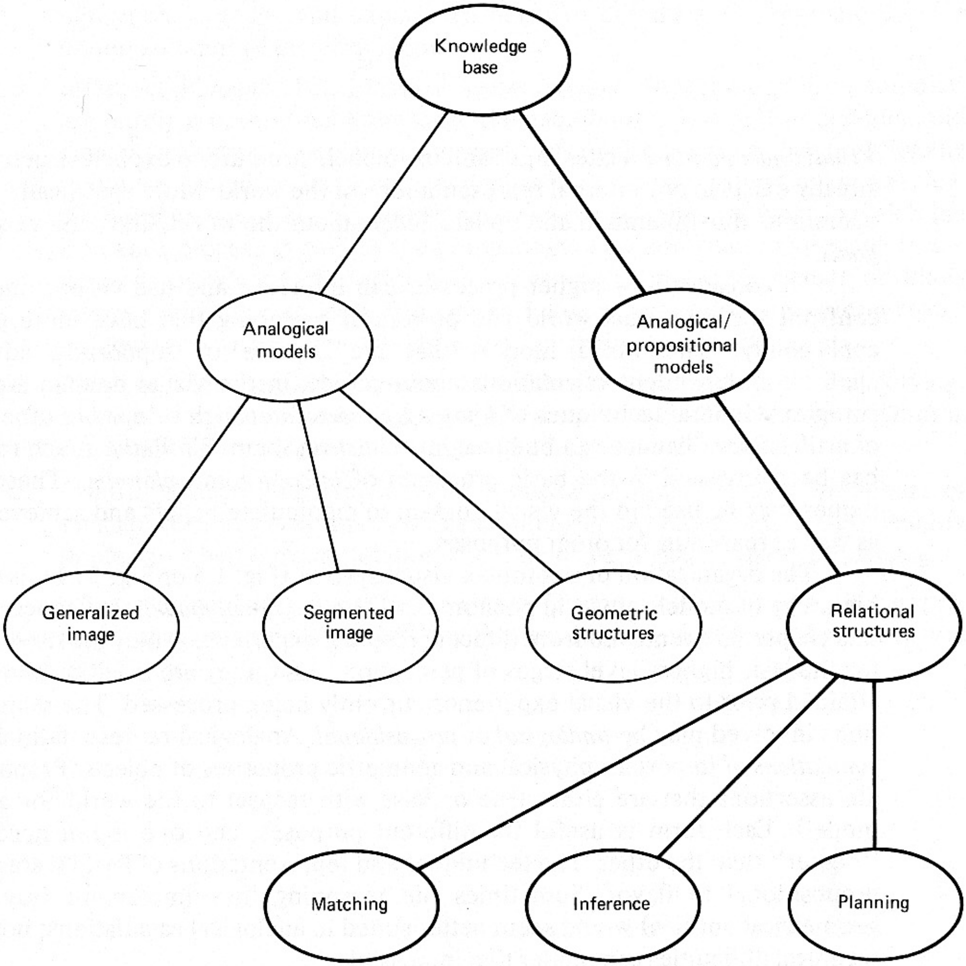
RELATIONAL STRUCTURES IV



Visual understanding relates input and its implicit structure to explicit structure that already exists in our internal representations of the world. More specifically, vision operations must maintain and update *beliefs* about the world, and achieve specific *goals.*

To consider how higher processes can influence and use vision, one must confront the nonvisual world and powers of reasoning that have more general applicability. The world models that are capable of supporting advanced application-dependent calculations about objects in the visual domain are quite complex. General techniques of *knowledge representation* developed in other fields of artificial intelligence can be brought to bear on them. Similarly, much research has been invested in the basic processes of *inference* and *planning.* These techniques may be used in the visual domain to manipulate beliefs and achieve goals, as well as reasoning for other purposes.

The organization of a complex visual system (Fig. 1.5 or Fig. 10.1), is a loose hierarchy of models of world phenomena. The *relational models* that concern us in this chapter are removed from direct perceptual experience—they are used mainly for the last, highest-level stages of perception. Also, they are used for knowledge attained prior to the visual experience currently being processed. The representations involved may be *analogical* or *propositional.* Analogical representations allow *simulations* of important physical and geometric properties of objects. Propositions are assertions that are either true or false with respect to the world (or a world model). Each form is useful for different purposes, and one is not necessarily "higher" than the other. The techniques and representations of Part IV are mainly propositional in flavor. Sometimes the reasoning they implement (say about geometrical entities) would seem better suited to analogical calculations; however, technical difficulties can render that impossible.

Part IV is concerned with techniques for making the "motivation" and "world view" of a vision system explicit and available. Such explicit models would be interesting from a scientific standpoint even if they were not directly useful. But explicitly available models are decidedly useful. They are useful to the system designer who desires to reconfigure or extend a system. They are useful to the system itself, which can use them to reason about its own actions, flexibly control its own resources in accordance with higher goals, dynamically change its goals, recover from mistakes, and so forth.

We organize the major topics of Part IV as follows.

1. Knowledge representation (Chapter 10). *Semantic nets* are an important technique for structuring complex knowledge, and can be used as a knowledge representation formalism in their own right.
2. Matching (Chapter 11). *Matching* puts a derived representation of an image into correspondence with an existing representation. This style of processing representations is more pronounced as domain-dependent knowledge, idiosyncratic goals, and experience begin to dominate the ultimate use (or understanding) of the visual input.
3. Inference (Chapter 12). Classical *logical inference* (a technique for manipulating purely propositional knowledge representations) is a well-understood and elegant reasoning technique. It has good formal properties, but occasionally seems restricted in its power to duplicate the range of human processing. *Extended inference* techniques such as *production systems* are those in which the inference process as well as the propositions may contribute materially to the derived knowledge. *Labeling* techniques can "infer" consistent or likely interpretations for an input from given rules about the domain. Inference can be used for both problem solving and belief-maintenance activity.
4. Planning (Chapter 13). *Planning* techniques are useful for problem solving, and are especially tailored to integrating vision with real-world *action*. Planning can be used for resource allocation and attentional mechanisms.
5. Control (Chapter 10; Appendix 2). Control *strategies* and *mechanisms* are of vital concern in any complex artificial intelligence system, and are particularly important when the computation is as expensive as that of vision processing.

Learning is missing from the list above. Disappointing as it is, at this writing the problem of learning is so difficult that we can say very little about it in the domain of vision.

Knowledge Representation and Use 10

10.1 REPRESENTATIONS

An internal representation of the world can help an intelligent system plan its actions and foresee their consequences, anticipate dangers, and use knowledge acquired in the past. In Part IV we investigate the creation, maintenance, and use of a *knowledge base*, an abstract representation of the world useful for computer vision. Chapter 1 introduced a layered organization for the knowledge base and divided its contents into "analogical" and "propositional" models. In this section we consider this high-level division more deeply.

The outside world is accessible to a computer vision program through the imaging process. Otherwise, the program is manipulating its internal representations, which should correspond to the world in understood ways. In this sense, the knowledge base of generalized images, segmented images, and geometric entities contains "models" of the phenomena in the world. Another more abstract sense of "model" is high-level, prior expectations about how the world fits together. Such a high-level model is often much more complex than the lower-level representations, often has a large "propositional" component, and is often manipulated by "inference-like" procedures. Explicit knowledge and belief structures are a relatively new phenomenon in computer vision, but are playing an increasingly important role.

The goals of this chapter are three.

1. To develop in more depth some issues of high-level models (Section 10.1).
2. To describe *semantic nets*—an important and general tool for both organizing and representing models (Sections 10.2 and 10.3).
3. To address issues of *control*, at both abstract and implementational levels (Section 10.4 augmented by Appendix 2).

10.1.1 The Knowledge Base—Models and Processes

Figure 10.1 shows the representational layers in the knowledge base as we have developed it through the book, and shows the place of important processes. This organization might be compared with that in [Barrow and Tenenbaum 1981].

The knowledge base organization is mirrored in the organization of the book. Parts I to III dealt with analogical models and their construction; Part IV is concerned with propositional and complex analogical models. In Chapters 11 to 13, the emphasis moves from the structure of models to the processes (matching, inference, and planning) needed to manipulate and use them.

The knowledge base should have the following properties.

- Represent analogical, propositional, and procedural structures
- Allow quick access to information
- Be easily and gracefully extensible
- Support inquiries to the analogical structures
- Associate and convert between structures
- Support belief maintenance, inference, and planning

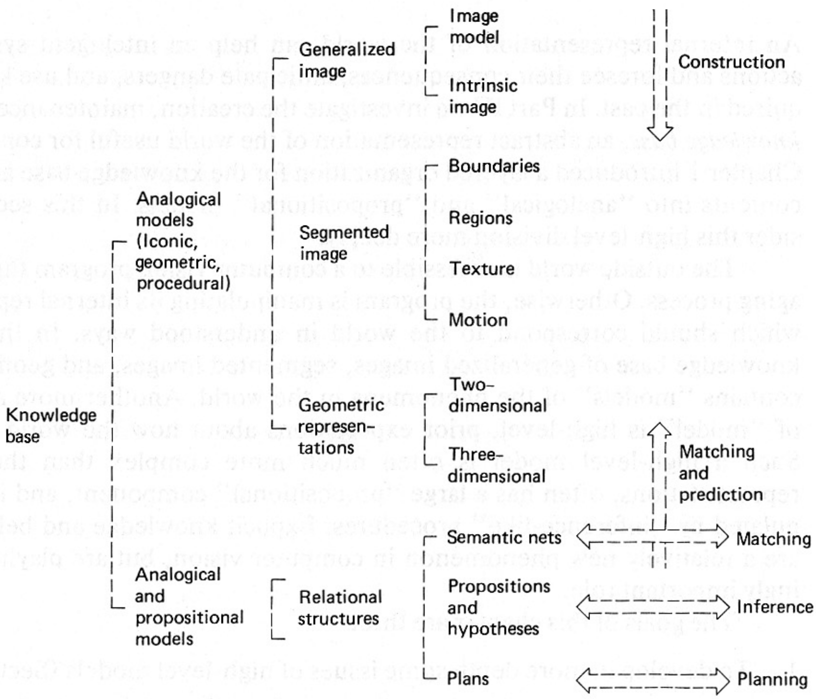


Fig. 10.1 The knowledge base and associated processes in a computer vision system.

The highest levels of the knowledge base contain both *analogical* and *propositional* models. Analogical tools do not exist for many important activities, and when they do exist they are often computationally intensive. A three-dimensional geometric modeling system for automatic manufacturing has very complex data structures and algorithms compared to their elegant and terse counterparts in a propositional model that may be used to plan the highest-level actions. In general it makes sense to do some computation at the analogical level and some at the propositional. This multiple-representation strategy seems more efficient than translating all problems into one representation or the other.

The computations in a vision system should be organized so that information can flow efficiently and unnecessary computation is kept to a minimum. This is the function of the *control* disciplines that allocate effort to different processes. Even the simplest biological vision systems exhibit sophisticated control of processing.

Constructive processes dominate the activity in building lower-level models, and *matching* processes become more important as prior expectations and models are brought into play. Chapter 11 is devoted to the process of matching.

We postulate that an advanced vision system is engaged in two sorts of high-level activity: *belief maintenance* and *goal achievement.* The former is a more or less passive, data-driven, background activity that keeps beliefs consistent and updated. The latter is an active, knowledge-driven, foreground activity that consists of planning future activities. Planning is a problem-solving and simulation activity that anticipates future world states; in computer vision it can determine how the visual environment is expected to change if certain actions are performed. Planning can occur with symbolic, propositional representations (Chapter 13) or in a more analogical vein with such simulations as trajectory planning [Lozano-Perez and Wesley 1979]. Planning is useful as an implementational mechanism even in contexts that are not analogous to human "conscious" problem solving [Garvey 1976]. Helmholtz likened the results of perception to "unconscious conclusions" [Helmholtz 1925]. Similarly even "primitive" vision processes (computer or biological) may use planning techniques to accomplish their ends.

Inference and planning are both classical subfields of artificial intelligence. Neither has seen much application in computer vision. Inference seems useful for belief maintenance. Extended inference can deal with inconsistent beliefs and with beliefs that are maintained with various strengths. We treat inference in Chapter 12. Applications of planning to vision [Garvey 1976; Bolles 1977] show good promise. Planning is treated in Chapter 13.

10.1.2 Analogical and Propositional Representations

Our division of the internal knowledge base into "analogical" and "propositional" reflects a similar division in theories of how human beings represent the world [Johnson-Laird 1980]. Psychological data are not compelling toward either pure theory; there are indications that human beings use both forms of representation. We introduce the division in this book because we find it conceptually useful in the following way. Low-level representations and processes tend to be purely analogical; high-level representations and processes tend to be both analogical and propositional.

Analogical representations have the following characteristics [Kosslyn and Pomerantz 1977; Shepard 1978; Sloman 1971; Kosslyn and Schwartz 1977, 1978; Waltz and Boggess 1979].

1. *Coherence.* Each element of a represented situation appears once, with all its relations to other elements accessible.
2. *Continuity.* Analogous with continuity of motion and time in the physical world; these representations permit continuous change.
3. *Analogy.* The structure of the representation mirrors (and may be isomorphic to) the relational structure of the represented situation. The representation is a description of the situation.
4. *Simulation.* Analogical models are interrogated and manipulated by arbitrarily complex computational procedures that often have the flavor of (physical or geometric) simulation.

Propositional representations have the following characteristics [Anderson and Bower 1973; Palmer 1975; Pylyshyn 1973].

1. *Dispersion.* An element of a represented situation can appear in several propositions. However, the propositions can be represented in a coherent manner by using semantic nets.
2. *Discreteness.* Propositions are not usually used to represent continuous change. However, they may be made to approximate continuous values arbitrarily closely. Small changes in the representation can thus be made to correspond to small changes in the represented situation.
3. *Abstraction.* Propositions are true or false. They do not have a geometric resemblance to the situation; their structure is not analogous to that of the situation.
4. *Inference.* Propositional models are manipulated by more or less uniform computations that implement "rules of inference" allowing new propositions to be developed from old ones.

Each sort of model derives its "meaning" differently; the distinctions are interesting, because they can point out weaknesses in each theory [Johnson-Laird 1980; Schank 1975; Fodor, et al. 1975]. Especially in computer implementations, the two representations only differ essentially in the last two points. It is often possible to transform one representation to another without loss of information.

Some examples are in order. A generalized image (Part I) is an analogical model: to find an object above a given object, a procedure can "search upward" in the image. An unambiguous three-dimensional model of a solid (Chapter 9) is analogical. It may be used to calculate many geometric properties of the solid, even those unimagined by the designer of the representation. A set of predicate calculus clauses (Chapter 12) is a propositional model. Closely related models can be used to solve problems and make plans [Nilsson 1971, 1980; Chapter 13].

A short digression: It is interesting that people do not seem to perform syllogistic inference (formal propositional deduction) in a "mechanical" way. Given two clauses such as "Some appliances are telephones" and "All telephones are black," we are much more likely to conclude "Some appliances are black" than the equally valid "Some black things are appliances." There is not a satisfying theory of the mental processes underlying syllogistic inference. An interesting speculation [Johnson-Laird 1980] is that inference is primarily done through analogical mental models (in which, for example, a population of individuals is conjured up and manipulated). Then syllogistic inference techniques may have arisen as a bookkeeping mechanism to assure that analogical reasoning does not "miss any cases."

10.1.3 Procedural Knowledge

Procedures as explicit elements in a model pose problems because they are not readily "understood" by other knowledge base components. It is very hard to tell what a procedure does by looking at its code.

In our taxonomy we think of "procedural" knowledge as being analogical. The sequential nature of a program's steps is analogous to an ordering of actions in time that can only be clumsily expressed in current propositional representations. Knowledge about "how-to" perform a complex activity is most propitiously represented in the form of explicit process descriptions. Descriptions not involving the element of time may be naturally represented as passive (analogical or propositional) structures.

There have been several attempts to organize chunks of procedural knowledge by associating with the procedure a description of what it is to accomplish. For example, procedural knowledge can be stored in the internal model structure (knowledge base) indexed under *patterns* that correspond to the arguments of the procedure. *Pattern-directed invocation* involves going to the knowledge base for a procedure that matches the given pattern, matching pattern elements to bind arguments, and invoking the procedure. Several advantages accrue in pattern-directed invocation, such as not having to know the "proper names" of procedures, only their descriptions (what they claim to do). Also, when several procedures match a pattern, one either gets nondeterminism or a chance to choose the best. Often system facilities include a procedure to run to choose the best procedure dynamically. Similar pattern matching is involved in resolution theorem provers and production systems (Chapter 12).

As an example, in a program to locate ribs in a chest radiograph [Ballard 1978], procedures to find ribs under different circumstances are attached to nodes in a mixed analogic and propositional model of the ribcage as shown in Fig. 10.2. Each procedure has an associated description which determines whether it can be run. For example, some programs require instances of neighboring ribs to be located before they can run, whereas others can run given only rudimentary scaling information. When invoked, each procedure tries to find a geometric structure corresponding to the associated rib in a radiograph. Instead of searching for ribs in a mechanical order, descriptors allow a choice of order and procedures and hence a

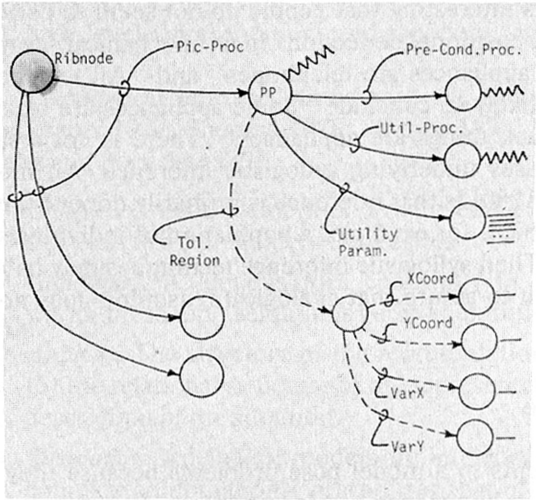


Fig. 10.2 A portion of a ribcage model (see text). Procedural attachment to a model is denoted by jagged lines.

more flexible, efficient and robust program (Appendix 2).

The representation and use of procedural knowledge is an important topic [Schank and Abelson 1977; Winograd 1975; Freuder 1975]. We expect it to be increasingly important for computer vision.

10.1.4 Computer Implementations

A computer implementation can (and often does) obscure the sharp divisions imposed by pure philosophical differences between analogical and propositional models. A propositional representation need not be an unordered set of clauses, but may have a coherent structure; the coherent versus dispersed distinction is thus blurred. A geometry theorem prover or a block-stacking program may manipulate diagrams or simulate physical phenomena such as gravitational stability and wobble in the manipulator [Gelernter 1963; Fahlman 1974; Funt 1977]. "Nonstandard inference" is an important tool that extends classical inference techniques. Although techniques such as production systems and relaxation labeling algorithms (Chapter 11) bear little superficial resemblance to predicate logic, both may be naturally used to manipulate propositional models.

Propositions may be implemented as procedures. If a proposition "evaluates" to true or false, it is perhaps most naturally considered a function from a world (or world model) to a truth value. This is not to say that all such functions exist or are evaluated when the proposition is "brought to mind"; perhaps "understanding a proposition" is like compiling a function and "verifying a proposition" is like evaluating it. The function may be implicit in an evaluation (inference) mechanism or more explicit, as in a "procedural" semantics such as that of the programming languages PLANNER and CONNIVER [Hewitt 1972; Sussman and McDermott 1972; Winograd 1978]. A proposition may thus be encoded as an (analogical!) procedural recipe for establishing the proposition. An example might be this representation of the fact "In California, Grass and Trees produce green regions."

```
(To-Establish (GreenRegion x)
  Establish (AND (InCalifornia())
    (OR (Establish (Grass x))
    (Establish (Trees x)))))
```

This might mean: To infer that x is a green region, establish that you are in California and then try to establish that x arose from grass. Should the grass inference fail, try to establish that x arose from trees. Since the full power of the programming language is available to an Establish statement, it can perform general computations to establish the inference.

The important point here: Rather than a set of clauses whose application must be organized by an interpreter, propositions may be represented by an explicit control sequence, including procedure calls to other programs. In the example, (Grass x) and (Trees x) may be procedures which have their own complicated control structures.

To say that in a computer "everything is propositions" is a truism; any program can be reduced to a Turing machine described by a finite set of "propositions" with a very simple rule of "inference." The issue is at what level the program should be described. A program may be doing propositional resolution theorem proving or analogical trajectory planning with three-dimensional models; it is not helpful to blur this basic functional distinction by appealing to the lowest implementational level.

10.2 SEMANTIC NETS

10.2.1 Semantic Net Basics

Semantic nets were first introduced under that name as a means of modeling human associative memory [Quillian 1968]. Since then they have received much attention [Nilsson 1980; Woods 1975; Brachman 1976; Findler 1979]. We are concerned with three aspects of semantic nets.

1. Semantic nets can be used as a data structure for conveniently accessing both analogical and propositional representations. For the latter their construction is straightforward and based solely on propositional syntax (Chapter 12).
2. Semantic nets can be used as an analogical structure that mirrors the relevant relations between world entities.
3. Semantic nets can be used as a propositional representation with special rules of inference. Both classical and extended inference can be supported, but it is a challenging enterprise to design net structure that provides the properties of formal logic [Schubert 1976; Hendrix 1979].