason for an expected primitive not to be observable could be (if using 3D models) that parts of an object are hidden because of the viewpoint (self-occlusion). In cases where scene objects are defined using 3D models, visibility of the primitive of an object can be decided based on the *framing knowledge* of the image. The framing knowledge of the image consists of such facts as the viewpoint, the perspective constants, lighting, etc. Note that in cases where the objects are defined using 2D line drawings where no self-occlusion is possible, all the parts of

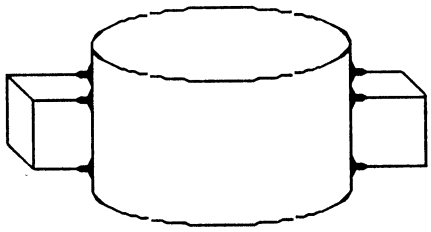


Figure 3.3: T junctions as evidence of occlusion

the objects are *a priori* expected to be visible since the viewpoint has no influence in these cases[8] .

In Perceptual Reasoning, our theory formation characterization of model-based approaches to visual recognition, we have grouped the desired properties of an interpretations in the following way:

Definition 4: *Given a set of facts* Γ *known to be true of the scene and image domains, a set* I *consisting of instances of the symbolize relation is said to be a* **theory** *that explains the explicit part of the intermediate image, Obs, if*

$$\Gamma \cup I \models Obs \text{ and}$$
$$\Gamma \cup I \text{ is consistent}^9$$

Definition 5: *Given* Γ*, a set facts known to be true of the scene and image domains, a theory* I *that explains the explicit part of the intermediate image, Obs, is* **coherent** *if for any* P*, where* P *is the symbolize relation involving a primitive:*

if $\Gamma \cup I \models P$ *then either:* $\Gamma \cup Obs \models P$

or

P *is verifiable in the image*

or

P *is accounted for.*

The conditions for a primitive P to be *accounted for*, will include potential reasons for which P has not been observed or cannot be verified in the image, and the required evidence for these reasons to be sensible. We informally discussed

[8]Occlusion from other objects is still possible.

[9]The term consistent here refers to logical consistency.

above some of these potential reasons, namely occlusion and self-occlusion because of the viewpoint, and indicated the evidence that could be required for them to be acceptable. It may also be the case that in a simpler instance of Perceptual Reasoning there is no need for such conditions[10] . In these cases the primitives must either have been observed or be verifiable in the image.

We do not here pursue a formal characterization of all the conditions for a primitive to be accounted for, as such an endeavor would require a very elaborate study of all possibilities. The two possibilities presented above indicates how complicated and wide this definition can become. The conditions specified in the above definition still provide an appropriate basis as they state needed conditions and leave room for completion. They then make perfect sense if only to make progress on the problem.

We can now define an interpretation in the following way:

Definition 6: *A theory I that explains the observations from an image (i.e., the explicit part of the intermediate image) is said to be an* **interpretation** *of this image if it is coherent.*

This clear and simple definition provides a semantics for obtaining a description of the scene through an interpretation of the image. This definition of an interpretation follows our definition of computational visual recognition, as it defines clearly and precisely what the correspondence between the observations of the image domain, and the objects of the scene domain means. Note that this definition is independent of the implementation and can therefore be used for both a characterization of model-based approaches to computational vision, and as a guide for an implementation.

[10] E.g.,systems that do not deal with 3D models or occlusion, for example MAPSEE [HM83].

It is important to realize that an interpretation will not necessarily be a single symbol for every possible complete image; objects can be recognized individually in unexpected contexts, and complex scenes by proposing an interpretation that is a conjunction of instances of different objects. A complete and accurate account of the content of an image can be obtained by computing closure under theoremhood of an interpretation of the image. What we mean by complete and accurate account of an image is the full breadth of both explicit and abstract scene and image domain objects that can be inferred about the image.

3.2.5 Preferring Interpretations

For a given image we can expect to find many interpretations with the above properties. We would now like to have some well founded criteria to select or prefer one interpretation over the others. Using our formal definition of what it means to have an interpretation of an image, we can provide logically founded criteria to define the concept of a *preferred* interpretation. This notion of preferred interpretations may vary from one application to the another. Our criteria of preference should then be generic ones, and we should allow the possibility of changing them so that Perceptual Reasoning can be tailored to a particular application.

One more or less universal criterion that seems to provide a basis for the preference of interpretations is the notion of specificity. For example, suppose that we have an image of a frame properly connected to two wheels, a seat, and curly handles. An intuitively desirable description might be that it is a scene of a racing bicycle. However, many interpretations for the given set of observations will satisfy our above definitions for interpretations, e.g., in this case, it is just as intuitively correct to say that the image depicts a bicycle. When preferring interpretations on

the basis of specificity, a racing bicycle would be preferred, although a bicycle is also a valid interpretation, because it is a more specific interpretation of the observations. Therefore, we say we are interested in the *most specific interpretation* for which we have evidence.

The intuition behind specificity is that the symbols used in a preferred interpretation refer to more specific classes of objects than those used in another interpretation. We define what it means for an interpretation to be *more specific* than another with the following semantics:

Definition 7: *Given the set* Γ *of facts known to be true of the image and scene domains. For a given image, we say that an interpretation* I_1 *is* **more specific** *than an interpretation* I_2 *if*

$$\Gamma \cup I_1 \models I_2$$

This relation over the interpretations of an image induces a partial ordering of interpretations.

Definition 8: *A* **most specific** *interpretation for which we have evidence is an interpretation that is a maximal element in the partial ordering of more specific interpretations.*

We are assuming that there is always such a maximal element, which means we assume that there are no infinite specializations. Note that there may be multiple incomparable most specific interpretations for an image, e.g., for the Necker cube. In these cases any one of the most specific interpretations can be used.

Now, for different applications, the most specific interpretations may not be the preferrable ones, some other preference criteria may be more suitable for a

particular application. For example, the vision recognition system could serve as the input source of another system to which the recognition of peculiar objects, or classes of objects, is important. In general, the interpretations are preferred on the basis of the symbols that are used in them. We will now define the general basis for preferring interpretations based on the symbols they use.

A binary relation defining a preorder[11] over the set of symbols is needed to formally define the notion of preference, we will use $\rhd$ to denote this preorder. An instance of Perceptual Reasoning can be tuned to a particular application by providing an alternative relation ($\rhd$) over the set of symbols that defines a tuned preordering of the symbols for the specific application.

We now can define how to *prefer* one interpretation over another, on the basis of this preordering of the symbols. Given the set of all interpretations for an image, we define the following binary relation over the interpretations.

Definition 9: *For a given image, we say that we* **prefer** *an interpretation I_1 to an interpretation I_2, written $I_1 \rho I_2$, if*

$$\forall\, symbolize(x,y) \in I_2,$$
$$\exists\, symbolize(u,v) \in I_1,\ s.t.$$
$$u \rhd x \ and \ v \supseteq y$$

The proper superset condition between v and y is introduced to insure that we are comparing elements of the interpretations for the same image features and that the largest possible set of features from the image are included as being of the same type.

[11] A preorder is a transitive and reflexive relation.

We see that ρ is a preorder over all the interpretations of an image. We can now define an equivalence relation over all the interpretations of an image as follow:

Definition 10: *For a given image, we say that an interpretation I_1 is* **equally specific** *to an interpretation I_2 written*

$$I_1 \equiv I_2 \leftrightarrow I_1 \rho I_2 \text{ and } I_2 \rho I_1$$

This equivalence relation partitions the interpretations of an image into equivalence classes of interpretations. We introduce the concept of equivalence classes of interpretations in order to obtain a partial ordering as the basis for preferring interpretations. This also provides us with the ability to qualify two interpretations of an image as equally specific. It is easy to demonstrate that, as a result of applying ρ over these equivalence classes, we obtain a partial ordering of equivalence classes of interpretations.

Definition 11: A preferred interpretation *for an image is an interpretation $I_i \in [I]$, where $[I]$ is an equivalence class of interpretations that is a maximal element of the partial ordering induce by ρ over the equivalence classes of interpretations for the image.*

Now, this ordering also applies to theories before they are checked for coherence and become interpretations. We should then rank the theories before verifying their coherence with the image. By doing so, the computational visual recognition search space is further reduced, since the first coherent theory that we will find will also be a preferred interpretation. This is a common practice in computational vision and is referred to as "focusing the attention".

3.3 Summary

We have provided a characterization, based on theory formation, of model-based approaches to computational vision. Hypotheses invocation schemes, found in computational vision, have been characterized as finding logically consistent theories that have as logical consequences the observations made from the image. The focusing of attention was characterized as selecting a maximal element from a partial ordering as being the most promising theory to pursue. The generation of expectations and their verification was characterized as ensuring coherence of the theories with the image.

This characterization of model-based computational vision provides us a simple and clear semantics for interpreting images that is independent of implementation. The Cycle of Perception, discussed in chapter 2, can be recast in this framework. The reformulation provides a motivation to the different processes of the Cycle of Perception, and provides explicit definition of the criteria to be satisfied during the cycle and explicitly defines when we can stop the cycle. One important thing to notice is that the semantics is independent of the control strategy, but the hypothesize-and-test strategy follows naturally from theory formation.

Figure 3.4 provides an abstract representation of Perceptual Reasoning. Represented in this figure are the ideas that, from the set of possible hypotheses, Δ, theories, $\{I_1 \ldots I_n\}$, are found such that along with the facts, Γ, they logically imply the observations collected in the intermediate image. The theory preference criterion based on specificity or on a particular $\rhd$ provides a most promising theory, I_i, for which we verify coherence with the image. If coherent, this theory is then a most specific interpretation for which we have evidence, otherwise the next preferred theory is verified for coherence with the image.

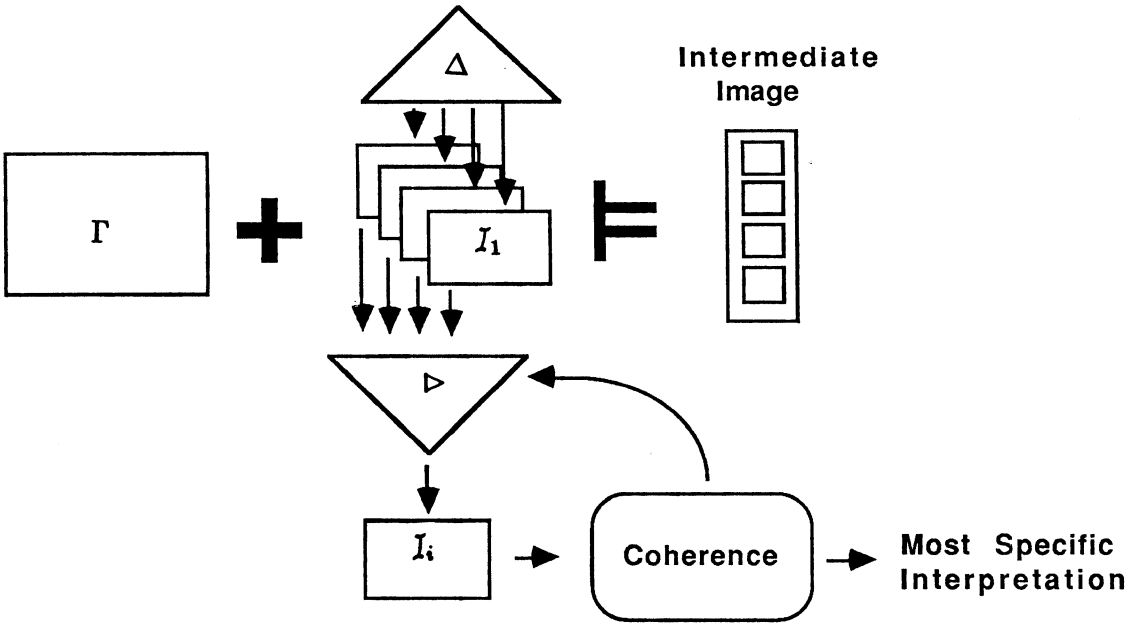


Figure 3.4: An abstraction of Perceptual Reasoning

Chapter 4

An Instance of Perceptual Reasoning: Exploiting Composition and Specialization

In this chapter, we first discuss the use of hierarchies of abstraction for the representation of visual knowledge in model-based approaches. We then show how the formal theory presented in the previous chapter can be used to define computational vision systems. The demonstration consists of defining a particular instance of Perceptual Reasoning that exploits composition and specialization hierarchies for the representation of visual knowledge. An example that demonstrates the particular recognition process, and arguments of correctness of this particular instance of Perceptual Reasoning are provided at the end of the chapter.

4.1 Hierarchical Representation of the Visual Knowledge

A concept that is basically agreed upon amongst the researchers in computational vision is that the interpretation of an image can more easily be obtained by a series of transformations of the image that incrementally provide more abstract descriptions of the content of the image. An adequate representation of the visual knowledge under these circumstances should provide for an organization of the knowledge into a hierarchy of incrementally more organized and abstract objects and scene descriptors[1] .

Most model-based vision systems exploit hierarchical abstractions of prototypical objects or scene descriptors in their representations of the visual knowledge, e.g., the *Aerial Photographs Analysis System* [MNI78,MNI79], *VISIONS* [HR78], *ACRONYM* [Bro81,Bro84,Bro86], *MAPSEE* [Mac77,HM83], *ALVEN* [Tso85], and many others.

Tsotsos [Tso84] discusses the decomposition of visual knowledge in such hierarchical way, referring to *orthogonal axes of representation*: the composition axis, the specialization axis, and the analogical axis. This layering is also supported elsewhere, e.g., Dana Ballard supports and defends the idea of hierarchical structuring in computational vision, based on new discoveries about the organization of the human brain[2] .

In what follows, we will explore the use of the composition and specialization axes for the representation of the visual knowledge, and will present a particular

[1] A scene descriptor is a prototypical scene which is made up of scene domain objects and certain relationships between them, for example, we may have a scene descriptor of an office.

[2] Dana Ballard presented his recent research in a seminar at the University of Waterloo in March 1987.

instance of Perceptual Reasoning that explicitly uses this layering to find interpretations.

4.2 A Theory of Schema Labelling

In "*A Theory of Schema Labelling*" [Hav85], Havens represents his visual knowledge as schemas organized in hierarchies. For his framework, Havens provides an algebraic account of how schemas get instantiated for a given image.

We present an instance of Perceptual Reasoning inspired by Havens' "Theory of Schema Labelling." This instance provides a declarative semantics for Havens' theory, and also indicates how Perceptual Reasoning can be used to characterize vision systems that exploit hierarchically organized visual knowledge. Furthermore, this instance of Perceptual Reasoning can be argued to satisfy the adequacy criteria recently listed by Mackworth [Mac87].

As an example of a domain, we consider the hypothetical recognition of drawings of trains as presented in [Hav85]. In this domain there is no occlusion possible and the objects are not defined as 3-D models. We present a restricted axiomatization of the visual knowledge explicitly organized into composition and specialization axioms, and show how we can exploit such a representation to obtain interpretations for an image. A more precise and realistic axiomatization of the domain is possible, but the one given here is adequate for our purpose.

4.3 Axiomatization of a Domain

For the representation of scene domain objects or scene descriptors, we suggest an approach similar to Minsky's *frames* [Min85] or Havens and Mackworth's *schemas* [HM83,Hav85], except that no explicit notion of prototypical objects is used[3] . Informally, if x is a scene domain object or scene descriptor, then a *prototype* for x is simply the collection of all axioms that contain $symbolize(x, y)$ for some y. During the recognition process, we informally refer to an individual y as something recognized as an x, or as an "*instance of prototype* x."

In order to adequately capture the desired visual knowledge in a hierarchy of abstraction of prototypes, we suggest that the visual knowledge be classified into "*composition*" and "*specialization*" axioms. Prototypes can be members of two hierarchies, a *part-of* hierarchy on the composition axis, and a *is-a* hierarchy on the specialization axis.

The composition axioms provide a set of rules which states acceptable decomposition of prototypical objects into their subparts. The axioms should express this decomposition down to the level where primitives and their relationships are described.

The specialization axioms assert relations that exists between pairs of objects, one of which is a specialization of the other, and states the particular properties that makes that object a specialization of the other.

In such cases the following holds about the set of facts available to Perceptual Reasoning:

[3]See Schubert [SGC78] and Feldman [Fel75] for some arguments against the explicit representation of frames.

Definition 12: *If C is the set of all composition axioms and S is the set of all specialization axioms, then for Γ, the* **set of facts** *available to Perceptual Reasoning, we have:*

$$C \cup S \subseteq \Gamma$$

An object can be defined at more than one resolution level by having many decomposition rules using the same object symbol in an axiom. In these cases, a disjunction of the decomposition rules defining the object is added as a composition axiom to the set of facts. For example, it might be desired to have a description of a tree at two different level of abstraction. One decomposition rule defining a tree as composed of two parts: the trunk and foliage, where the leaves are seen as a whole, another decomposition rule describing a tree as a trunk with branches and many leaves attached to the branches. The decomposition rules in such axioms use different sets of primitives.

4.3.1 Axiomatization Methodology

We now present an instance of Perceptual Reasoning that exploits composition and specialization axioms. For the purpose of the demonstration we do not present a general algorithm. The algorithm presented in the following sections depends on a particular axiomatization methodology. We define a very **restricted** form of composition and specialization axioms so that the recognition process defined will remain intuitive. A more general axiomatization of the visual knowledge is possible, but the restricted form used here is adequate for our purposes.

Instead of representing in one axiom the decomposition of an object into its subparts all the way down to the level of the primitives, we write composition axioms

at many layers of abstraction. The lowest layer uses primitives or general classes of primitives (for example see figure 4.2). Similarly, we write the specialization axioms at many layers of specialization. The finer grain of these specializations can be primitives of the image (for example see figure 4.3).

The composition axioms will form many hierarchies of symbols of scene domain objects and scene descriptors. The lowest levels of these hierarchies are primitives, and the top of these hierarchies are symbols of the most general objects and scene descriptors that we know about. Each symbol in each composition hierarchies can be the root of a specialization hierarchy. Of all the symbols in specialization hierarchies only symbols that are roots of these hierarchies can be used as symbols in the composition hierarchies. See figure 4.1.

In summary, the set of symbols used in composition hierarchies can be maximal elements in the set of symbols used in the specialization hierarchies. The symbols in the middle of specialization hierarchies cannot be in the set of symbols used in composition hierarchies.

4.3.2 Composition Axioms

We first present the syntax and semantics of *composition* axioms, which are used to describe how objects are composed (*part-of* hierarchy). Intuitively, we want to write axioms that express how a prototypical object is composed of a collection of other prototypical objects so that recognizing an object implies recognizing its subparts.

For example, recognizing a bicycle implies having recognized two wheels and appropriate relations amongst them, but recognizing an isolated wheel does not imply having recognized a bicycle. Consequently we represent a composition axiom for symbol S as disjunctions of all decomposition rules of the following form:

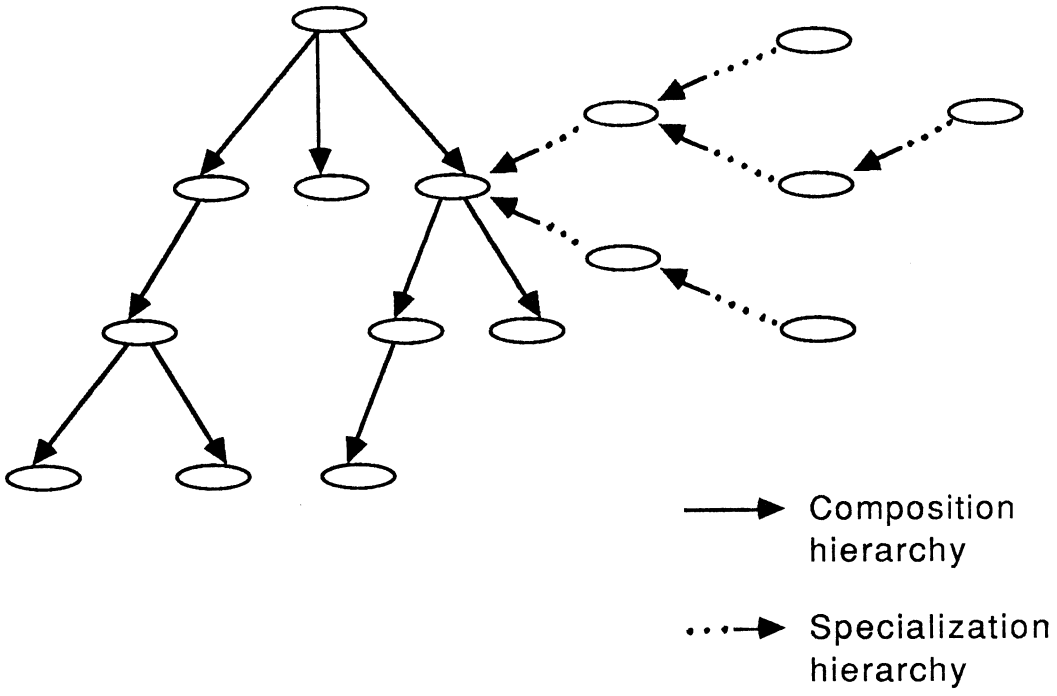


Figure 4.1: Composition and Specialization Axes

$$
\begin{aligned}
symbolize(S,T) \supset\ & aggregation[X_1, X_2, \ldots, X_n] \ \wedge \\
& symbolize(Part_1, X_1) \ \wedge \\
& symbolize(Part_2, X_2) \ \wedge \\
& \vdots \\
& symbolize(Part_n, X_n) \ \wedge \\
& relations[X_1, X_2, \ldots, X_n];
\end{aligned}
$$

Where *relations* is a finite conjunction of atomic assertions expressing the relations between the parts, and *aggregation* is a finite conjunction of atomic assertions expressing the aggregation of the cues from the parts into one cue for the object. In this example, since Prolog was used for the implementation, we use lists as second arguments of the predicate *symbolize* to denote the aggregation of the cues from the image.

The intended interpretation of the above decomposition rule schema is that any object which can be labelled with $Symbol$ implies the existence of all its appropriately labelled parts, $Part_1 \ldots Part_n$, and the truth of the relations *relations*. Note that the above is a decomposition rule schema and that $Symbol$ and $Part_1 \ldots Part_n$ can be the name of any scene domain objects (e.g., tree, bicycle). The variables in composition axioms are universally quantified.

In our example of the train domain, a train consists of an engine, followed by a non-empty set of wagons (carSet), optionally ending with a caboose. See figure 4.2. The composition axiom for trains is then defined by the disjunction of the following two decomposition rules:

$$
\begin{aligned}
symbolize(train,T) \supset\ & T = [X,Y] \ \wedge \\
& symbolize(engine, X) \ \wedge \\
& symbolize(carSet, Y);
\end{aligned}
$$

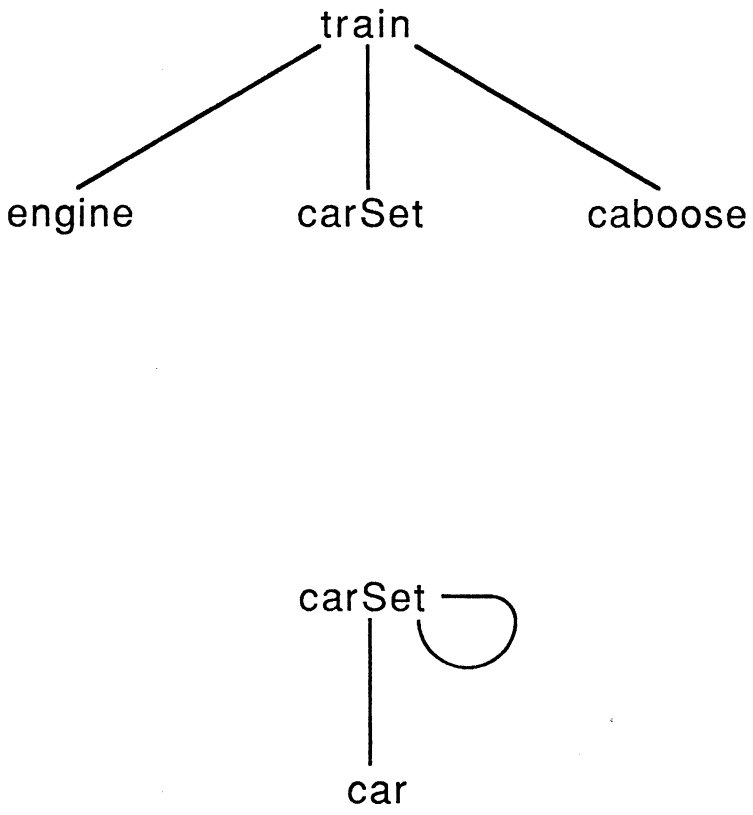


Figure 4.2: Composition Hierarchies of the Train Domain

$$\begin{aligned} symbolize(train, T) \supset\ & T = [X, Y, Z] & \wedge \\ & symbolize(engine, X) & \wedge \\ & symbolize(carSet, Y) & \wedge \\ & symbolize(caboose, Z); \end{aligned}$$

We give a recursive definition of a carSet through a composition axiom formed of the disjunction of the following decomposition rules:

$$\begin{aligned} symbolize(carSet, C) \supset\ & C = [X] & \wedge \\ & symbolize(car, X); \end{aligned}$$

$$\begin{aligned} symbolize(carSet, C) \supset\ & C = [X|Y] & \wedge \\ & symbolize(car, X) & \wedge \\ & symbolize(carSet, Y); \end{aligned}$$

$$\begin{aligned} symbolize(carSet, C) \supset\ & append(X, Y, C) & \wedge \\ & ne(X, []) \wedge ne(Y, []) & \wedge \\ & symbolize(carSet, X) & \wedge \\ & symbolize(carSet, Y); \end{aligned}$$

The intended interpretation of the last decomposition rule is that recognizing a carSet could imply recognizing two non-empty carSets together. This decomposition rule is added because we didn't specified that cars in a carset are connected together, therefore two carsets are considered as forming only one carset. The *append* predicate defines the aggregation of the cues and the *ne* predicate insure that the two carSets are not empty by verifying that the carSet symbols are not attached to an empty set of cues from the image.

Note that further relations (constraints) could have been added to those axioms, e.g., that the parts of the train are connected together. Note also that we could

have given a more complete axiomatization of the drawings of trains by giving composition axioms for the parts of the parts of a train, and taking it down to the edges and surfaces found in the image, or any level appropriate to the low level vision system.

4.3.3 Specialization Axioms

Each object symbol of a composition hierarchy may be the root a hierarchy of other object symbols, where all object symbols part of such a hierarchy are specializations of that root.

Specialization axioms are of two forms. The first is

$$\forall X \quad (\, conds[X] \wedge symbolize(Class, X) \supset symbolize(Special, X)\,) \wedge$$
$$(\, symbolize(Special, X) \supset symbolize(Class, X)\,)$$

where $conds[X]$ is a conjunction of constraints.

We use the following notation as a shorthand for the above axiom as it seems to follow a more natural way in which the knowledge come up:

$$symbolize(Special, X) : symbolize(Class, X) \leftarrow Conds;$$

The intended interpretation of the axiom is that the symbol *Special* is a specialization of the symbol *Class* if the conditions *Conds* hold. The colon represents the specialization/generalization relation between the symbol *Special* and the symbol *Class*[4] ; the connective $\leftarrow$ indicates the conditions.

The second form of specialization axioms relate specialized objects that are primitives to their general class in the composition hierarchies.

[4]The colon can also be thought of as denoting the *ISA* relation between the *Special* symbol and the *Class* symbol.

$$\forall X \; (\; symbolize(Special, X) \supset symbolize(Class, X) \;)$$

and where *Special* is a primitive.

Which in the shorthand notation is written:

$$symbolize(Special, X) : symbolize(Class, X);$$

In summary, specialization axioms form a hierarchy of scene objects which are refinements of their composition root. See figure 4.3.

As defined above, specialization axioms can be pictured as hierarchies of specialization for the different composition symbols. For the domain of trains we have the following specialization axioms. All specialization axioms are universally quantified. Trains are subdivided into two specialization classes, longHauls and shortHauls. The criteria of classification is the specialization of the engine symbol. The engine symbol has two specialized symbols: a switcher (a small yard engine) or a loco (a larger long-haul locomotive).

Trains:

$$symbolize(longHaul, [X|Y]) \; : \; symbolize(train, [X|Y]) \leftarrow symbolize(loco, X);$$

$$symbolize(shortHaul, [X|Y]) \; : \; symbolize(train, [X|Y]) \leftarrow symbolize(switcher, X);$$

Engines:

$$symbolize(loco, X) \; : \; symbolize(engine, X);$$

$$symbolize(switcher, X) \; : \; symbolize(engine, X);$$

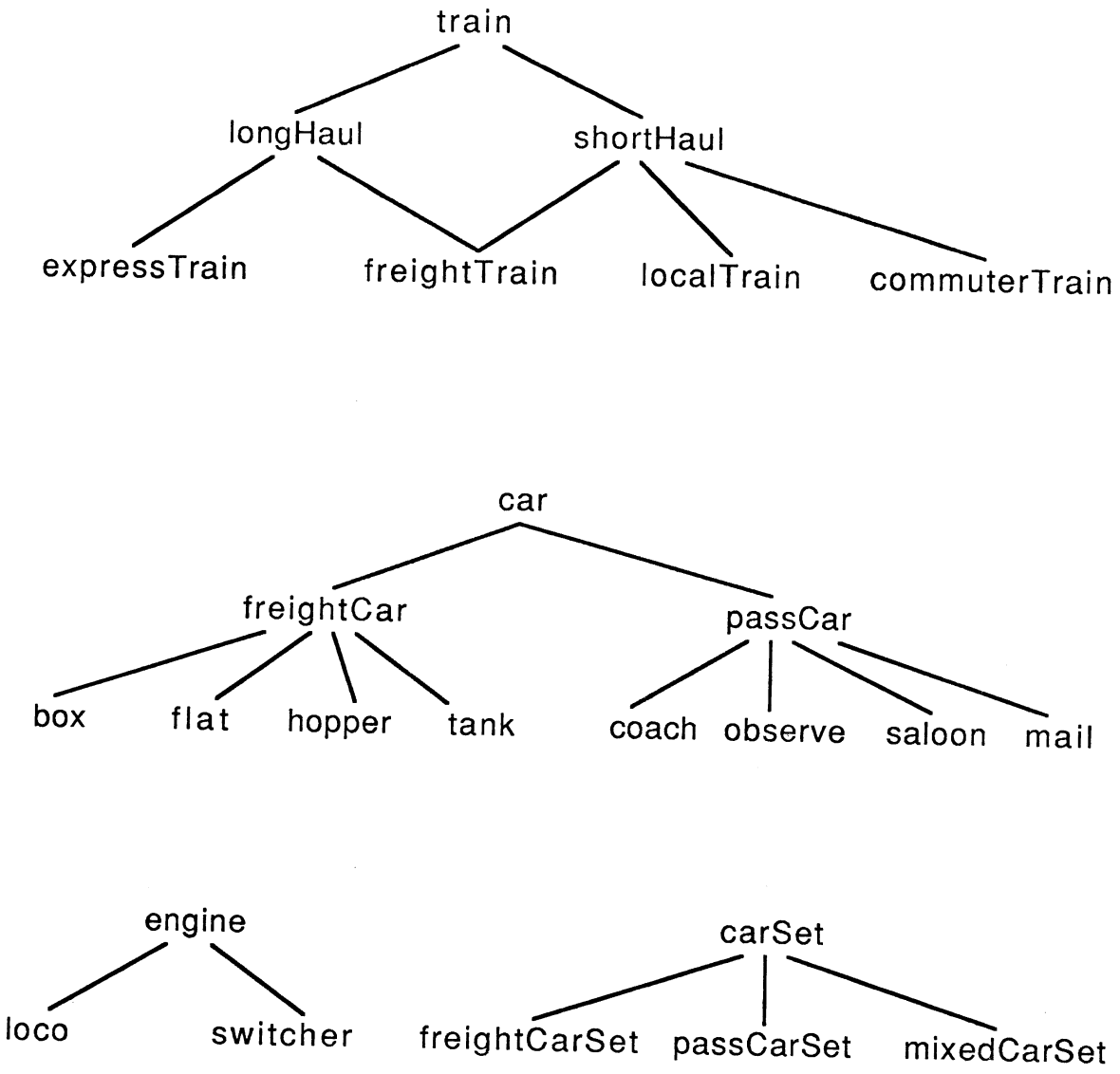


Figure 4.3: Specialization Hierarchies of the Train Domain

The same comment about details, as for the composition axioms, apply for the specialization axioms; more precision could be specified in the axioms, which could force a deeper focusing.

LongHauls and shortHauls are further specialized. LongHauls can be express trains or freight trains, and shortHauls can be local trains, commuter trains, or freight trains. The further specialization of longHauls and shortHauls depends on the specialization of the carSet symbol.

LongHauls:

$$symbolize(expressTrain, [X, Y]) \quad : symbolize(longHaul, [X, Y]) \leftarrow symbolize(passCarSet, Y);$$

$$symbolize(freightTrain, [X, Y, Z]) \quad : symbolize(longHaul, [X, Y, Z]) \leftarrow symbolize(freightCarSet, Y);$$

ShortHauls:

$$symbolize(localTrain, [X, Y|Z]) \quad : symbolize(shortHaul, [X, Y|Z]) \leftarrow symbolize(mixedCarSet, Y);$$

$$symbolize(commuterTrain, [X, Y]) \quad : symbolize(shortHaul, [X, Y]) \leftarrow symbolize(passCarSet, Y);$$

$$symbolize(freightTrain, [X, Y, Z]) \quad : symbolize(shortHaul, [X, Y, Z]) \leftarrow symbolize(freightCarSet, Y);$$

Cars (wagons) can be subdivided into two classes: freight cars which includes boxes, flats, tanks and hoppers, and passenger cars which includes coaches, mails, observes and saloon cars. Consequently carSets can be specialized to freight carSets, passenger carSets or mixed carSets.

CarSets:

$$
\begin{aligned}
&symbolize(freightCarSet, [X]) && : symbolize(carSet, [X]) \\
&&& \leftarrow symbolize(freightCar, X);
\end{aligned}
$$

$$
\begin{aligned}
&symbolize(freightCarSet, [X|Y]) && : symbolize(carSet, [X|Y]) \\
&&& \leftarrow symbolize(freightCar, X) \wedge \\
&&& symbolize(freightCarSet, Y);
\end{aligned}
$$

$$
\begin{aligned}
&symbolize(passCarSet, [X]) && : symbolize(carSet, [X]) \\
&&& \leftarrow symbolize(passCar, X);
\end{aligned}
$$

$$
\begin{aligned}
&symbolize(passCarSet, [X|Y]) && : symbolize(carSet, [X|Y]) \\
&&& \leftarrow symbolize(passCar, X) \\
&&& \wedge symbolize(passCarSet, Y);
\end{aligned}
$$

$$
\begin{aligned}
&symbolize(mixedCarSet, X) && : symbolize(carSet, X) \\
&&& \leftarrow member(C_1, X) \wedge \\
&&& symbolize(passCar, C_1) \wedge \\
&&& member(C_2, X) \wedge ne(C_1, C_2) \wedge \\
&&& symbolize(freightCar, C_2);
\end{aligned}
$$

The predicates *member* and *ne* used above have the following meaning:

$member(X,Y)$ is true *if* X is an element of the list Y, $ne(X,Y)$ is true $if\ X \neq Y$. The intended interpretation for these predicate is that *member* refers to a subset relationship between the cues of the image and *ne* insure that we are not dealing with the same subset of cues.

Freight cars:

$$
\begin{aligned}
&symbolize(freightCar, X) &&: symbolize(car, X) \leftarrow symbolize(box, X);\\
&symbolize(freightCar, X) &&: symbolize(car, X) \leftarrow symbolize(flat, X);\\
&symbolize(freightCar, X) &&: symbolize(car, X) \leftarrow symbolize(tank, X);\\
&symbolize(freightCar, X) &&: symbolize(car, X) \leftarrow symbolize(hopper, X);\\
&symbolize(box, X) &&: symbolize(freightCar, X);\\
&symbolize(flat, X) &&: symbolize(freightCar, X);\\
&symbolize(tank, X) &&: symbolize(freightCar, X);\\
&symbolize(hopper, X) &&: symbolize(freightCar, X);
\end{aligned}
$$

Passenger cars:

$$
\begin{aligned}
&symbolize(passCar, X) &&: symbolize(car, X) \leftarrow symbolize(mail, X);\\
&symbolize(passCar, X) &&: symbolize(car, X) \leftarrow symbolize(observe, X);\\
&symbolize(passCar, X) &&: symbolize(car, X) \leftarrow symbolize(coach, X);\\
&symbolize(passCar, X) &&: symbolize(car, X) \leftarrow symbolize(saloon, X);\\
&symbolize(mail, X) &&: symbolize(passCar, X);\\
&symbolize(observe, X) &&: symbolize(passCar, X);\\
&symbolize(coach, X) &&: symbolize(passCar, X);\\
&symbolize(saloon, X) &&: symbolize(passCar, X);
\end{aligned}
$$

The axiomatization given above for the domain of trains makes some assumptions. One is that we are able to recognize drawings of the prototypes: coach, box,

mail, etc, and therefore the observations, presented in the intermediate image, are instantiations of these symbols. In other word, these symbols are the primitives for this example. Using this assumption we have all the visual knowledge we need to complete our example.

For the purpose of this example we define $\rhd_{train}$ such that the "deepest" symbols in the specialization hierarchies of the highest symbols of the composition hierarchies will correspond to the maximal elements of the $\rhd_{train}$ relation.

Definition 13: *The relation* $x \rhd_{train} y$ *holds if from* Γ *there exists a derivation (inference chain) such that* $symbolize(x, u)$ *is used to derive (infer)* $symbolize(y, v)$ *in that derivation.*

Figure 4.4 exhibits a subset of the $\rhd_{train}$ relation over the symbols of the train domain. We intentionally left out the transitive and reflexive tuples in figure 4.4 for clarity of the presentation, but these links should also be present for the figure to adequately display the complete $\rhd_{train}$ relation. As examples of instances this relation we have $localTrain \rhd_{train} car$ and $train \rhd_{train} carSet$.

We will now describe a recognition process that is an instance of Perceptual Reasoning, and that exploits the hierarchies represented by the restricted forms of composition and specialization axioms presented.

4.4 A Recognition Process

Intuitively, for this example we want to specify an incremental process of the image such that the interpretation found is a "most specific" interpretation of the given image. To do so, we exploit the organization of the visual knowledge. As the

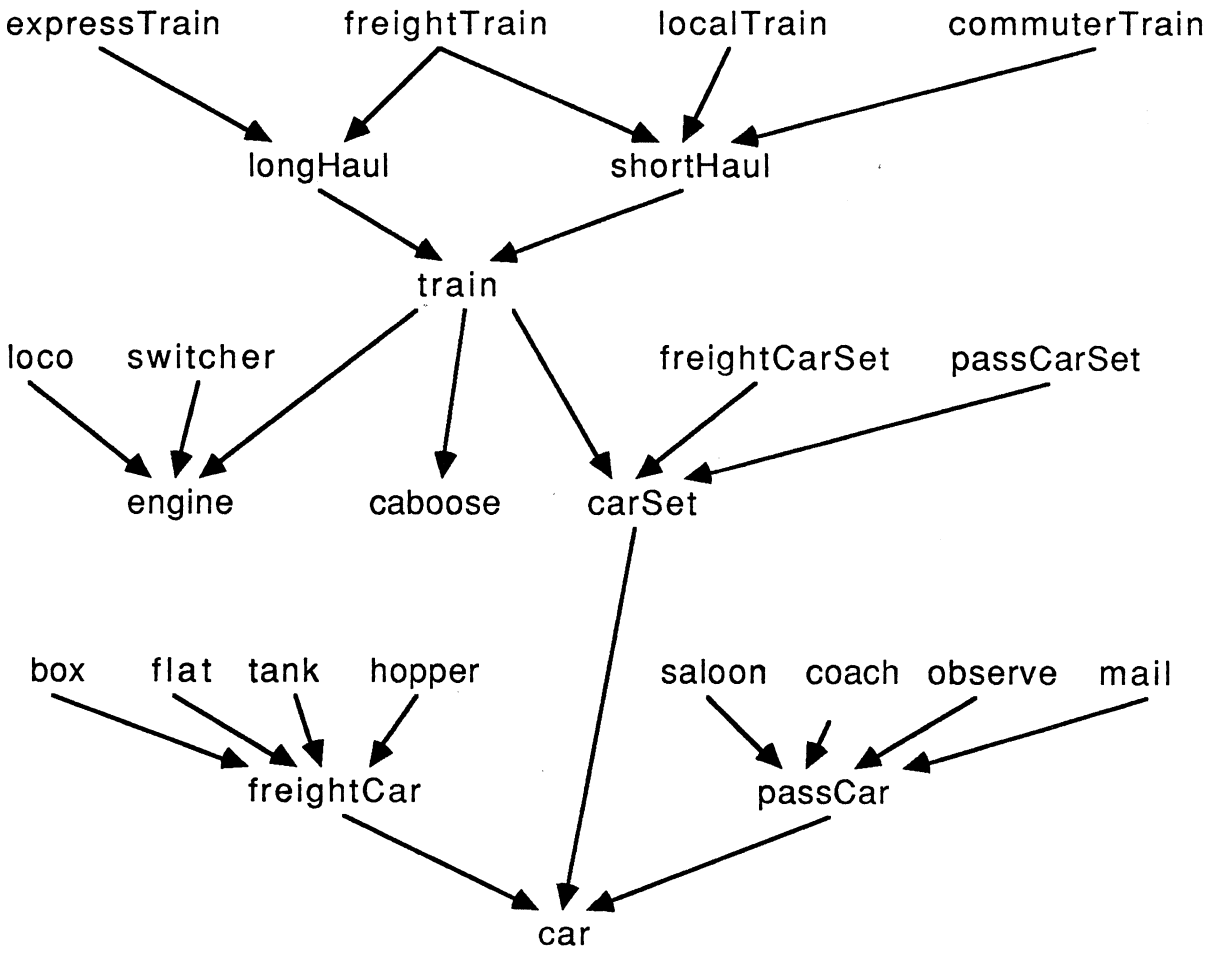


Figure 4.4: The $\triangleright_{train}$ Relation for the Train Domain

visual knowledge is organized into a hierarchy of abstraction, incrementally providing analysis of the image will correspond to moving up the structure of the visual knowledge.

In chapter 3, we defined the logical criteria for visual recognition by characterizing the process of finding an interpretation as finding theories that explain the observations, ranking the possible theories using a preference criteria, and verifying the preferred theories until one is found to be coherent with the image. We now describe how we can exploit the structure of the visual knowledge presented in the previous section to analyze images by dynamically climbing the hierarchies of the knowledge.

4.4.1 Exploiting the Visual Knowledge

We want to reduce the semantic problem of finding an interpretation, defined in chapter 3, to a syntactic one that a machine can solve. Having divided our visual knowledge into composition and specialization axioms, we define a separate phase of the recognition process for each of the two categories of the visual knowledge, namely the composition phase and the specialization phase. We use an iterative theory formation process which produces intermediate analyses of the scene during the composition phase, and then applies the specialization phase. The goal of the composition phase is to incrementally analyze the image so that we use objects and scene descriptors that are more abstract then the current analysis (i.e., progress up the composition hierarchies). The goal of the specialization phase is to specialize the objects and scene descriptors, provided by the composition phase, as far down their specialization hierarchies as possible[5] so that they are as specific as possible.

[5]Recall that each symbol of the composition hierarchies may be the root of a specialization hierarchy.

During the recognition process we maintain two environments: the general knowledge of the image and the composition environment. The *general knowledge of the image* environment corresponds to all the initial observations, together with every inference that was made to the present point of the process; we refer to it as **GKI**. The *composition environment*, **CE**, is an environment where the current analysis of the image is kept, only instances of the *symbolize* predicate with symbols from composition hierarchies are found in **CE**. For example, if we had an observation in the intermediate image of an instance of the prototype "box", the initial composition environment would contain the fact that it is a "car".

Since some observations can be primitives that are down in the specialization hierarchies, a generalization phase is first used in order to initialize adequately the two environments. The **GKI** is adequately initialized if it initially contains all the initial observations and all the valid generalization inferences for them. The **CE** is adequately initialized if it initially contains the most general forms for the initial observations. These definitions will become clearer in the following section.

The algorithm for the iterative recognition process is simply the generalization phase followed by some iterations of the composition phase, and finally the specialization phase. The details of each phase of the process are found in the following sections.

The Iterative Recognition Process

1. **do** generalization phase
2. **repeat** composition phase **until** no more composition theories can be found
3. **do** specialization phase

Composition theories are defined in the composition phase section (Section 4.4.3).

4.4.2 The Generalization Phase

The recognition process starts with the content of the explicit part of the intermediate image as its initial set of observations. The explicit part of the intermediate image is provided by the low level process. As some of the primitives are leafs of specialization hierarchies, we first need to generalize the observations from the intermediate image in order to initialize adequately the two environments.

A valid generalization is one derived from the specialization/generalization relations described in Section 4.3.2. For example, given the following specialization axiom:

$$symbolize(loco, X) : symbolize(engine, X);$$

a valid generalization of an observation $symbolize(loco, a)$, where a is an instance observed in the image of the prototype "loco", would be to infer $symbolize(engine, a)$, This generalization is valid since:

$$symbolize(loco, X) \supset symbolize(engine, X).$$

Here is an overview of the algorithm for the generalization phase:

The Generalization Phase

1. **set GKI** to be the observations from the intermediate image
2. **for each** observation in the intermediate image

(a) **repeat until** no more valid generalization possible

i. **apply** a valid generalization

ii. **add** the generalization to **GKI**

(b) **make CE** the set of most general forms found.

Because of the restricted way our knowledge is structured, only instances of the *symbolize* predicate with symbols that are in composition hierarchies will be found in the **CE** at the end of the generalization phase. This comes from the fact that the hierarchies are finite and explicitly known.

4.4.3 The Composition Phase

The role of the composition phase is to progressively update the composition environment. To do so, we find *composition theories* to explain the present elements of the composition environment (**CE**).

Definition 14: *Given the set of all composition axioms C, we say that T is a* **composition step** *that explains a non empty set of elements, Φ, from the composition environment* **CE** *if T is a ground instance of the hypothesis symbolize(X,Y) such that*

$$C \cup T \supset \Phi \text{ and}$$
$$C \cup T \text{ is consistent and}$$
$$T \notin \mathbf{CE} \text{ and}$$
$$\neg\exists T' \text{ s.t. } T' \neq T \text{ and } C \cup T \supset T'$$

Where T' is a ground instance of symbolize(X,Y) and $\Phi \subseteq$ CE.

Consistency of $C \cup T$ can be verified by failing to prove an inconsistency. See [PGA86] for more on this. The above definition for composition theories ensure that we only consider theories that represent one inference step.

When looking for a composition step that explains a subset of the composition environment, generally more than one theory is going to be candidate. We need a composition step selection scheme as a way to insure an adequate aggregation of the elements of **CE**, so that the selected composition theory represents a step toward a most specific interpretation. Preference, at each step of the iterative process (going up the composition hierarchy), is always given to the composition theories that uses symbols that are at a lower level in the composition hierarchies. Such cautious progress through the composition hierarchies, ensures that no observations are left behind in our search for a most specific interpretation. This is particularly important for objects defined recursively. For example, imagine a situation where the present state of the composition environment is that the image depicts an engine, a set of two cars (wagons), and an individual car (wagon). In this situation, the composition step with a set of three cars should be selected over the composition step that it is a train (composed of an engine and two cars), as the latter will leave a car behind and not incrementally lead us to a most specific interpretation.

This selection of composition theories is based on, but should not be confused with, the definition of "a preferred interpretation" of Chapter 3. Our criteria of preference here is based on our interest in taking the best possible step up a composition hierarchy.

We define the selection of a composition step over others in the following terms:

Definition 15: *We* **select** *composition step T_1 over composition step T_2 if either*

a) $T_1 = symbolize(p_1, x)$ *and* $T_2 = symbolize(p_2, y)$, *and*

$$p_2 \rhd p_1$$

b) $T_1 = symbolize(p, x)$ *and* $T_2 = symbolize(p, y)$ *and*

$$x \supseteq y$$

Otherwise the two theories are incomparable.

The candidate composition theories are ordered according to the above definition, and we commit to the preferred one. When theories are not comparable any one can be committed to. This arbitrary choice will be corrected later if wrong, when more context will provide grounds for preference. The context we refer to here will come from verifying coherence of theories that uses symbols higher up in the $\rhd$ relation of symbols ($\rhd_{train}$ for this example).

We now present an outline of the algorithm for the composition phase.

The Composition Phase

1. **find all** valid composition steps T for the **CE**
2. **order** T's according to selection criteria
3. **pick** first T that is coherent
   (a) **add** T to **GKI**
   (b) **replace** Φ by T in **CE**, where Φ is the set of elements from **CE** explained by T.

Note that verifying coherence in this example only means that either the primitives have been observed or they can be verified in the image, as we are not dealing with occlusion or 3-D models in this example. Also note that, when verifying coherence of the composition step, new primitives verified may suggest different symbols for cues or features, in these cases the new symbols replaces the previous ones, as these new symbols where suggested from higher levels of the composition hierarchy, and therefore arises from a more general consideration of the surrounding context. Any newly verified primitives during the verification of coherence are added to the **GKI** (i.e. new primitives for which the *symbolize* relation have been establish).

4.4.4 The Specialization Phase

The specialization phase is applied when no more composition theories can be found to explain a subset of **CE**. It outputs a set **I** consisting of the most specific specializations for the content of the **CE**. This is achieved by specializing as far down as possible the specialization hierarchies of the final composition theories. To take a step down the specialization hierarchy of the composition step T has the following meaning:

Definition 16: *Given the set of all specialization axioms S and the general knowledge of the image* **GKI**, *we say that ST is a* **specialization theory** *of T, if*

$$\exists \text{ a ground instance of } \{ST : T \leftarrow Conds\} \in S$$
$$\text{s.t. for each relation } \sigma \text{ of } Conds \text{ either}$$
$$\mathbf{GKI} \models \sigma$$
$$\text{or}$$
$$\sigma \text{ can be verified in the image.}$$

According to this definition, we have a specialization theory for an element of **CE**, if there exists a ground instance of a specialization axiom such that for every condition (constraint) of that axiom, either the condition is already recorded in the general knowledge of the image, or we can verify the condition in the image. Some of these conditions may be primitives to be detected, in which case they are added to **GKI** if detected. We stop going down a branch of the specialization hierarchy when one of the required conditions is false or unknown.

We now present an overview of the algorithm for the specialization phase.

The Specialization Phase

1. **for each** element T of **CE**
   (a) **repeat until** no more specialization possible
      i. **apply** valid specialization ST
      ii. **add** ST to **GKI**
   (b) **add** the last specialization to **I**

4.5 Summary and Example

Abstractly, we can define the algorithm in the following brief terms: Generalize all observations from the intermediate image, so that we are in the different composition hierarchies. For as long as it is possible, take a step up to a neighbour node in a composition hierarchy to explain a subset of **CE**. When you can't take any more steps up in any composition hierarchy then end the composition phase, and then go as far as possible in the specialization hierarchies of the final nodes of the composition phase. Figure 4.5 tries to express this process.

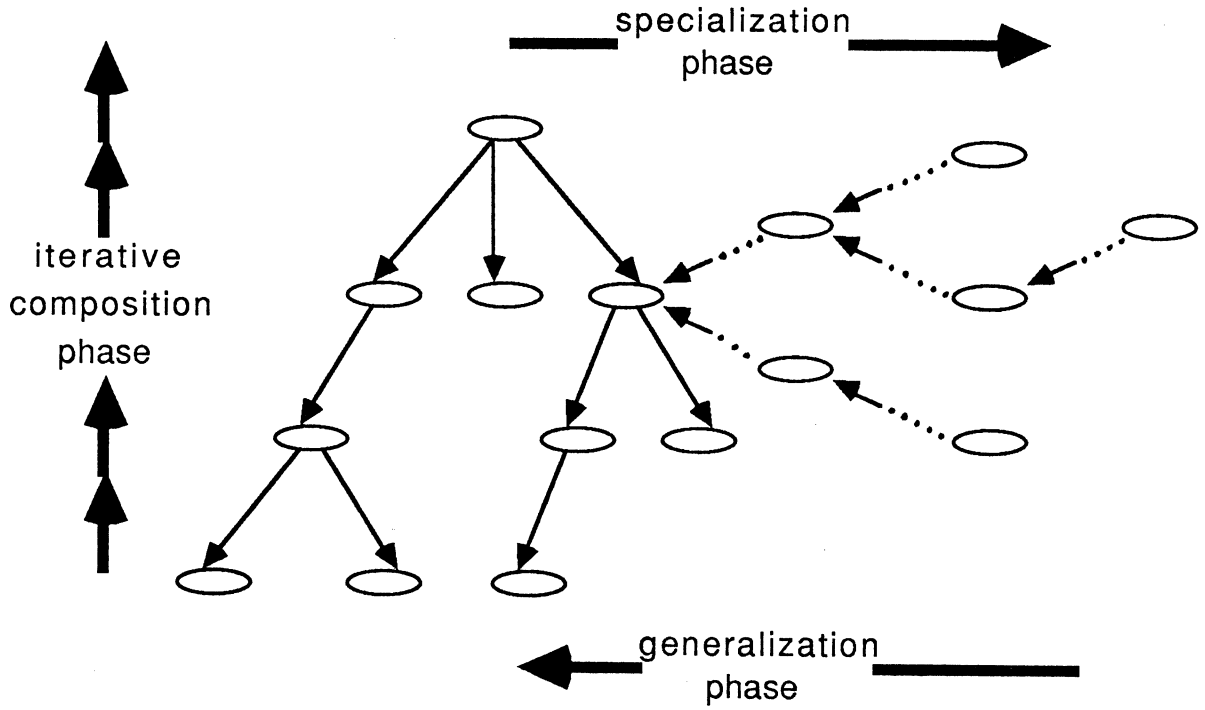


Figure 4.5: Abstract View of a Recognition Process

Note that this algorithm is not a general way or necessarily the most efficient way to compute interpretations, it is only meant as a demonstration of how we can use the semantics presented in Chapter 3 as a guide to define a recognition process and to characterize different model-based approaches that exploit hierarchies of abstractions in there representation of the visual knowledge.

At the end of this iterative recognition process, the composition environment **CE** will contain instances of the *symbolize* predicate that use the highest possible symbols in the different composition hierarchies for the initial observations, and the set **I** will contain the symbols that are as deep as possible in the specialization hierarchies of the symbols from **CE**. The set **I** obtained at the end of the iterative recognition process is referred to as the *answer* obtained by the process and and should correspond to a most specific interpretation of the image.

Arguments of the correctness of the iterative recognition process with respect to the semantics presented in Chapter 3 is presented in the next section. For now, let's take up an example used by Havens in [Hav85], and see how the iterative recognition process obtain a most specific interpretation for this example.

We will go through this example fairly quickly, because the details are not necessary, to give an idea of what is happening during the recognition process.

Imagine that the intermediate image contains the following observations:

$$\{\ symbolize(box, a), symbolize(switcher, b), symbolize(coach, c),$$
$$symbolize(flat, d), symbolize(caboose, e)\ \}$$

Where a,b,c,d,e are cues from the image which have been recognized to be instances of the corresponding symbols. Applying the generalization phase, the following valid generalizations can then be derived from the given set of observations:

$$
\begin{aligned}
&symbolize(box, a) && \supset symbolize(freightCar, a);\\
&symbolize(freightCar, a) && \supset symbolize(car, a);\\
&symbolize(switcher, b) && \supset symbolize(engine, b);\\
&symbolize(coach, c) && \supset symbolize(passCar, c);\\
&symbolize(passCar, c) && \supset symbolize(car, c);\\
&symbolize(flat, d) && \supset symbolize(freightCar, d);\\
&symbolize(freightCar, d) && \supset symbolize(car, d);
\end{aligned}
$$

After the generalization phase the **GKI** and **CE** contain the following:

GKI = all the observations from the intermediate image plus all the inferences from the generalization phase, i.e.

$$
\begin{aligned}
\{\ & symbolize(box, a), symbolize(freightCar, a), symbolize(car, a),\\
& symbolize(switcher, b), symbolize(engine, b), symbolize(coach, c),\\
& symbolize(passCar, c), symbolize(car, c), symbolize(flat, d),\\
& symbolize(freightCar, d), symbolize(car, d), symbolize(caboose, e)\ \}
\end{aligned}
$$

$$
\begin{aligned}
\textbf{CE} = \{\ & symbolize(car, a), symbolize(engine, b), symbolize(car, c),\\
& symbolize(car, d), symbolize(caboose, e)\ \}
\end{aligned}
$$

After the generalization phase, we apply the composition phase until no further composition theories can be found to explain subsets of the composition environment (**CE**). For our present example, the candidate composition theories according to the definition are:

$$
\begin{aligned}
T_1 &= symbolize(carSet, [a])\\
T_2 &= symbolize(carSet, [b])\\
T_3 &= symbolize(carSet, [c])
\end{aligned}
$$

According to our selection scheme, these theories are not comparable and any can be used. We arbitrarily pick T_1, so the composition environment becomes:

$$\begin{aligned}\mathbf{CE} = \{\ & symbolize(carSet, [a]), symbolize(engine, b), symbolize(car, c), \\ & symbolize(car, d), symbolize(caboose, e)\ \}\end{aligned}$$

Going on with the composition phase, we have, once more, to find the composition theories that explains a subset of the **CE**. The available theories are:

$$\begin{aligned} T_1 &= symbolize(carSet, [c]) \\ T_2 &= symbolize(carSet, [d]) \\ T_3 &= symbolize(train, [b, [a], e]) \\ T_4 &= symbolize(train, [b, [a]]) \end{aligned}$$

Theories T_1 and T_2 are not comparable, but are both preferred over theories T_3 and T_4 as $train \rhd_{train} carSet$. T_3 is preferred to T_4. We then pick T_1 and commit to it. The composition environment becomes then:

$$\begin{aligned}\mathbf{CE} = \{\ & symbolize(carSet, [a]), symbolize(engine, b), symbolize(carSet, [c]) \\ & symbolize(car, d), symbolize(caboose, e)\ \}\end{aligned}$$

We go on with the composition phase, picking the first theory in the partially ordered list of possible composition theories given, and commit to it.

Composition phase:

$$\begin{aligned} T_1 &= symbolize(carSet, [a, c]) \\ T_2 &= symbolize(carSet, [c, a]) \end{aligned}$$

$$T_3 = symbolize(carSet, [d])$$
$$T_4 = symbolize(train, [b, [a], e])$$
$$T_5 = symbolize(train, [b, [a]])$$
$$T_6 = symbolize(train, [b, [c], e])$$
$$T_7 = symbolize(train, [b, [c]])$$

Theories T_1 and T_2 are both preferred to one another since $carSet \rhd_{train} carSet$, but are incomparable to theory T_3. All three theories are preferred to theories T_4, T_5, T_6 and T_7 where T_4 is preferred to T_5 and T_6 is preferred to T_7. The composition environment is now:

$$\{ symbolize(carSet, [a, c]), symbolize(engine, b), symbolize(car, d), symbolize(caboose, e) \}$$

Again, we have

Composition phase:

$$T_1 = symbolize(carSet, [d])$$
$$T_2 = symbolize(train, [b, [a, c], e])$$
$$T_3 = symbolize(train, [b, [a, c]])$$

where T_1 is preferred to T_2 and T_3 since $train \rhd_{train} carSet$. After this invocation of the composition phase, we have:

$$\mathbf{CE} = \{symbolize(carSet, [a, c]), symbolize(engine, b), symbolize(carSet, [d]), symbolize(caboose, e) \}$$

Composition phase:

$$T_1 = symbolize(carSet, [a, c, d])$$
$$T_2 = symbolize(carSet, [d, a, c])$$
$$T_3 = symbolize(train, [b, [a, c], e])$$
$$T_4 = symbolize(train, [b, [a, c]])$$

where T_1 and T_2 are both preferred to one another, and are both preferred to T_3 and T_4. The composition environment is now:

$$\{ symbolize(carSet, [a, c, d]), symbolize(engine, b), symbolize(caboose, d) \}$$

Composition phase:

$$T_1 = symbolize(train, [b, [a, c, d], e])$$
$$T_2 = symbolize(train, [b, [a, c, d]])$$

where T_1 is preferred to T_2 as the cues of T_1 are a superset of the cues of T_2.

The composition phase is then done as there is no other composition theory that can be found. **CE** is now made up of only $symbolize(train, [b, [a, c, d], e])$. The specialization phase is then invoked to specialize the elements of the composition environment.

Specialization Phase:

Specialization $symbolize(shortHaul, [b, [a, c, d], e])$ is then made as there is a ground instance of a specialization axiom such that:

$$symbolize(shortHaul, [b, [a, c, d], e]) : symbolize(train, [b, [a, c, d], e]) \leftarrow symbolize(switcher, b);$$

and condition $symbolize(switcher, b)$ is a logical consequence of **GKI** since it was one of the initial observations. We can further specialize:

$$symbolize(shortHaul, [b, [a, c, d]e])$$
$$\text{to } symbolize(localTrain, [b, [a, c, d], e])$$
$$\text{since:}$$
$$symbolize(localTrain, [b, [a, c, d], e]) : symbolize(shortHaul, [b, [a, c, d], e]) \leftarrow symbolize(mixedCarSet, [a, c, d]);$$

and $symbolize(mixedCarSet, [a, c, d])$ can be verified with the axiom:

$$symbolize(mixtedCarSet, [a, c, d]) : symbolize(carSet, [a, c, d]) \leftarrow member(a, [a, c, d]) \wedge symbolize(passCar, c) \wedge member(c, [a, c, d]) \wedge ne(c, a) \wedge symbolize(freightCar, a)$$

where

$$symbolize(carSet, [a, c, d]) \wedge symbolize(passCar, c) \wedge symbolize(freightCar, a)$$

are logical consequences of **GKI**.

The process then terminates because no more specialization can be found for the elements of **CE**. At the end of the recognition process, the composition environment contains the prototype $symbolize(train, [b, [a, c, d], e])$ which is then the most general prototype, from the composition hierarchies, explaining the observations. The answer of the iterative recognition process, which is the content of **I** at the

end of the process, is $symbolize(localTrain, [b, [a, c, d], e])$. This answer is a most specific interpretation for the initial set of observations (see figure 4.4).

The description of the scene depicted by the image that produced the original set of observations used in this example:

$$\{\ symbolize(box, a), symbolize(switcher, b), symbolize(coach, c),$$
$$symbolize(car, d), symbolize(caboose, e)\ \}$$

is that it is a scene of a local train.

Note that many details were left out during the recognition process for this example. Our intention was to show the incremental analysis of the image that takes place during the iterative recognition process.

4.6 Correctness of the Recognition Process

We now discuss the correctness of the iterative recognition process presented in this chapter, with respect to the semantics presented in Chapter 3. Formal proofs of the correctness of the recognition process presented here would only result in voluminously unrequired formal discussion, since our purpose was to only indicate how Perceptual Reasoning can be use as a guide to define a recognition process, and to show how we can characterize, in more detail, model-based approaches that exploit hierarchical organization of the visual knowledge. We instead informally argue that the iterative recognition process presented is correct with respect to the semantics defined in Chapter 3.

It is very important to notice here that this argument is valid only because of the axiomatization methodology we used (i.e. the restrictions on the axioms), and that if this methodology is not followed the argument doesn't hold any more.

We first can informally defend the recognition process presented as sound, meaning that all answers found by the iterative recognition process are valid interpretations, by the following arguments:

Suppose the recognition process found an answer **I** that is not an interpretation according to our semantics of Chapter 3. This would then either mean that:

(1) There exists an observation from the intermediate image that is not a logical consequence of **I**.

(2) $\Gamma \cup$ **I** is not coherent.

But (1) cannot be the case since the recognition process presented, starts with all the observations from the intermediate image, and each step is like a rewriting system where observations only get rewritten by objects that imply them. Therefore there can't be an observation from the intermediate image that is not a logical consequence of **I**. It also impossible for (2) to be the case since for **I** to be an answer of the recognition process presented, it must be a set of specializations of the composition theories in **CE**, and as a criterion of the composition phase, each composition step use to modify the **CE** must be coherent, and the specialization phase doesn't affect the coherence.

There is then no possible way to obtain an answer that is not an interpretation, and we therefore say that recognition process is sound. □

The iterative recognition process presented is also complete, meaning that the recognition process will find a most specific interpretation for the image if one exists.

Completeness of the recognition process can be argued in the following manner: Suppose that wasn't the case, and that there exists an interpretation **A** that is more specific then the interpretation **I** found by the recognition process. This then means

that $\Gamma \cup \mathbf{A} \models \mathbf{I}$. Because of the restricted way the visual knowledge is structured in the presented process, there are only two possible way for this to be the case, either

(1) **A** must include at least one element that is a specialization of an element of **I** or

(2) **A** contains the specialization of one object that implies some objects for which the specializations are in **I**.

But if (1) was the case, the specialization phase would have done one more specialization, since the specialization phase only stops when no more specialization can be done. It is then not possible for (1) to be the case. It is also not possible for (2) to be the case, since the composition phase only stops when no more composition step can be found for the **CE**, and if (2) was the case, there would exist an object that could still imply some further aggregation of objects in **CE** when we stopped, which means that there would exist at least one more composition step. But then the composition phase would not have stopped.

Therefore the interpretation **I** found by the recognition process is a most specific one for the image if one exists. □

The recognition process being sound and complete, we can conclude that it is correct. □

Chapter 5

Conclusion

From a computational point of view, images give rise to very ambiguous information about the scene they depict. This ambiguity directly follows from the fact that the image underconstrains the scene depicted. Processing the intrinsic information included in an image, to obtain a description of the scene depicted, is then a very difficult task. In principle, a formal theory of computational vision should provide us with a definition of what it means for a computational vision system to conclude that an image depicts a particular scene. It should also serve as a guide for the implementation of more efficient systems, by providing us with the grounds for evaluating our current computational vision systems and indicate possible improvement in our approaches. It is clear that what we need are formal theories that guide our quest to achieve computational visual recognition.

Many of the aspects of the field of computational vision has not yet been formally defined. Even the formal foundations of what constitutes an "*interpretation*" of an image have not been paid much attention in the computer vision community.

The thesis that we presented and defended is that the meaning of the notion of

a symbolic interpretation of an image is clarified by viewing visual computational recognition as "theory formation".

This dissertation outlined a formal characterization, called "*Perceptual Reasoning*", that reformulates model-based approaches to computational vision. Perceptual Reasoning is based on the idea of "theory formation," and provides a clear and precise semantics of an interpretation for an image, and logical criteria as the basis for preferring one of possibly many interpretations.

In this concluding chapter, we summarize our progress toward this formal characterization, and highlight what we see as our contribution to the field of computational vision. We then suggest refinements and extensions as further research to be pursued in the future, to correct coarser elements and weaknesses of the presented research.

5.1 Summary and Contribution

We first present a summary of the dissertation by reviewing the concepts and ideas discussed and introduced in each chapters. This quick summary is then followed by a discussion of the formal theory presented which constitute our contribution to the field of computational vision.

5.1.1 Summary

In the introductory chapter of this dissertation, we briefly present the basic foundations of computational vision, mentioning concepts such as digital images, filtering, edge enhancement, etc. We also point out that the terms *low level-* and *high level-*

vision refers only to the extremities of a visual recognition "continuum," as opposed to being two distinct processes.

This very general introduction is followed by a statement of our thesis, some motivation for the need for a clear and precise semantics of what constitutes a valid image interpretation, and how one can be obtained.

The basic motivation is that this meaning (semantics) of an image interpretation, and how one can be obtained, can provide the formal grounds to evaluate existing model-based systems with respect to the criteria of soundness and completeness. It can also serve as guide lines for new and perhaps more efficient implementations.

Our second chapter examine existing control structures of computational vision. One of the guiding paradigms arising from a consensus of researchers of the field of computational vision is that contextual information should be used to ease visual recognition by a computer. Contextual information refers to the recognition of one object leading to the expectation of other objects. It is generally accepted that image understanding is almost impossible without such expectations. Model-based approaches to computational vision are founded on this idea of using contextual information. We point out the work of Mackworth [Mac78] who informally characterized the hypothesize-and-test paradigm of model-based approaches to computer vision in his *Cycle of Perception.*

We conclude chapter 2, by observing that intuitively, this cyclic way of proceeding with the analysis of images is very similar to the way scientists build theories to explain different phenomena. We then analogically relate different processes of the model-based approach to computational vision, such as "hypothesis invocation schemes," the "focus of attention," and the verification of the "expectations of an hypothesis," to the different steps of theory formation.

Chapter 3 present a formal characterization that synthesizes model-based approaches to computational vision, by viewing visual recognition as "theory formation". First order logic is chosen for the formal language of the characterization because of the formal treatments that are possible within it, and its clear semantics. From this characterization follows a clear and precise semantics of an interpretation for an image. Logical founded criteria that provides basis for ranking interpretations of an image are also provided. These criteria naturally follow from our characterization.

In chapter 4, we discuss the use of hierarchies of abstraction of prototypical objects for the representation of visual knowledge. Then, inspired by Haven's "Theory of Schema Labelling" [Hav85], we present a particular instance of Perceptual Reasoning that explicitly exploits the use of *composition* and *specialization* hierarchies for the representation of the visual knowledge. The outline of an algorithm for visual recognition is defined that specifies an incremental recognition process which exploits the contextual information explicitly encoded in the visual knowledge.

Using the train domain used by Havens, we present an example of this particular recognition process, and conclude with an informal argument of the correctness of this recognition process with respect to the semantics defined in the previous chapter.

5.1.2 Contribution

We view our formal theory formation characterization of model-based approaches to computational vision as a contribution to the field of computational vision for the following reasons:

First, the characterization provides a precise semantics of an image interpreta-

tion. This semantics describes what it means for a model-based vision system to come to a conclusion about what is the scene depicted by an image. The semantics presented is independent of implementation and can therefore be use to formally analyze existing vision systems with respect to soundness and completeness.

Second, this formal characterization provides logical criteria for preferring one interpretation over others. These criteria also provide a semantics for the "focusing of attention" that takes place in model-based approaches to computation vision.

Finally, the theory formation foundations of Perceptual Reasoning clarify the the motivation for the hypothesize-and-test control structure used in model-based recognition systems, by providing a control framework for different processes of model-based approaches to recognition.

We don't believe that the incremental recognition process presented in Chapter 4 provides a complete account of all the details of how model-based approaches to computational vision use contextual knowledge. We presented it only to show that, by following a particular programming methodology for the composition and specialization axioms, it is possible to precisely specify each step towards computational visual recognition. This demonstrates how the formal theory can be used to define new computational approaches to recognition.

5.2 Future Research

One possible refinement to the theory formation characterization of Chapter 3 is to formally define all the conditions for which some *primitives* may not be observable in the image. Such a formalization would require an extensive study of the semantics underlying occlusion and non-visibility due to the viewpoint, and maybe other such conditions.

For occlusion, one solution would be to permit the use of assumptions about occlusion to explain why a primitive cannot be observed, provided that we have proper evidence to believe that an object is in front of the object predicting this primitive. We already indicated that lower level cues like T-junctions could be used as such evidence of occlusion.

A formal characterization of non-visibility due to the viewpoint to explain that a predicted primitive can not be observed would rely on a precise study of perspective. Non-visibility due to the viewpoint has already been given a lot of attention from the computer graphics community; useful insight may be gained from that research.

The algorithm and the programming methodology that was imposed for composition and specialization axioms in chapter 4, could be improved by generalizing the algorithm to deal with more general forms of representation. This generalization would require attention to insure that we don't loose efficiency. There is a trade off between expressiveness and efficiency.

Finally, we would also like to use Perceptual Reasoning as a specification to design and implement a computer vision system for a practical application.

References

[BB82] D.H. Ballard and C.M. Brown. *Computer Vision.* Prentice-Hall, New Jersey, 1982.

[Bro81] R.A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17:285–348, 1981.

[Bro84] R.A. Brooks. *Model-Based Computer Vision.* UMI Research Press, Ann Arbor, Michigan, 1984.

[Bro86] R.A. Brooks. Model-based 3-d interpretation of 2-d images. In Pentland A.P., editor, *From Pixels to Predicates*, pages 299–321, Ablex Publishing Co., New Jersey, 1986.

[BT78] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. In Hanson A.R. and Riseman E.M., editors, *Computer Vision System*, pages 3–26, Academic Press, New York, 1978.

[BT80] H.G Barrow and J.M Tenenbaum. Interpreting line drawings as three-dimensional surfaces. In *Proceeding of the AAAI Conference*, pages 11–14, Stanford University, August 1980.

[CF82] P.R. Cohen and E.A. Feigenbaum. *The Handbook to Artificial Intelligence, Volume 3.* William Kaufmann, California, 1982.

[CM84] E. Charniak and D.V. McDermott. *Introduction to Artificial Intelligence.* Addison-Wesley, 1984.

[Fel75] J. Feldman. Bad-mouthing frames. In *Proceedings of the First Conference on Theoritical Issues in Natural Language Processing.*, pages 102–103, MIT, Cambridge, Massachusetts, June 10-13 1975.

[Gar86] H. Gardner. *The Mind's New Science.* Basics Books, New York, 1986.

[GFP86] R.G. Goebel, K. Furukawa, and D.L. Poole. Using definite clauses and integrity constraint as the basis for a theory formation approach to diagnostic reasoning. In *Proceedings of the Third International Conference on Logic Programming.*, pages 211–222, London, England, July 14-18 1986.

[GG87] S.D. Goodwin and R.G. Goebel. Applying theory formation to the planning problem. In F.M. Brown, editor, *The Frame Problem in Artificial Intelligence: Proceedings of the 1987 Workshop*, pages 207–232, Morgan Kaufmann, 1987.

[Gli82a] J. Glicksman. *A Cooperative Scheme for Images Understanding Using Multiple Sources of Information.* Technical Report TN 82-13, Department of Computer Science, University of British Columbia, 1982.

[Gli82b] J. Glicksman. A schemata-based system for utilizing cooperating knowledge sources in computer vision. In *Proceedings of the Fourth CSCSI Conference*, pages 33–40, Saskatoon, Canada, May 1982.

[Gli83] J. Glicksman. Using multiple information sources in a computational vision system. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1078–1080, Karlsruhe, W. Germany, Aug 1983.

[Gli84] J. Glicksman. Procedural adequacy in an image understanding system. In *Proceedings of the Fifth CSCSI Conference*, pages 44–48, University of Western Ontario, May 1984.

[Goa86] C. Goad. Fast 3-d model-based vision. In Pentland A.P., editor, *From Pixels to Predicates*, pages 371–391, Ablex Publishing Co., 1986.

[Goo87] S.D. Goodwin. *Representing Frame Axioms as Defaults.* Master's thesis, Computer Science Department, University of Waterloo, 1987.

[Hav85] W.S Havens. A theory of schema labelling. *Computational Intelligence*, 1(3-4):127–139, 1985.

[Hay77] P. Hayes. In defence of logic. In *Proceedings of the fifth International Joint Conference on Artificial Intelligence*, pages 559–565, MIT, Cambridge, August 1977.

[Hay81] P. Hayes. The logic of frames. In B.L. Weber and N. Nilsson, editors, *Readings in Artificial Intelligence*, pages 451–458, Morgan Kaufmann, 1981.

[Hem65] C.G. Hempel. *Aspects of Scientific Explanation and other Essays in the Philosophy of Science.* The Free Press, New York, 1965.

[HM83] W.S. Havens and A.K. Mackworth. Representing the visual world. *IEEE Computer*, 16(10):90–96, 1983.

[Hof86] W. Hoff. Surfaces from stereo. In *Proceedings of the International Conference on Patern Recognition*, pages 516–518, Paris, France, October 1986.

[HR78] A.R Hanson and E.M. Riseman. Visions: a computer system for interpreting scenes. In Hanson A.R. and Riseman E.M., editors, *Computer Vision System*, pages 303–333, Academic Press, 1978.

[HYI86] K. Sato H. Yamamoto and S. Inokuchi. Range imaging system based on binary image accumulation. In *Proceeding of the International Conference On Pattern Recognition*, pages 233–238, Paris, France, October 1986.

[Isr80] D. J. Israel. What's wrong with non-monotonic logic. In *Proceedings of AAAI Conference*, pages 99–101, Stanford University, 1980.

[Jac86] W.K. Jackson. *A Theory Formation Framework for Learning by Analogy.* Master's thesis, Department of Computer Science, University of Waterloo, 1986.

[Kan78] T. Kanade. Region segmentation: signal vs. semantics. In *Proceedings of the Forth International Joint Conference on Pattern Recognition*, pages 95–105, 1978.

[Low85] D.G. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, Hingham, MA., 1985.

[LWR85] S.W. Lu, A.K.C. Wong, and M. Rioux. Recognition of 3-d objects in range images by attributed hypergragh monomorphism and synthesis. In *The First International Federation of Automated Control Symposium on Robot Control*, pages 389–394, Barcelona, Spain, Nov. 1985.

[Mac77] A.K. Mackworth. On reading sketch maps. In *Proceedings of the seventh International Joint Conference on Artificial Intelligence*, pages 598–606, 1977.

[Mac78] A.K. Mackworth. Vision research strategy: black magic, methaphors, miniworld and maps. In Hanson A.R. and Riseman E.M., editors, *Computer Vision Systems*, pages 53–59, Academic Press, 1978.

[Mac87] A.K. Mackworth. *Adequacy Criteria for Visual Knowledge Representation.* Technical Report technical report 87-4, Department of Computer Science, University of British Columbia, 1987.

[Mar78] D. Marr. Representing visual information. In Hanson A.R. and Riseman E.M., editors, *Computer Vision Systems*, pages 61–80, Academic Press, 1978.

[Mar82] D. Marr. *Vision.* Freeman, San Francisco, CA, 1982.

[Min85] M. Minsky. A framework for representing knowledge. In Brachman R.J. and Levesque H.J., editors, *Readings in Knowledge Representation*, pages 245–262, Morgan Kaufmann, 1985.

[MNI78] T. Matsuyama M. Nagao and Y. Ikeda. Region extraction and shape analysis of aerial photographs. In *Proceeding of the International Conference On Pattern Recognition*, pages 620–628, Kyoto, Japan, November 1978.

[MNI79] T. Matsuyama M. Nagao and Y. Ikeda. Structural analysis of complex aerial photographs. In *Proceeding of the sixth International Joint Conference in Artificial Intelligence.*, pages 610–616, Tokyo, Japan, August 1979.

[Moo82] R.C. Moore. The role of logic in knowledge representation. In *Proceedings of the AAAI Conference*, pages 428–433, Carnegie Mellon University, August 1982.

[Mul85] J.A. Mulder. *Using Discrimination Graphs to Represent Visual Knowledge.* PhD thesis, Department of Computer Science, University of British Columbia, 1985.

[Oxf79] *Oxford Paperback Dictionary.* Oxford University Press, 1979.

[Pen86] A.P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–332, 1986.

[PGA86] D.L. Poole, R.G. Goebel, and R. Aleliunas. *THEORIST: A Logical Reasoning System for Defaults and Diagnosis.* Technical Report CS-86-06, Department of Computer Science, University of Waterloo, 1986. [to appear in Knowledge Representation, N. Cercone and G. McCalla (Eds.), Springer-Verlag].

[Poo86] D.L. Poole. *Default Reasoning and Diagnosis as Theory Formation.* Technical Report, Department of Computer Science, University of Waterloo, 1986.

[Pop58] K. Popper. *The Logic of Scientific Discovery.* Basic Book, 1958.

[QU78] W.V. Quine and J.S. Ullian. *The Web of Belief.* Random House, New York, 1978.

[Roc83] I. Rock. *The Logic of Perception.* MIT Press, Cambridge, Mass., 1983.

[Rub81] S. M. Rubin. Film:knowledge sources in vision. In *Proceedings of the International Joint Conference on Artificial Intelligence.*, page 1067, 1981.

[SGC78] L.K. Schubert, R.G. Goebel, and N.J. Cercone. *The Structure and Organization of Semantic Net for Comprehension and Inference.* Technical Report TR78-1, Department of Computing Science, University of Alberta, 1978.

[Tso84] J.K. Tsotsos. *Representational Axes and Temporal Cooperative Process.* Technical Report RCBV-TR-84-2, Department of Computer Science, University of Toronto, 1984.

[Tso85] J.K. Tsotsos. Knowledge organization and its role in representation and interpretation for time-varying data: the alven system. *Computational Intelligence*, 1(1):16–32, 1985.

[Won86] G.M. Wong. *Depiction and Domains in Visual Knowledge Representation.* Master's thesis, Department of Computer Science, University of British Columbia, 1986.