The Logic of Depiction

by

Raymond Reiter[†*] and Alan K. Mackworth[†**]

Technical Report 87-24

June 1987

[†]Fellow, Canadian Institute for Advanced Research

*Department of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 1A4

** Department of Computer Science
University of British Columbia
Vancouver, B.C.
Canada V6T 1W5

Abstract

We propose a theory of depiction and interpretation that formalizes image domain knowledge, scene domain knowledge and the depiction mapping between the image and scene domains. This theory is illustrated by specifying some general knowledge about maps, geographic objects and their depiction relationships in first order logic with equality.

An interpretation of an image is defined to be a logical model of the general knowledge and a description of that image. For the simple map world we show how the task level specification may be refined to a provably correct implementation by invoking model preserving transformations on the logical representation. In addition, we sketch logical treatments for querying an image, incorporating contingent scene knowledge into the interpretation process, occlusion, ambiguous image descriptions, and composition.

This approach provides a formal framework for analyzing existing systems such as Mapsee, and for understanding the use of constraint satisfaction techniques. It also can be used to design and implement vision and graphics systems that are correct with respect to the task and algorithm levels.

1. Introduction

Computational vision requires, no less than any other area of artificial intelligence, representations of knowledge that are complete, correct, flexible and efficient. In pursuit of that goal researchers have exploited a wide variety of knowledge representation schemes including grammars, semantic nets, programs, logics, schemas, rules, constraints and neural nets. McCarthy and Hayes (1969) proposed some adequacy criteria for knowledge representation schemes in general and used them to argue for a logical representation. Vision researchers have, by and large, ignored that suggestion. Clowes (1971) and Huffman (1971), for example, advocated a knowledge representation based on simple constraints in the scene domain, in a non-logical framework. Mackworth (1987b) argues that any adequate representation scheme for visual knowledge should satisfy various criteria of descriptive and procedural adequacy. In particular, the representation scheme must maintain the distinction between knowledge of the image and knowledge of the scene and it must carry information about the depiction relation: how objects in the scene domain appear in the image. The task specification must be precise. Only if the concept of an interpretation is precisely defined can we determine if an implementation correctly finds all and only the interpretations allowed by the general knowledge and the particular image. This paper provides a logical framework for a theory of depiction and interpretation that allows us to achieve those goals, and demonstrates its application in a simple world.

2. An Illustrative Specification: Mapsee's Sketch Maps

As an example of how one might logically specify the knowledge base for an image interpretation application, we focus on Mapsee, a longterm research project at the University of British Columbia designed to interpret hand drawn sketch maps of geographical regions.

The Mapsee project is a series of experiments in visual knowledge representation (Mulder et al, 1987). Mapsee-1 (Mackworth, 1977) used n-ary constraints in the scene domain and a network consistency constraint satisfaction algorithm, Mapsee-2 (Havens and Mackworth, 1983) adopted a schema representation of knowledge that was simplified and enhanced in Mapsee-3 (Mulder, 1986) and augmented with a hierarchical constraint satisfaction algorithm (Mackworth et al, 1985). These systems served as testbeds for new knowledge representation techniques and as useful artefacts in their own right, for example, acting as knowledge sources for the interpretation of satellite and aerial imagery, and as prototypes for more autonomous image understanding systems. However, since no precise definition of the notion of an interpretation has been provided and since much of the knowledge is procedurally encoded and distributed, it is not possible to determine if these programs are functioning correctly according to a formal specification of the task. One purpose of this paper is to provide a "logical reconstruction" of a fragment of the Mapsee project.

For expository purposes, we considerably simplify the kinds of image and geographic features which Mapsee deals with, as well as the kinds of knowledge it uses in image interpretation. As a further caveat, we emphasize that the following specification is appropriate for the world of sketch maps; other applications may require very different axioms and assumptions.

2.1. Specifying the Image Domain

We assume that there are just two kinds of image primitives - chains and regions - so that the taxonomy of image objects is given by Figure 1, which pictorially represents the following first order formulas[1]:

$(\forall x)\, image\text{-}object(x) \equiv chain(x) \;\vee\; region(x)$

$(\forall x) \neg (chain(x) \;\wedge\; region(x))$

In addition there are the following relationships which may hold between image primitives:

$tee(c,c')$ - chain c meets chain c' at a T-junction, as in Figure 2(a).

$chi\ (c,c')$ - chains c meets chain c' at a χ-junction, as in Figure 2(b).

$bounds(c,r)$ - chain c bounds region r, as in Figure 2(c).

$closed(c)$ - chain c is a simple closed figure, as in Figure 2(d).

$interior(c,r)$ - an interior of closed chain c is region r, as in Figure 2(e).

$exterior(c,r)$ - an exterior of closed chain c is region r, as in Figure 2(f).

[1] We denote image domain predicates using lower case characters and scene domain predicates in upper case.

A given image will consist of finitely many chains and regions, together with finitely many instances of the above relations. Mapsee makes the following:

Closure Assumption (Closed World Assumption (Reiter, 1978) for the Image Domain)

All image domain predicates are completely known. This closure assumption is logically specified by *closure axioms* of the form:

$$(\forall x)\, chain(x) \equiv x = i_1 \ \vee \ \cdots \ \vee \ x = i_m$$

$$(\forall x)\, region(x) \equiv x = i_1 \ \vee \ \cdots \ \vee \ x = i_n$$

$$(\forall x,y)\, tee(x,y) \equiv x = i_1 \ \wedge \ y = i_1{}' \ \vee \ \cdots \ \vee \ x = i_k \ \wedge \ y = i_k{}'$$

$$(\forall x,y)\, bounds(x,y) \equiv x = i_1 \ \wedge \ y = i_1' \ \vee \ \cdots \ \vee \ x = i_j \ \wedge \ y = i_j'$$

etc.

where the i and i' are all constants.

Example 2.1

Figure 3 shows a simple hand drawn sketch map with its chains and regions labeled by suitable constants. The closure axioms for this image are:

$$(\forall x)\, chain(x) \equiv x = c_1 \ \vee \ x = c_2 \ \vee \ x = c_3 \ \vee \ x = c_4 \ \vee \ x = c_5 \ \vee \ x = c_6$$

$$(\forall x)\, region(x) \equiv x = r_1 \ \vee \ x = r_2 \ \vee \ x = r_3 \ \vee \ x = r_4$$

$$(\forall x,y)\, tee(x,y) \equiv x = c_2 \wedge y = c_1 \vee x = c_2 \wedge y = c_3 \vee$$
$$x = c_4 \wedge y = c_5 \vee x = c_3 \wedge y = c_5$$

$$(\forall x,y)\, chi(x,y) \equiv x = c_3 \wedge y = c_4 \vee x = c_4 \wedge y = c_3$$

$$(\forall x,y)\, bounds(x,y) \equiv x = c_1 \wedge y = r_1 \vee x = c_2 \wedge y = r_1 \vee$$
$$x = c_3 \wedge y = r_1 \vee x = c_3 \wedge y = r_2 \vee$$
$$x = c_4 \wedge y = r_1 \vee x = c_4 \wedge y = r_2 \vee$$
$$x = c_5 \wedge y = r_1 \vee x = c_5 \wedge y = r_2 \vee$$
$$x = c_5 \wedge y = r_3 \vee x = c_6 \wedge y = r_3 \vee$$
$$x = c_6 \wedge y = r_4$$

$$(\forall x)\, closed(x) \equiv x = c_5 \vee x = c_6$$
$$(\forall x,y)\, interior(x,y) \equiv x = c_6 \wedge y = r_4 \vee x = c_5 \wedge y = r_3$$
$$(\forall x,y)\, exterior(x,y) \equiv x = c_5 \wedge y = r_1 \vee x = c_5 \wedge y = r_2 \vee x = c_6 \wedge y = r_3$$

In addition to the closed world assumption for the image domain, Mapsee also makes the

Unique Names Assumption (Reiter, 1980):

All image primitives (i.e. the chains and regions) are pairwise distinct. In other words if i and i' are different constants denoting image primitives, they denote different image primitives. Thus, the specification of the image domain includes the following **unique names axioms:**

$i \neq i'$ for all distinct constants i, i' mentioned in the closure axioms for *chain* and *region*.

Notice that we have been implicitly assuming suitable **type constraints** on the arguments of image predicates, e.g. that the first argument of *bounds* is a chain, and the second a region. We also want no constant mentioned in the closure axiom for *chain* to be mentioned in the closure axiom for *region;* otherwise, for any such constant i, both *chain(i)* and *region(i)* would hold, contradicting the taxonomic axiom $(\forall x)\neg(chain(x) \ \wedge \ region(x))$.

We make these two assumptions explicit by imposing the following simple syntactic requirements on the above closure axioms:

Coherence Requirements

C1. Each constant occurring in the closure axiom for *chain* is distinct from any occurring in the closure axiom for *region*.

C2. All constants mentioned in the closure axioms for *tee, chi* and *closed* are mentioned in the closure axiom for *chain*. If the closure axiom for *bounds* is

$$(\forall x,y)\, bounds(x,y) \equiv x = i_1 \ \wedge \ y = i'_1 \ \vee \cdots \vee \ x = i_m \ \wedge \ y = i'_m$$

then $i_1,\cdots,i_m$ are all mentioned in the closure axiom for *chain*, and $i'_1,\cdots,i'_m$ are all mentioned in the closure axiom for *region*. Similarly for the closure axioms for *interior* and *exterior*.

2.2. Specifying the Scene Domain

We assume that the taxonomy of scene objects is given by Figure 4, which pictorially represents the following first order formulas[2]:

$$(\forall x)\ SCENE\text{-}OBJECT(x) \equiv LINEAR\text{-}SCENE\text{-}OBJECT(x) \lor AREA(x)$$

$$(\forall x)\ \neg\ (LINEAR\text{-}SCENE\text{-}OBJECT(x) \land AREA(x))$$

$$(\forall x)\ LINEAR\text{-}SCENE\text{-}OBJECT(x) \equiv ROAD(x) \lor RIVER(x) \lor SHORE(x)$$

$$(\forall x)\ \neg\ (ROAD(x) \land RIVER(x))$$

$$(\forall x)\ \neg\ (ROAD(x) \land SHORE(x))$$

$$(\forall x)\ \neg\ (RIVER(x) \land SHORE(x))$$

$$(\forall x)\ AREA(x) \equiv LAND(x) \lor WATER(x)$$

$$(\forall x)\ \neg\ (LAND(x) \land WATER(x))$$

In addition to this taxonomic information, we assume the following general facts about the real world of roads, rivers etc.

(i) Rivers do not cross each other.

$$(\forall x,y)\ RIVER(x) \land RIVER(y) \supset \neg\ CROSS(x,y)$$

(ii) Shorelines form closed loops.

$$(\forall x)\ SHORE(x) \supset LOOP(x)$$

(iii) Rivers cannot form loops.

$$(\forall x)\ RIVER(x) \supset \neg\ LOOP(x)$$

---

[2] Recall that our convention is that scene domain predicates are denoted by upper case characters, and image domain predicates by lower case.

(iv) The inside area of a shoreline is land iff its outside is water; its inside is water iff its outside is land.

$$(\forall x,y,z)\ SHORE(x) \ \wedge\ INSIDE(x,y) \ \wedge\ OUTSIDE(x,z)$$
$$\supset LAND(y) \equiv WATER(z) \ \wedge\ WATER(y) \equiv LAND(z).$$

(v) If a road or a river is beside an area then that area is land.

$$(\forall x,y)\ BESIDE(x,y) \ \wedge\ (ROAD(x) \ \vee\ RIVER(x)) \supset LAND(y)$$

(vi) Rivers flow into other rivers, or into shores.

$$(\forall x)\ RIVER(x) \supset (\exists y) RIVER(y) \ \wedge\ JOINS(x,y) \ \vee$$
$$(\exists z) SHORE(z) \ \wedge\ JOINS(x,z)$$

Finally, we require the following axioms which restrict the scene predicates to scene objects only:

Scene Predicate Type Constraint Axioms

$(\forall x,y) CROSS(x,y) \supset SCENE\text{-}OBJECT(x) \ \wedge\ SCENE\text{-}OBJECT(y)$

$(\forall x) LOOP(x) \supset SCENE\text{-}OBJECT(x)$

$(\forall x,y) INSIDE(x,y) \supset SCENE\text{-}OBJECT(x) \ \wedge\ SCENE\text{-}OBJECT(y)$

$(\forall x,y) OUTSIDE(x,y) \supset SCENE\text{-}OBJECT(x) \ \wedge\ SCENE\text{-}OBJECT(y)$

$(\forall x,y) BESIDE(x,y) \supset SCENE\text{-}OBJECT(x) \ \wedge\ SCENE\text{-}OBJECT(y)$

$(\forall x,y) JOINS(x,y) \supset SCENE\text{-}OBJECT(x) \ \wedge\ SCENE\text{-}OBJECT(y)$

2.3. Specifying the Image-Scene Domain Mappings

In any given application, there will be relations which hold between the image and scene domains, for example, relations specifying how various three dimensional objects project onto the two dimensional image plane, or what kinds of scene objects are depicted by image objects. We refer to such relations as **mappings**, and represent them by appealing to a distinguished binary predicate $\Delta(i,s)$ meaning that image object i depicts scene object s.

In the case of Mapsee, the following assumptions are made:

(i) The world consists of image objects and scene objects, and these are disjoint categories.

$$(\forall x)\ image\text{-}object(x) \ \vee \ SCENE\text{-}OBJECT(x)$$
$$(\forall x) \neg\ (image\text{-}object(x) \ \wedge \ SCENE\text{-}OBJECT(x))$$

(ii) Every image object i depicts a unique scene object which we denote by $\sigma(i)$.

$$(\forall i)\ image\text{-}object(i) \supset SCENE\text{-}OBJECT(\sigma(i)) \ \wedge \ \Delta(i,\sigma(i)) \ \wedge \ [(\forall s)\Delta(i,s) \supset s = \sigma(i)]$$

(iii) Every scene object is depicted by a unique image object.

$$(\forall s)\ SCENE\text{-}OBJECT(s) \supset (\exists! i) image\text{-}object(i) \ \wedge \ \Delta(i,s)$$

Assumptions (ii) and (iii) are very strong. For example, (ii) forces the conclusion that a noise patch in the image depicts something real in the scene, while (iii) precludes occluded objects in the scene. Clearly, there are settings where these assumptions are unwarranted, where some of (ii), (iii) and the other image, scene and mapping axioms require more complex representations. We gloss over this issue for now but return to it briefly in Section 7.4 where we sketch a logical treatment of occlusion.

(iv) Depiction holds only between image and scene objects.

$$(\forall i,s)\ \Delta(i,s) \supset image\text{-}object(i)\ \wedge\ SCENE\text{-}OBJECT(s)$$

(v) Taxonomic mappings:

Regions in the image depict areas in the scene.

$$(\forall i,s)\ \Delta(i,s)\ \wedge\ region(i) \supset AREA(s)$$

Chains in the image depict linear scene objects in the scene.

$$(\forall i,s)\ \Delta(i,s)\ \wedge\ chain(i) \supset LINEAR\text{-}SCENE\text{-}OBJECT(s)$$

(vi) Relational mappings:

Tee relations in the image depict join relations in the scene, and vice versa.

$$(\forall i_1,i_2,s_1,s_2)\ \Delta(i_1,s_1)\ \wedge\ \Delta(i_2,s_2) \supset tee(i_1,i_2) \equiv JOINS(s_1,s_2)$$

Similarly, for the other image relations (Figure 2) and their corresponding scene relations:

$$(\forall i_1,i_2,s_1,s_2)\ \Delta(i_1,s_1)\ \wedge\ \Delta(i_2,s_2) \supset chi(\mathrm{i}_1,\mathrm{i}_2) \equiv \mathrm{CROSS}\,(\mathrm{s}_1,\mathrm{s}_2)$$

$$(\forall i_1,i_2,s_1,s_2)\ \Delta(i_1,s_1)\ \wedge\ \Delta(i_2,s_2) \supset bounds(i_1,i_2) \equiv BESIDE(s_1,s_2)$$

$$(\forall i,s)\ \Delta(i,s) \supset closed(i) \equiv LOOP(s)$$

$$(\forall i_1,i_2,s_1,s_2)\ \Delta(i_1,s_1)\ \wedge\ \Delta(i_2,s_2) \supset interior(i_1,i_2) \equiv INSIDE(s_1,s_2)$$

$$(\forall i_1,i_2,s_1,s_2)\ \Delta(i_1,s_1)\ \wedge\ \Delta(i_2,s_2) \supset exterior(i_1,i_2) \equiv OUTSIDE(s_1,s_2)$$

3. What Is an Interpretation?

In general, not simply for our sketch map example, a logical specification of the relevant knowledge and underlying assumptions for an image understanding application will consist of:

(i) **Image axioms:** an axiomatization of the image domain

(ii) **Scene axioms:** an axiomatization of the scene domain, and

(iii) **Mapping axioms:** an axiomatization of the mappings between the image and scene domains.

With such an axiomatization in hand, we can provide a formal definition of an interpretation as follows:

> An **interpretation** of an image is a model of the image, scene and mapping axioms.

We use the term "model" here in its strict logical sense (Mendelson, 1964)[3].

At this point it is appropriate to say a few words about computational issues. Determining the models of an arbitrary set of first order axioms is a wildly impractical task. To begin, it is undecidable in general whether such a set of formulas even has a model. Moreover, there may be infinitely many models. Is there anything special about vision which precludes these problems?

---

[3] The term "interpretation" has a logical meaning (Mendelson, 1964) which differs from our use of the word. Since we are grounding high level vision in logic, there is a risk of terminological confusion. Since "interpretation" is so firmly entrenched in the computational vision literature, we choose to continue use of the term in this paper. We emphasize that its use does not refer to its logical meaning.

At this stage of our research we can only speculate. The most promising observation is that an image is finite. There are just finitely many primitive image objects and relations between objects. Provided the depiction relation allows for just finitely many scene objects corresponding to the image primitives, then all quantifiers will have finite range. As we shall see, this is the case for our sketch map domain. Whenever this is the case, quantified formulas reduce to propositional ones and image interpretations are all computable. It is unclear just how general this observation is. Very likely a variety of vision tasks must be formalized before some general principles emerge regarding decidability issues.

4. Some Results Derivable from Mapsee's Axiomatization

Let MAP-AXIOMS be those axioms specified above for our simplified Mapsee domain, namely the image axioms, the scene axioms, and the mapping axioms. In this section, we state various logical consequences of these axioms which will simplify the process of computing the interpretations for a hand-drawn sketch map. We omit the proofs, which are contained in Appendix A.

Notation

Whenever MAP-AXIOMS entails a closure formula of the form

$$(\forall x_1)\cdots(\forall x_n)P(x_1,\cdots,x_n) \equiv x_1 = t_1^1 \wedge \cdots \wedge x_n = t_n^1$$
$$\vee \cdots \vee \; x_1 = t_1^m \wedge \cdots \wedge x_n = t_n^m$$

where the t_i^j are all terms, then $|P|$ denotes $\{(t_1^1,\cdots,t_n^1), \cdots, (t_1^m,\cdots,t_n^m)\}$.

Result 1 (Closure on image objects)

$$MAP\text{-}AXIOMS \models (\forall x)\, image\text{-}object(x) \equiv \bigvee_{i \in |chain| \cup |region|} (x = i)$$

Result 2 (Closure for Δ)

$$MAP\text{-}AXIOMS \models (\forall x,y)\Delta(x,y) \equiv \bigvee_{i \in |image\text{-}object|} (x = i \wedge y = \sigma(i))$$

Result 3 (Uniqueness of all objects)

If $I_m, I_n \in |image\text{-}object|$,

1. $MAP\text{-}AXIOMS \models \sigma(I_m) \neq \sigma(I_n)$ when $m \neq n$

2. $MAP\text{-}AXIOMS \models \sigma(I_m) \neq I_n$

3. $MAP\text{--}AXIOMS \models I_m \neq I_n$ when $m \neq n$

Result 4 (Closure for scene objects)

$$MAP\text{--}AXIOMS \models (\forall s) SCENE\text{--}OBJECT(s) \equiv \bigvee_{i \in |image\text{--}object|} (s = \sigma(i))$$

Result 5 (Domain closure)

$$MAP\text{--}AXIOMS \models (\forall x)[\bigvee_{i \in |image\text{--}object|} (x = i \ \vee \ x = \sigma(i))]$$

Result 6 (Closure for linear scene objects and areas)

1. $MAP\text{--}AXIOMS \models (\forall s) LINEAR\text{--}SCENE\text{--}OBJECT(s) \equiv \bigvee_{i \in |chain|} (s = \sigma(i))$

2. $MAP\text{--}AXIOMS \models (\forall s) AREA(s) \equiv \bigvee_{i \in |region|} (s = \sigma(i))$

Result 7 (Closure for Scene Domain Relations)

1. $MAP\text{--}AXIOMS \models (\forall x,y) JOINS(x,y) \equiv \bigvee_{(i,i') \in |tee|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

2. $MAP\text{--}AXIOMS \models (\forall x,y) CROSS(x,y) \equiv \bigvee_{(i,i') \in |chi|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

3. $MAP\text{--}AXIOMS \models (\forall x,y) BESIDE(x,y) \equiv \bigvee_{(i,i') \in |bounds|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

4. $MAP\text{--}AXIOMS \models (\forall x) LOOP(x) \equiv \bigvee_{i \in |closed|} (x = \sigma(i))$

5. $MAP\text{--}AXIOMS \models (\forall x,y) INSIDE(x,y) \equiv \bigvee_{(i,i') \in |interior|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

6. $MAP\text{--}AXIOMS \models (\forall x,y) OUTSIDE(x,y) \equiv \bigvee_{(i,i') \in |exterior|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

5. Simplifying MAP-AXIOMS

We now show how the results of the previous section allow us to systematically eliminate from consideration many of the axioms of MAP-AXIOMS. This in turn will considerably simplify the task of determining all interpretations of an image, as we shall see in Section 6 below.

Let SIMP-AXIOMS consist of the following groups of formulas:

S1. The closure axioms for *tee, chi, bounds, closed, interior, exterior, chain* and *region* of Section 2.1, augmented by the closure formulas for *image-object,* Δ, *SCENE–OBJECT*, *LINEAR–SCENE–OBJECT*, *AREA*, *JOINS*, *CROSS*, *BESIDE*, *LOOP*, *INSIDE* and *OUTSIDE*, derived in the previous section.

S2. Unique names formulas of Result 3, together with the domain closure formula of Result 5.

S3. (i) For $i \in |image\text{-}object|$,

$\neg\, ROAD(i)$

$\neg\, RIVER(i)$

$\neg\, SHORE(i)$

$\neg\, LAND(i)$

$\neg\, WATER(i)$

(ii) For $s \in |AREA|$,

$\neg\, RIVER(s)$

$\neg\, ROAD(s)$

$\neg\, SHORE(s)$

$LAND(s) \lor WATER(s)$

$\neg LAND(s) \lor \neg WATER(s)$

(iii) For $s \in |LINEAR\text{–}SCENE\text{–}OBJECT|$,

$\neg LAND(s)$

$\neg WATER(s)$

(iv) For $s \in |LOOP|$,

$ROAD(s) \lor SHORE(s)$

$\neg ROAD(s) \lor \neg SHORE(s)$

$\neg RIVER(s)$

(v) For $s \in |LINEAR\text{–}SCENE\text{–}OBJECT| - |LOOP|$,

$ROAD(s) \lor RIVER(s)$

$\neg ROAD(s) \lor \neg RIVER(s)$

$\neg SHORE(s)$

S4. The following groups of formulas:

(i) For $(x,y) \in |CROSS|$,

$\neg RIVER(x) \lor \neg RIVER(y)$

(ii) For (x,y,z) such that $x \in |LOOP|$, $(x,y) \in |INSIDE|$ and $(x,z) \in |OUTSIDE|$,

$SHORE(x) \supset LAND(y) \equiv WATER(z)$

(iii) For $(x,y) \in |BESIDE|$ and $x \notin |LOOP|$,

$LAND(y)$

For $(x,y) \in |BESIDE|$ and $x \in |LOOP|$,

$ROAD(x) \supset LAND(y)$

(iv) For $x \in |LINEAR\text{-}SCENE\text{-}OBJECT| - |LOOP|$,

$$RIVER(x) \supset [\bigvee_{\{y \mid (x,y) \in |JOINS| \text{ and } y \notin |LOOP|\}} RIVER(y)]$$

$$\vee \; [\bigvee_{\{z \mid z \in |LOOP| \text{ and } (x,z) \in |JOINS|\}} SHORE(z)]$$

Proposition 1

MAP-AXIOMS and SIMP-AXIOMS are logically equivalent.

Proof:

See Appendix B. The details can safely be ignored on first reading.

In the next section we show how SIMP-AXIOMS may be used to compute interpretations of sketch maps.

6. Determining the Interpretations of a Map

It remains to compute the interpretations of a hand-drawn sketch map, which means, by the definition of Section 3, computing all models of MAP-AXIOMS, hence of SIMP-AXIOMS. All such models share the following properties:

1. Suppose $|image\text{-}object| = \{i_1,...,i_n\}$. By the domain closure and unique names formulas of S2, the universe of any such model consists of 2n pairwise unequal elements. If we denote the elements of this universe corresponding to $i_1,...,i_n$, $\sigma(i_1),...,\sigma(i_n)$ by themselves, then all models of SIMP-AXIOMS share the same universe $\{i_1,...,i_n,\sigma(i_1),...,\sigma(i_n)\}$ of pairwise unequal elements.
2. The closure formulas of S1 completely characterize their predicates. Accordingly each predicate with a closure axiom has the same extension in all models of SIMP-AXIOMS, and these extensions are known to us a priori. For example, $|BESIDE|$ is the extension of the predicate *BESIDE* common to all models of SIMP-AXIOMS.

The only predicates lacking closure formulas are *ROAD*, *RIVER*, *SHORE*, *LAND* and *WATER*. Thus, the models of SIMP-AXIOMS can differ from one another only in the extensions they assign to these predicates. It follows that the only formulas of SIMP-AXIOMS we need consider in computing these models are those of S3 and S4. Moreover, these are quantifier-free formulas, so the problem reduces to determining the set of all propositional models of a set of formulas of the propositional calculus. While this is in general an NP-hard problem, at least it is decidable and various algorithms are known (Bibel, 1981; Purdom, 1984;

Mackworth, 1987a).

We illustrate the result of this calculation with the example sketch map of Figure 3.

Example (The map of Figure 3)

All models share the same universe $\{c_1,...,c_6,r_1,...,r_4,C_1,...,C_6,R_1,...,R_4\}$ where C_i and R_j denotes $\sigma(c_i)$ and $\sigma(r_j)$ respectively. Table 1 summarizes the extensions common to all these models of the predicates with closure formulas in S1.

It remains to determine all models of S3 and S4 which, for this example, are the following groups of formulas:

S3(i) For $i \in \{c_1,...,c_6,r_1,...,r_4\}$,

$\neg\, ROAD(i)$

$\neg\, RIVER(i)$

$\neg\, SHORE(i)$

$\neg\, LAND(i)$

$\neg\, WATER(i)$

(ii) For $s \in \{R_1,...,R_4\}$,

$\neg\, RIVER(s)$

$\neg\, ROAD(s)$

$\neg\, SHORE(s)$

$LAND(s) \ \vee \ WATER(s)$

$\neg\, LAND(s) \ \vee \ \neg\, WATER(s)$

PREDICATE	EXTENSION (|PREDICATE|)
chain	c_1, c_2, c_3, c_4, c_5, c_6
region	r_1, r_2, r_3, r_4
tee	(c_2,c_1), (c_2,c_3), (c_4,c_5), (c_3,c_5)
chi	(c_3,c_4), (c_4,c_3)
bounds	(c_1,r_1), (c_2,r_1), (c_3,r_1), (c_3,r_2), (c_4,r_1), (c_4,r_2), (c_5,r_1), (c_5,r_2), (c_5,r_3), (c_6,r_3), (c_6,r_4)
closed	c_5, c_6
interior	(c_6,r_4), (c_5,r_3)
exterior	(c_5,r_1), (c_5,r_2), (c_6,r_3)
image–object	c_1, c_2, c_3, c_4, c_5, c_6, r_1, r_2, r_3, r_4
Δ	(c_1,C_1), (c_2,C_2), (c_3,C_3), (c_4,C_4), (c_5,C_5), (c_6,C_6), (r_1,R_1), (r_2,R_2), (r_3,R_3), (r_4,R_4)
LINEAR–SCENE–OBJECT	C_1, C_2, C_3, C_4, C_5, C_6
AREA	R_1, R_2, R_3, R_4
JOINS	(C_2,C_1), (C_2,C_3), (C_4,C_5), (C_3,C_5)
CROSS	(C_3,C_4), (C_4,C_3)
BESIDE	(C_1,R_1), (C_2,R_1), (C_3,R_1), (C_3,R_2), (C_4,R_1), (C_4,R_2), (C_5,R_1), (C_5,R_2), (C_5,R_3), (C_6,R_3), (C_6,R_4)
LOOP	C_5, C_6
INSIDE	(C_6,R_4), (C_5,R_3)
OUTSIDE	(C_5,R_1), (C_5,R_2), (C_6,R_3)
SCENE–OBJECT	C_1, C_2, C_3, C_4, C_5, C_6, R_1, R_2, R_3, R_4

Table 1. The Interpretations of Predicates with Closure Formulas

(iii) For $s \in \{C_1,...,C_6\}$

$\neg\, LAND(s)$

$\neg\, WATER(s)$

(iv) For $s \in \{C_5,C_6\}$

$ROAD(s) \lor SHORE(s)$

$\neg\, ROAD(s) \lor \neg\, SHORE(s)$

$\neg\, RIVER(s)$

(v) For $s \in \{C_1,...,C_4\}$

$ROAD(s) \lor RIVER(s)$

$\neg\, ROAD(s) \lor \neg\, RIVER(s)$

$\neg\, SHORE(s)$

S4(i) $\neg\, RIVER(C_3) \lor \neg\, RIVER(C_4)$

(ii) $SHORE(C_5) \supset LAND(R_3) \equiv WATER(R_1)$

$SHORE(C_5) \supset LAND(R_3) \equiv WATER(R_2)$

$SHORE(C_6) \supset LAND(R_4) \equiv WATER(R_3)$

(iii) $LAND(R_1)$

$LAND(R_2)$

$ROAD(C_5) \supset LAND(R_1)$

$ROAD(C_5) \supset LAND(R_2)$

$ROAD(C_5) \supset LAND(R_3)$

$ROAD(C_6) \supset LAND(R_3)$

$ROAD(C_6) \supset LAND(R_4)$

(iv) $RIVER(C_1) \supset false$

$RIVER(C_2) \supset RIVER(C_1) \vee RIVER(C_3)$

$RIVER(C_3) \supset SHORE(C_5)$

$RIVER(C_4) \supset SHORE(C_5)$

After a certain amount of simplification (which would require a propositional theorem prover in general) we obtain the following equivalent set of formulas:

S3(i) - as above.

For $s \in \{R_1,...,R_4\}$,

$\neg RIVER(s)$

$\neg ROAD(s)$

$\neg SHORE(s)$

$LAND(R_1)$

$LAND(R_2)$

$\neg WATER(R_1)$

$\neg WATER(R_2)$

For $s \in \{R_3,R_4\}$

$LAND(s) \vee WATER(s)$

$\neg LAND(s) \vee \neg WATER(s)$

S3(iii) - as above.

S3(iv) - as above.

For $s \in \{C_2,...,C_4\}$

$ROAD(s) \vee RIVER(s)$

$\neg ROAD(s) \vee \neg RIVER(s)$

$\neg\ SHORE(s)$

$ROAD(C_1)$

$\neg\ RIVER(C_1)$

$\neg\ SHORE(C_1)$

S4(i) - as above.

$SHORE(C_5) \supset WATER(R_3)$

$SHORE(C_6) \supset LAND(R_4) \equiv WATER(R_3)$

$ROAD(C_5) \supset LAND(R_3)$

$ROAD(C_6) \supset LAND(R_3)$

$ROAD(C_6) \supset LAND(R_4)$

$RIVER(C_2) \supset RIVER(C_3)$

$RIVER(C_3) \supset SHORE(C_5)$

$RIVER(C_4) \supset SHORE(C_5)$

It is a simple but tedious matter to determine all propositional models of these formulas; there are six of them, as summarized in Table 2. This means there are six possible interpretations of the original image.

PREDICATE	EXTENSION1	EXTENSION2	EXTENSION3
ROAD	$C_1, C_2, C_3, C_4, C_5, C_6$	C_1, C_2, C_3, C_4, C_5	C_1, C_2, C_3, C_4
RIVER			
SHORE		C_6	C_5, C_6
LAND	R_1, R_2, R_3, R_4	R_1, R_2, R_3	R_1, R_2, R_4
WATER		R_4	R_3

PREDICATE	EXTENSION4	EXTENSION5	EXTENSION6
ROAD	C_1, C_2, C_3	C_1, C_2, C_4	C_1, C_4
RIVER	C_4	C_3	C_2, C_3
SHORE	C_5, C_6	C_5, C_6	C_5, C_6
LAND	R_1, R_2, R_4	R_1, R_2, R_4	R_1, R_2, R_4
WATER	R_3	R_3	R_3

Table 2. The Six Interpretations of The Map of Figure 3.

The problem of determining all propositional models of these formulas can be formulated as a classical constraint satisfaction problem (CSP) (Mackworth, 1987a) in two different ways. First, the problem of satisfiability of a propositional conjunctive normal form formula, SAT, is a CSP in which each atom is a variable with domain {true,false} and each clause is a constraint on the values of the atoms in the clause. In an alternative formulation, there are ten variables $\{C_1,...,C_6,R_1,...,R_4\}$. For the variables $\{C_1,...,C_6\}$ the domain of possible values is $\{ROAD,RIVER,SHORE\}$; for the variables $\{R_1,...,R_4\}$ the domain of possible values is $\{WATER,LAND\}$. Each propositional formula corresponds to a constraint (either unary, binary or ternary) on the sets of possible values allowed for the variables mentioned in the formula. Although, in general, CSP's are NP-hard there are several efficient approximation algorithms that may be useful. Network consistency approximation algorithms have been developed and used extensively in the Mapsee project (Mulder et al, 1987).

In connection with implementing an image interpretation system, notice that the general form of SIMP-AXIOMS of Section 5, specifically the formula groups S3 and S4, strongly suggests the use of a relational database system (Maier, 1983). Predicates like *CROSS*, *LOOP* etc. can be naturally viewed as relations, and $|CROSS|$, $|LOOP|$ etc. as their coresponding relational tables. For computations involving these tables, we can use the relational algebra which was designed specifically for the manipulation of such tables (Maier, 1983, Chapter 2). For example by appealing to the join operator of the relational algebra, the formula group S4(ii) may be expressed as:

For $(x,y,z) \in |LOOP| \bowtie_{1,1} (|INSIDE| \bowtie_{1,1} |OUTSIDE|)$,

$$SHORE(x) \supset LAND(y) \equiv WATER(z)$$

where $\bowtie_{i,j}$ indicates that the join is taken over the i-th and j-th columns of the first and second operands respectively of the join operator.[4]

By appealing to relational database systems in this way, computational vision can exploit the efficient storage, retrieval, and special purpose hardware of current and future database technologies. This can be especially important for vision applications since the relational tables obtained from complex images are likely to be quite large.

In connection with databases and vision, it is interesting to note that Bibel (1987) proposes solving constraint satisfaction problems by means of the relational algebra. As we have just seen, SIMP-AXIOMS leads to a constraint satisfaction problem whose solution yields all interpretations of a sketch map. We therefore have the prospect of relational databases playing a major implementation role in high level vision.

[4] The reader unfamiliar with the relational algebra can safely ignore this example. The important point is that the relational algebra provides operators for manipulating relational tables and that these have been implemented and optimised in current relational database systems.

7. Some Additional Features of a Logic of Depiction

We have emphasized that a logical foundation for high level vision provides a rigorous definition for the concept of an interpretation of an image. We have also demonstrated how logic can be used to refine a logical specification of an interpretation task to an algorithmic realization of this task. There are, however, other important advantages of a logical perspective. We sketch some of these here.

7.1. Incorporating Contingent Knowledge

In our axiomatization of hand-drawn sketch maps, the scene axioms of Section 2.2 reflected general knowledge of the scene domain. These axioms were fixed in advance and, with the help of the other axioms, were refined to the groups of propositional formulas S3 and S4 of SIMP-AXIOMS. These formulas are used to determine all interpretations of a given image.

It often happens, however, that *contingent* knowledge is available about a *particular* scene. Such knowledge is not universal to all scenes, nor can it be anticipated in advance. For example, we may know a priori something about the geographic region depicted by a particular sketch map, perhaps that the area contains a river with two tributaries, and it flows into a shore. This item of contingent knowledge is an additional constraint on the possible interpretations of the map, and must be exploited in computing these. The particular fact has the following logical representation:

$$(\exists r,s)\ RIVER(r) \wedge SHORE(s) \wedge JOINS(r,s) \wedge$$
$$(\exists r_1,r_2)\ RIVER(r_1) \wedge RIVER(r_2) \wedge r_1 \neq r_2 \wedge$$
$$JOINS(r_1,r) \wedge JOINS(r_2,r).$$

Conceptually, to accommodate this new information, we need only add it to MAP-AXIOMS and find all models of the resulting formulas. Computationally, because MAP-AXIOMS must be refined to SIMP-AXIOMS, the contingent knowledge must similarly be refined. For the example at hand, it is straightforward to carry out this refinement using the methods of Section 5. In fact, it could be automated for arbitrary contingent scene formulas. We obtain the formula

$$\bigvee_{(r,s) \in |JOINS|} [RIVER(r) \wedge SHORE(s) \wedge$$
$$\bigvee_{\{(r_1,r_2) | r_1 \neq r_2 \text{ and } (r_1,r) \in |JOINS| \text{ and } (r_2,r) \in |JOINS|\}} RIVER(r_1) \wedge RIVER(r_2)]$$

This can be added to SIMP-AXIOMS, and interpretations computed as before.

It is clear in general how contingent knowlege can be accommodated by a logical approach to high level vision, at least conceptually. One merely augments the axiomatization with the contingent facts. The interpretations of an image are the models of the enlarged axiom set. Computationally realizing this approach is another matter entirely. Such contingent scene knowledge must be transformed in exactly the same way as general scene knowledge as a first step in computing the interpretations, and these transformations must be algorithmically determined. For our simple sketch map world, specifying these transformations would be relatively straightforward, although we have not done so in this paper.

For more general settings, the problem of automatically accommodating contingent knowledge remains a future research topic.

7.2. Querying an Image

In many applications one is not concerned with finding some or all interpretations of an image. Rather, one is concerned with determining whether some property of the scene is depicted in a given image. For example, in our map world, we might wish to know whether part of what the image depicts is a road leading to a shore. Formally, this query is

$$(\exists r,s)\ ROAD(r)\ \wedge\ SHORE(s)\ \wedge\ JOINS(r,s).$$

In general, a query Q can be any formula. If $AXIOMS$ is a set of formulas formalizing the application under consideration, the query has answer "yes" provided it is true in all interpretations of the image, i.e. provided

$$AXIOMS\ \models\ Q.$$

Q has answer "no" provided it is false in all interpretations of the image, i.e. provided

$$AXIOMS\ \models\ \neg\ Q.$$

Otherwise, its answer is "possibly", which is to say it is true in some, but not all interpretations of the image.

One approach to answering a query is to compute all interpretations of the image, then determine the truth values of Q in each such interpretation. The obvious problem with such an approach is that it is completely bottom up; the query does not participate in the computation of interpretations. If answering the query requires just a few image properties, or involves only a small local region of the image, we can hope to do better than a generate and test algorithm. The natural approach is to invoke a theorem prover, which attempts to derive one or both of Q and $\neg Q$ using *AXIOMS* as premises. Notice, however, that just as was the case for accommodating contingent knowledge, the axioms to be used for image interpretation will be some refined version of the original specification. In our map world, SIMP-AXIOMS is such a refinement of MAP-AXIOMS. The example query above would also have to be similarly refined to the equivalent

$$\bigvee_{(r,s) \in |JOINS|} ROAD(r) \wedge SHORE(s)$$

prior to a theorem proving computation with SIMP-AXIOMS as premises. Moreover, the theorem to be proved should be instrumental in guiding the search for its proof, so some mechanism will be required analogous to the set of support strategy in resolution theorem proving (Wos et al, 1965), or top down derivations in Prolog (Kowalski, 1979). Since one can expect that this final theorem proving task will frequently be propositional, it is likely to appeal to constraint satisfaction techniques. In this case, we shall require mechanisms whereby the theorem actively guides the search for solutions to a constraint satisfaction problem. Finally, when the answer to a query is "possibly", we shall normally want to determine those image interpretations in which the query is true. All these issues remain totally unexplored in the vision setting.

7.3. Accommodating Ambiguity in Image Descriptions

Ambiguity arises in vision in two fundamentally different ways. First, a well-specified image, for example the sketch map in Figure 3, may have multiple scene interpretations. This scene ambiguity is reflected in the fact that the image, scene and mapping axioms may have multiple models (six in the case of Figure 3). Second, the image itself may have multiple descriptions. Here we deal with this possibility.

The image axioms of Section 2.1 for our map world formalize the assumption that our information about the image is complete; the closure axioms state that we know all and only the instances of image relations like *tee* and *bounds*, while the unique names axioms provide complete information about the equality relation. This assumption of complete information is a gross simplification.

Consider Figure 5 where the result of imperfect segmentation or careless drawing leaves open the possibility of a *tee* or a *chi* in the image. This setting can easily be represented by the image axiom

$$tee(c_1,c_2) \ \vee \ chi(c_1,c_2)$$

Of course, we now lose the closure axioms for *tee* and *chi*. This in turn leads to the loss of closure axioms for *JOINS* and *CROSS* which will have repercussions for the simplifications of MAP-AXIOMS derived in Section 5. Exploring the consequences of such ambiguities in the image description remains an open problem.

Figure 6 illustrates a more interesting example of ambiguity in an image description, because it affects the treatment of the equality relation. The question is whether to treat chains c_1 and c_2 as a single continuous chain, in which case r_1 and r_2 must be distinct regions, or as two separate chains, in which case r_1 and r_2 are identical regions. We adopt the convention that $c_1 = c_2$ means that c_1 and c_2 define a single continuous chain i.e. that this single chain has two different *names*. Similarly with respect to the regions r_1 and r_2. This setting can now be formalized as follows:

$$(\forall x)\ chain(x) \equiv x = c_1 \ \vee \ x = c_2 \ \vee \ x = c_3$$

$$(\forall x)\ region(x) \equiv x = r_1 \ \vee \ x = r_2$$

$$c_1 = c_2 \equiv r_1 \neq r_2$$

$$c_1 \neq c_3 \quad c_2 \neq c_3 \quad c_i \neq r_j$$

Notice that closure axioms for *chain* and *region* are preserved. The unique names axioms of Section 2.1 are *not* preserved. Specifically, the image axioms do not contain the unique names axioms $c_1 \neq c_2$ and $r_1 \neq r_2$. In settings like this, where the full set of unique names axioms must be abandoned, an equality reasoner will be necessary for computing image interpretations. The consequences for vision of incomplete information about the equality relation remains an open problem.

It is precisely with respect to the specification of incomplete information that logic excels as a representation language. While the consequences of such incomplete axiomatizations may be far from obvious, there can be no question of just what it is about an image that is being formally specified. This is particularly important when the image description is ambiguous.

7.4. Occlusion

To this point our sketch map world admits only two dimensional scenes. For example, MAP-AXIOMS precludes occlusion. This results from the mapping axiom 2.3(iii):

$$(\forall s)\ SCENE\text{-}OBJECT(s) \supset (\exists ! i)\ image\text{-}object(i)\ \wedge\ \Delta(i,s).$$

To see why, consider Figure 7 which depicts a bridge passing over what might be a river or a road occluded by the bridge. If R denotes this occluded road or river, then $\Delta(c_1,R)$ and $\Delta(c_2,R)$; since $c_1 \neq c_2$ the uniqueness property of the above mapping axiom is violated.

To accommodate occlusions of this kind we must relax the above mapping to

$$(\forall s)\ SCENE\text{-}OBJECT(s) \supset (\exists i) image\text{-}object(i)\ \wedge\ \Delta(i,s).$$

One consequence of this is that we lose the unique names formulas $\sigma(I_m) \neq \sigma(I_n)$ when $m \neq n$ for scene objects (see the proof of Result 3(i)). But as we are about to see, this price must be paid anyway in order to properly formalize occlusions of this kind.

Following the approach of the previous section, if a linear scene object is occluded so that its image contains two distinct chains c_1 and c_2, we adopt the convention that $\sigma(c_1) = \sigma(c_2)$ means that the two chains depict one and the same scene object. Equivalently, $\sigma(c_1)$ and $\sigma(c_2)$ are two different names for the same scene object. With respect to Figure 7, $\sigma(c_1) = \sigma(c_2)$ means that there is a single scene object depicted by the two chains c_1 and c_2. We can formalize bridge occlusions by the following scene axiom:

$$(\forall b,s_1,s_2,l_1,l_2)\ BRIDGE(b,s_1,s_2)\ \wedge\ JOINS(l_1,s_1)$$
$$\wedge\ JOINS(l_2,s_2) \supset l_1 = l_2\ \wedge\ (ROAD(l_1)\ \vee\ RIVER(l_1))$$

Here, $BRIDGE(b,s_1,s_2)$ means that b is a bridge with sides s_1 and s_2.

Notice that this axiom forces us to abandon unique names for scene objects (Result 3(i)), much as the representation of ambiguous image descriptions of the previous section led to the rejection of some unique names for image objects. Notice also that the centrality of the equality relation for a proper treatment of occlusion is not unique to our analysis. Whenever Guzman's (1968) SEE program uses the back-to-back T's heuristic to link two regions in an image of a polyhedral scene it is, in effect, declaring that those regions depict a single surface.

We do not presume to have solved the occlusion problem. Also there may well be other reasons for weakening the mapping axiom 2.3(iii). Many scene objects may not appear at all in the image because they are at the wrong scale, outside the frame of the map, inappropriate to the theme of the map or are totally occluded by, for example, a legend. The ramifications of abandoning unique names for image and scene objects requires exploration, as does the weakening of the mapping axiom 2.3(iii). What does emerge clearly is the centrality of the equality relation for reasoning about and representing occlusion.

7.5. Complex Objects

In our treatment of sketch maps, we have considered only simple scene objects like roads and rivers, that is, objects with no component parts. Most vision settings involve complex objects consisting of aggregations of components which in turn may have components, etc. This observation has motivated the designers of several vision systems to incorporate *composition hierarchies* for the definition of complex objects (Brooks, 1981; Havens and Mackworth, 1983; Tsotsos, 1985).

We indicate how such complex objects may be defined in our logical setting. As an example, consider the concept of a river system which informally is a maximal collection of interconnecting rivers at least one of which flows into a shoreline. As in most treatments of composition in the vision literature, we appeal to a predicate $PART\text{–}OF(x,y)$ meaning that object x is a component of the more complex object y.

1. Every river r is part of a unique river system which we denote by $\rho(r)$:

$$(\forall r)\ RIVER(r) \supset RIVER\text{–}SYSTEM(\rho(r)) \ \wedge\ PART\text{–}OF(r,\rho(r)) \ \wedge \\ (\forall y)\ RIVER\text{–}SYSTEM(y) \ \wedge\ PART\text{–}OF(r,y) \supset y = \rho(r).$$

2. Definition of a river system:

$$(\forall x)\ RIVER\text{–}SYSTEM(x) \equiv [(\forall y)\ PART\text{–}OF(y,x) \supset RIVER(y)] \ \wedge \\ [(\forall r,p)\ RIVER(r) \ \wedge\ PART\text{–}OF(p,x) \ \wedge\ JOINS(r,p) \supset PART\text{–}OF(r,x)]$$

$$\wedge \ [(\exists s,z)\ SHORE(s)\ \wedge\ PART\text{-}OF(z,x)\ \wedge\ JOINS(z,s)]$$

3. Equality of river systems:

$$(\forall x,y)\ RIVER\text{-}SYSTEM(x)\ \wedge\ RIVER\text{-}SYSTEM(y)$$
$$\supset x = y \equiv [(\forall p)\ PART\text{-}OF(p,x) \equiv PART\text{-}OF(p,y)]$$

The introduction of complex objects into our sketch map world necessitates a number of minor changes to the axiomatization of Section 2. First, the scene domain taxonomy must be expanded to that of Figure 8. Second, all references to the predicate *SCENE–OBJECT* in Section 2 and the subsequent analysis must now be replaced by the predicate *SIMPLE–SCENE–OBJECT*. In all other respects, the image interpretation process of the preceeding sections remains the same, with one exception. When the axioms for river systems are taken into account, there may be fewer interpretations of an image. This is not too surprising since adding axioms may eliminate models. For example, under MAP-AXIOMS the image of Figure 9 has an interpretation in which $RIVER(C_1)$ and $RIVER(C_2)$, but the first two axioms above for river systems preclude this interpretation.[5]

In this section we have merely sketched how complex objects may be accommodated in a logic of depiction. The details of their logical representation remain to be worked out, as are algorithms for using such axioms in the interpretation process.

[5] We omit the proof of this, although it is straightforward. The proof makes use of the taxonomy of Figure 8. It also requires unique names axioms of the form $\rho(x) \neq \sigma(y)$ i.e. that complex scene objects are different than simple scene objects.

7.6. Characterizing Preferred Interpretations

On our account of high level vision, scene ambiguity is a purely logical property; multiple interpretations of an image arise from multiple models of the corresponding task axiomatization. The fact is, however, that frequently humans are unaware of all or even some of the ambiguities inherent in an image; certain interpretations are preferred over others.

In this paper we have not addressed the important problem of characterizing preferred interpretations. At this level one can expect domain specific probabilistic information to be significant, as well as psychological data. It is possible that purely logical considerations will be relevant. For example, certain preferred interpretations may well satisfy suitable extremal properties with respect to the space of all possible image interpretations. Such extremal properties arise in various formalizations of nonmonotonic reasoning (Reiter, 1987). In fact, since nonmonotonic reasoning is primarily concerned with plausible inferences, it is likely to play an important role in characterizing preferred (i.e. plausible) interpretations in vision.

Whatever considerations turn out to be relevant for characterizing preferred interpretations, we believe that a theory of high level vision must provide an account of all possible interpretations, not simply the psychologically preferred ones. In other words, it must provide a competence as well as a performance theory.

7.7. Graphics Applications

Although we have concentrated on the task of interpreting images, the vision problem, the logic of depiction can equally well be applied to the task of generating images, the graphics problem (Mackworth, 1983). One of the criteria of procedural adequacy is flexibility: the capacity of a knowledge representation scheme to support analysis and synthesis (Mackworth, 1987b).

If we adopt the simple axioms of Section 2.3 then, based on the assumption that each scene object is depicted by a unique image object, we can postulate a function $\iota(s)$ satisfying the axiom:

$$\begin{aligned}\forall s\ SCENE\text{–}OBJECT(s) \supset image\text{–}object(\iota(s)) \ \wedge \ \Delta(\iota(s),s) \\ \wedge \ [(\forall i)\Delta(i,s) \supset i = \iota(s)].\end{aligned}$$

Coordinate frame transformations including metrical constraints on the scale, location and orientation of image and scene objects can be specified by the depiction relation $\Delta(i,s)$ or, equivalently, by the function $\iota(s)$.

To generate an image of a scene, one computes all models of the general image, scene and mapping axioms and the particular scene description. If the scene description is consistent (internally and with respect to the general axioms) and denotes a unique scene then it is well-specified in the sense that it is neither anomalous nor ambiguous. In that case there would be but one model of the axioms which would specify a unique image.

One of the advantages claimed for logic-based systems such as Prolog is that there is often an element of "reversibility" in the definition of predicates: one can sometimes interchange the roles of input and output variables (Clocksin and Mellish, 1981). However, in practice, one finds that Prolog programs are usually designed to exploit a particular direction of procedural interpretation. The analogy carries through to the logic of depiction. Just as we manipulated the axioms to support an efficient interpretation process, one would have to manipulate the axioms to support an efficient generation process. Although the knowledge base may have been optimized for a particular direction of use, these optimizations are model-preserving, which ensures that the same knowledge underlies image interpretation and generation. This guarantees, for example, that interpretation and generation are correct inverses of each other with the qualification, of course, that interpretation is, in general, a one-to-many mapping, and generation is many-to-one.

Using this approach, there are advantages for building user-computer interfaces. If an applications program is manipulating a database of objects a graphical display representing a view of those objects could be maintained by a separate system built on the principles outlined here. While the user actually interacts with the graphical description in the image domain both the user and the applications program can interpret the effects of each other's graphical actions in the scene domain.

Without changing the scene domain rules one can easily change the image formatting and object depiction rules. For example, if the applications program and the user are manipulating sets and set inclusion relationships then a scene configuration could be depicted as a conventional tree (as in Figure 4) or the user may prefer to use Venn diagram conventions based on containment of closed regions (Wong, 1986). The separation of the image, scene and mapping knowledge encourages the design of modular and correct graphics systems that go beyond device independence to image domain independence.

8. Conclusion

We are far from having presented an adequate logic of depiction; however, we have outlined a formal treatment of a task level theory of model-based vision. General knowledge of the image domain, the scene domain and the depiction mapping can be expressed in first order logic with equality. An interpretation of a particular image is a logical model of the general knowledge and a description of that image. This perspective provides a purely logical account of scene ambiguity. It also provides a task level formulation of the interpretation problem. This specification is refined, through model-preserving transformations, to the equivalent problem of determining the satisfiability of a set of propositional formulas to which known constraint satisfaction algorithms can be applied.

This approach provides a framework for analyzing existing vision systems by a process of logical reconstruction. It also shows, for significant task domains, how to design and implement vision systems that are correct with respect to both the task and algorithm levels. The modular separation of the knowledge into three sets of axioms encourages portability and generality in the application of the theory of depiction to other domains. The theory is independent of the particular axiomatization exploited here as an example. Moreover, it has applications in intelligent computer graphics as well.

We have sketched logical approaches to the problems of contingent scene knowledge, image queries, image segmentation, occlusion, complex objects and preferred interpretations. These and many other issues of descriptive and procedural adequacy remain to be explored in depth.

9. Acknowledgements

We gratefully acknowledge several important discussions with Jan Mulder and Bill Havens. May Vink and Theresa Fong formatted the paper in their usual perfect ways. This work was supported by the Natural Sciences and Engineering Research Council of Canada and the Canadian Institute for Advanced Research.

Appendix A: Proofs of Results 1-7

In most cases, the proofs of the results of Section 4 are straightforward so we omit them. Instead, we normally simply list those formulas from which the results follow.

Result 1 (Closure on image objects)

$$MAP\text{-}AXIOMS \models (\forall x)\, image\text{-}object(x) \equiv \bigvee_{i \in |chain| \cup |region|} (x = i)$$

Proof:

Use the taxonomic axiom for *image-object* and the closure axioms for *chain* and *region* of Section 2.1.

Result 2 (Closure for Δ)

$$MAP\text{-}AXIOMS \models (\forall x,y)\Delta(x,y) \equiv \bigvee_{i \in |image\text{-}object|} (x = i \ \wedge \ y = \sigma(i))$$

Proof:

2.3(ii), 2.3(iv).

Result 3 (Uniqueness of all objects)

If $I_m, I_n \in |image\text{-}object|$,

1. $MAP\text{-}AXIOMS \models \sigma(I_m) \neq \sigma(I_n)$ when $m \neq n$

2. $MAP\text{-}AXIOMS \models \sigma(I_m) \neq I_n$

3. $MAP\text{-}AXIOMS \models I_m \neq I_n$ when $m \neq n$

Proof:

1. 2.3(ii), 2.3(iii), unique names axioms for the image domain (Section 2.1).

2. 2.3(ii), 2.3(i) (2nd axiom).

3. Unique names axioms for the image domain (Section 2.1).

Result 4 (Closure for scene objects)

$$MAP\text{-}AXIOMS \models (\forall s) SCENE\text{-}OBJECT(s) \equiv \bigvee_{i \in |image\text{-}object|} (s = \sigma(i))$$

Proof:

2.3(ii), 2.3(iii), Result 2.

Result 5 (Domain closure)

$$MAP\text{-}AXIOMS \models (\forall x)[\bigvee_{i \in |image\text{-}object|} (x = i \lor x = \sigma(i))]$$

Proof:

2.3(i) (first axiom), Result 1, Result 4.

Result 6 (Closure for linear scene objects and areas)

1. $MAP\text{-}AXIOMS \models (\forall s) LINEAR\text{-}SCENE\text{-}OBJECT(s) \equiv \bigvee_{i \in |chain|} (s = \sigma(i))$

2. $MAP\text{-}AXIOMS \models (\forall s) AREA(s) \equiv \bigvee_{i \in |region|} (s = \sigma(i))$

Proof:

1. The proof here is sufficiently complicated that we give some of its details.

$\Rightarrow$

Suppose *LINEAR–SCENE–OBJECT*(s) for some s. By the taxonomy for the scene domain, we conclude *SCENE–OBJECT*(s). By Result 4, $s = \sigma(I)$ for some $I \in |image\text{–}object|$. We prove $I \notin |region|$ from which, by Result 1, the result follows. For suppose, to the contrary, that $I \in |region|$ so that *region(I)*. By Result 2, $\Delta(I, \sigma(I))$. By the first taxonomic mapping 2.3(v) we obtain $AREA(\sigma(I))$, i.e. $AREA(s)$. But by the taxonomy for the scene domain, we cannot have both $AREA(s)$ and *LINEAR–SCENE–OBJECT*(s).

$\Leftarrow$

By Results 1 and 2, and the second taxonomic mapping 2.3(v).

2. The proof is essentially the same as for 1.

Result 7 (Closure for Scene Domain Relations)

1. $MAP\text{–}AXIOMS \models (\forall x,y)\, JOINS(x,y) \equiv \bigvee_{(i,i') \in |tee|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

2. $MAP\text{–}AXIOMS \models (\forall x,y)\, CROSS(x,y) \equiv \bigvee_{(i,i') \in |chi|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

3. $MAP\text{–}AXIOMS \models (\forall x,y)\, BESIDE(x,y) \equiv \bigvee_{(i,i') \in |bounds|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

4. $MAP\text{–}AXIOMS \models (\forall x)\, LOOP(x) \equiv \bigvee_{i \in |closed|} (x = \sigma(i))$

5. $MAP\text{–}AXIOMS \models (\forall x,y)\, INSIDE(x,y) \equiv \bigvee_{(i,i') \in |interior|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

6. $MAP\text{–}AXIOMS \models (\forall x,y)\, OUTSIDE(x,y) \equiv \bigvee_{(i,i') \in |exterior|} (x = \sigma(i) \ \wedge \ y = \sigma(i'))$

Proof:

1. Coherence Requirement C2 (Section 2.1) for *tee*, taxonomy for the image domain, 2.3(ii), relational mapping 2.3(vi) for *tee* and *JOINS*, Scene Predicate Type Constraint Axioms (Section 2.2).

2-6. Similar

Appendix B: Proof of Proposition 1

Proposition 1.

MAP-AXIOMS and SIMP-AXIOMS are logically equivalent.

Proof:

The proof is in two stages. We first establish the equivalence of MAP-AXIOMS and an intermediate set of axioms, INT-AXIOMS, and next the equivalence of INT-AXIOMS and SIMP-AXIOMS. To that end, let INT-AXIOMS consist of the formulas S1, S2 and S3 of SIMP-AXIOMS, together with

$\Sigma 4$. The general scene domain axioms 2.2(i) and 2.2(iv)-(vi) which, for convenience, we repeat here.

(i) $(\forall x,y) RIVER(x) \wedge RIVER(y) \supset \neg CROSS(x,y)$

(ii) $(\forall x,y,z) SHORE(x) \wedge INSIDE(x,y) \wedge OUTSIDE(x,z)$

$$\supset LAND(y) \equiv WATER(z) \wedge WATER(y) \equiv LAND(z)$$

(iii) $(\forall x,y) BESIDE(x,y) \wedge (ROAD(x) \vee RIVER(x)) \supset LAND(y)$

(iv) $(\forall x) RIVER(x) \supset (\exists y) RIVER(y) \wedge JOINS(x,y) \vee$

$$(\exists z) SHORE(z) \wedge JOINS(x,z)$$

Proof of Equivalence of MAP-AXIOMS and INT-AXIOMS

Recall that MAP-AXIOMS consists of the following groups of axioms:

M1. The image domain taxonomy.

M2. Closure axioms for *tee, chi, bounds, closed, interior, exterior, chain* and *region*.

M3. Unique names axioms for image constants.

M4. The scene domain taxonomy.

M5. The general scene domain axioms (2.2(i)-(vi)).

M6. Scene predicate type constraint axioms.

M7. Axioms defining the image-scene domain mappings (2.3(i)-(vi)).

We begin by showing that MAP-AXIOMS entails INT-AXIOMS. Trivially, MAP-AXIOMS entails each formula of the formula groups S1, S2 and $\Sigma 4$, by the results of Section 4. The formulas of S3(i)-(iii) are each entailed by M4 together with S1 and S2. We must prove that MAP-AXIOMS entails the formulas of S3(iv)-(v). To show S3(iv), begin by observing that the scene axiom $(\forall x)\ RIVER(x) \supset \neg\ LOOP(x)$ of M5 together with the closure formula for $LOOP$ entails $\neg\ RIVER(s)$ for $s \in |LOOP|$. This establishes the third formula of S3(iv). To establish the first two, it suffices to show that $|LOOP| \subseteq |LINEAR\text{–}SCENE\text{–}OBJECT|$. These then follow from the scene taxonomic axioms and the S3(iv) formulas $\neg\ RIVER(s)$ for $s \in |LOOP|$. Now $|LOOP| = \{\sigma(i) |\ i \in |closed|\}$. By the Coherence Requirement C2 on *closed*, $|closed| \subseteq |chain|$. Since $|LINEAR\text{–}SCENE\text{–}OBJECT| = \{\sigma(i) |\ i \in |chain|\}$, $|LOOP| \subseteq |LINEAR\text{–}SCENE\text{–}OBJECT|$. We have proved that MAP-AXIOMS entails S3(iv). To show this for S3(v) use the scene taxonomic formulas, the scene axiom $(\forall x)\ SHORE(x) \supset LOOP(x)$, and the closure formulas for $LOOP$ and $LINEAR\text{-}SCENE\text{-}OBJECT$.

It remains to show that INT-AXIOMS entails each axiom group of MAP-AXIOMS, which we do as follows:

1. S1 entails the taxonomic axiom

$$(\forall x)\, image\text{-}object(x) \equiv chain(x) \;\vee\; region(x)$$

and S1 and S2 together with the Coherence Requirement C1 entail the taxonomic axiom

$$(\forall x) \neg\, (chain(x) \;\wedge\; region(x)).$$

Hence INT-AXIOMS entails the formulas of axiom group M1.

2. Since M2 $\subseteq$ S1, S1 entails each closure axiom of M2.

3. Since M3 $\subseteq$ S2, S2 entails each unique names axiom of M3.

4. S1 entails the first scene domain taxonomic axiom. S1, S2 and the Coherence Requirement entail the second axiom. S1, S2 and S3 entail the remaining taxonomic axioms.

5. $\Sigma 4$ contains the general scene domain axioms 2.2(i), and (2.2)(iv)-(vi) and hence entails these axioms of M5. We must show that INT-AXIOMS entails axioms 2.2(ii) and 2.2(iii) of M5. To show 2.2(ii), suppose $SHORE(s)$ for some s. In 4 above we showed that INT-AXIOMS entails the scene taxonomic axioms, and hence entails $LINEAR\text{-}SCENE\text{-}OBJECT(s)$. By S3(v), $s \in |LOOP|$, so INT-AXIOMS entails $SHORE(s) \supset LOOP(s)$ and therefore axiom 2.2(ii). It remains to show that INT-AXIOMS entails 2.2(iii). This axiom is equivalent to $(\forall x)\; LOOP(x) \supset \neg\, RIVER(x)$ which, by the S1 closure formula for LOOP, is equivalent to $\bigwedge_{s \in |LOOP|} \neg\, RIVER(s)$, which is

entailed by S3(iv). We have now proved that INT-AXIOMS entails all the axioms of M5.

6. S1 entails each axiom of M6.

7. INT-AXIOMS entails each axiom of M7. We omit the details of verifying this claim, but give a few representative examples. The first axiom of 2.3(i) is entailed by the S1 closure formulas for *image–object* and *SCENE–OBJECT*, together with the S2 domain closure formula. The second axiom of 2.3(i) is entailed by the S1 closure formulas for *image–object* and *SCENE–OBJECT*, together with the S2 unique names axioms. 2.3(ii) is entailed by the S1 closure formulas for *image–object*, *SCENE–OBJECT* and Δ, together with S2. Similarly for 2.3(iii).

Proof of Equivalence of INT-AXIOMS and SIMP-AXIOMS

We show that S1, S2 and S3 together entail the equivalence of $\Sigma 4$ and S4.

Equivalence of $\Sigma 4$(i) and S4(i)

$\Sigma 4$(i) is equivalent to

$$(\forall x,y)\, CROSS(x,y) \supset \neg\, RIVER(x) \ \vee\ \neg\, RIVER(y)$$

Suppose the closure formula for *CROSS* in S1 is

$$(\forall x,y)\, CROSS(x,y) \equiv x = \alpha_1 \ \wedge\ y = \beta_1 \ \vee\ \cdots\ x = \alpha_m \ \wedge\ y = \beta_m$$

Substituting with this into the previous formula yields

$$(\forall x,y)(x = \alpha_1 \ \wedge\ y = \beta_1 \ \vee\ \cdots\ \vee\ x = \alpha_m \ \wedge\ y = \beta_m)$$

$$\supset \neg\, RIVER(x) \;\vee\; \neg\, RIVER(y)$$

This is equivalent to

$$\bigwedge_{(x,y)\,\in\,|CROSS|} (\neg\, RIVER(x) \;\vee\; \neg\, RIVER(y))$$

Equivalence of $\Sigma 4$(ii) and S4(ii)

By the domain closure formula of S2, universally quantified variables range over $|image\text{-}object| \cup |SCENE\text{-}OBJECT|$. S3 entails $\neg\, SHORE(x)$ when $x \notin |LOOP|$. The closure formulas for *INSIDE* and *OUTSIDE* of S2 together with the unique names formulas of S2 entail $\neg\, INSIDE(x,y)$ and $\neg\, OUTSIDE(x,z)$ when $(x,y) \notin |INSIDE|$ and $(x,z) \notin |OUTSIDE|$. It follows that S1 and S2 entail the equivalence of $\Sigma 4$(ii) and the following formulas:
For (x,y,z) such that $x \in |LOOP|$, $(x,y) \in |INSIDE|$ and $(x,z) \in |OUTSIDE|$,

$$SHORE(x) \supset LAND(y) \equiv WATER(z) \;\wedge\; WATER(y) \equiv LAND(z).$$

By the Coherence Requirement C2 and the closure formulas for *AREA*, *INSIDE* and *OUTSIDE*, any such $y,z \in |AREA|$. By S3(ii) any such y,z satisfy $LAND(y) \equiv \neg\, WATER(y)$, $LAND(z) \equiv \neg\, WATER(z)$. These two equivalences establish that of the pair of equivalences in the above group of formulas. Hence $\Sigma 4$(ii) is equivalent to S4(ii)

Equivalence of $\Sigma 4$(iii) and S4(iii)

We first prove that whenever $(x,y) \in |BESIDE|$ then $x \in |LINEAR\text{-}SCENE\text{-}OBJECT|$. To see this, notice that $|BESIDE| = \{(\sigma(i), \sigma(i')) \mid (i,i') \in |bounds|\}$. By the Coherence Requirement C2 on *bounds*, $i \in |chain|$ whenever $(i,i') \in |bounds|$. Since $|LINEAR\text{-}SCENE\text{-}OBJECT| = \{\sigma(i) | i \in |chain|\}$ we are done.

Now, as in the proof of the equivalence of $\Sigma4$(i) and S4(i), S1 entails the equivalence of $\Sigma4$(iii) and the following formulas:

For $(x,y) \in |BESIDE|$,

$$(ROAD(x) \ \vee \ RIVER(x)) \supset LAND(y)$$

Now there are two possibilities:

Case 1: $x \notin |LOOP|$

Then $x \in |LINEAR\text{-}SCENE\text{-}OBJECT| - |LOOP|$ and so by S3(v), $ROAD(x) \ \vee \ RIVER(x)$ so we derive $LAND(y)$.

Case 2: $x \in |LOOP|$

Then by S3(iv), $\neg\, RIVER(x)$ and we derive $ROAD(x) \supset LAND(y)$.

Equivalence of $\Sigma4$(iv) and S4(iv)

By the domain closure formula of S2, universally quantified variables range over $|image\text{-}object| \cup |SCENE\text{-}OBJECT|$, so S2 entails the equivalence of $\Sigma4$(iv) with the conjunction of the following formulas:

For $x \in |image\text{-}object| \cup |SCENE\text{-}OBJECT|$,

$$\begin{aligned} RIVER(x) \supset (\exists y) RIVER(y) \ \wedge \ JOINS(x,y) \ \vee \\ (\exists z) SHORE(z) \ \wedge \ JOINS(x,z) \end{aligned}$$

By S3(i), $\neg\, RIVER(i)$ for $i \in |image\text{-}object|$.

By S3(ii), $\neg\, RIVER(s)$ for $s \in |AREA|$.

By S3(iv), $\neg\, RIVER(s)$ for $s \in |LOOP|$.

Hence, S3 entails that the above group of formulas is equivalent to:

For $x \in |LINEAR\text{-}SCENE\text{-}OBJECT| - |LOOP|$,

$$RIVER(x) \supset (\exists y) RIVER(y) \wedge JOINS(x,y) \vee$$
$$(\exists z) SHORE(z) \wedge JOINS(x,z)$$

For some fixed $x \in |LINEAR\text{-}SCENE\text{-}OBJECT| - |LOOP|$, consider the subformula $(\exists y) RIVER(y) \wedge JOINS(x,y)$. The domain closure formula of S2 entails that this is equivalent to

$$\bigvee_{y \in |image\text{-}object| \cup |SCENE\text{-}OBJECT|} RIVER(y) \wedge JOINS(x,y)$$

For some fixed $y \in |image\text{-}object| \cup |SCENE\text{-}OBJECT|$ consider the subformula $RIVER(y) \wedge JOINS(x,y)$ in this disjunction. If $(x,y) \in |JOINS|$, then the closure formula for $JOINS$ in S1 entails $JOINS(x,y)$ so the subformula $RIVER(y) \wedge JOINS(x,y)$ is equivalent to $RIVER(y)$. If $(x,y) \notin |JOINS|$, then the closure formula for $JOINS$ in S1 together with the unique names formulas of S2 entail $\neg\, JOINS(x,y)$ in which case the subformula $RIVER(y) \wedge JOINS(x,y)$ is equivalent to false. Hence, S1 and S2 entail that for a fixed x, $(\exists y) RIVER(y) \wedge JOINS(x,y)$ is equivalent to $\bigvee_{(x,y) \in |JOINS|} RIVER(y)$. Finally, notice that by S3(iv), if $y \in |LOOP|$ then $\neg\, RIVER(y)$. Hence the previous disjunction is equivalent to

$$\bigvee_{\{y \mid (x,y) \in |JOINS| \text{ and } y \notin |LOOP|\}} RIVER(y)$$

A similar argument establishes that S1 and S2 entail the equivalence of $(\exists z) SHORE(z) \wedge JOINS(x,z)$ and

$$\bigvee_{\{z \mid z \in |LOOP| \text{ and } (x,z) \in |JOINS|\}} SHORE(z).$$

$\square$

References

Bibel, W. (1981). On matrices with connections. *J.ACM, 28* (4), 633-645.

Bibel, W. (1987). Constraint satisfaction from a deductive viewpoint. Artificial Intelligence Research Group Technical Report, The Technical University of Munich.

Brooks, R.A. (1981). Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence,* Vol. 17, pp. 285-348.

Clocksin, W.F. and Mellish, C.S. (1981). *Programming in Prolog.* Springer-Verlag, Berlin.

Clowes, M.B. (1971). On seeing things. *Artificial Intelligence,* 2, pp. 79-112.

Guzman, A. (1968). Decomposition of a visual scene into three-dimensional bodies. *Proc. AFIPS 1968 Fall Joint Computer Conference,* pp. 291-304.

Havens, W.S. and Mackworth, A.K. (1983). Representing knowledge of the visual world. *IEEE Computer, 16,* 10, 90-96.

Huffman, D.A. (1971). Impossible objects as nonsense sentences. In *Machine Intelligence* 6, B. Meltzer and D. Michie (eds.), American Elsevier, N.Y., pp. 295-323.

Kowalski, R. (1979). *Logic for Problem Solving.* North Holland, New York.

Mackworth, A.K. (1977). On reading sketch maps. *Proc. Fifth International Joint Conf. on Artificial Intelligence,* MIT, Cambridge, MA, pp. 598-606.

Mackworth, A.K. (1983). Recovering the meaning of diagrams and sketches. *Proc. Graphics Interface, '83,* Edmonton, pp. 313-317.

Mackworth, A.K., Mulder, J.A., and Havens, W.S. (1985). Hierarchical arc consistency: exploiting structured domains in constraint satisfaction problems. *Computational Intelligence,* 1(2), 71-79.

Mackworth, A.K. (1987a). Constraint satisfaction. In *Encyclopedia of Artificial Intelligence,* S. Shapiro (ed.), J. Wiley and Sons, N.Y., pp. 205-211.

Mackworth, A.K. (1987b). Adequacy criteria for visual knowledge representation. In *Computational Processes in Human Vision,* Z. Pylyshyn (ed.), Ablex Publishers, Norwood, N.J. (in press).

Maier, D. (1983). *The Theory of Relational Databases.* Computer Science Press, Rockville, Maryland.

McCarthy, J. and Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence* 4, B. Meltzer and D. Michie (eds.), American Elsevier, N.Y., pp. 463-502.

Mendelson, E. (1964). *Introduction to Mathematical Logic.* Van Nostrand, Princeton.

Mulder, J.A., Mackworth, A.K. and Havens, W.S. (1987). Knowledge structuring and constraint satisfaction: the Mapsee approach. Technical Report 1987CS-7, Div. Comp. Sci., Dalhousie Univ. and Technical Report 87-21, Dept. of Comp. Sci., Univ. of British Columbia.

Mulder, J.A. (1986). Using discrimination graphs to represent visual knowledge. Ph.D. Thesis, Dept. of Computer Science, Univ. of British Columbia.

Purdom, P.W. (1984). Solving satisfiability with less searching. *IEEE Trans. Pattern Analysis and Machine Intelligence, 6* (4), 510-513.

Reiter, R. (1978). On closed-world data bases. In *Logic and Data Bases,* H. Gallaire and J. Minker (eds.). Plenum Press, New York, pp. 55-76.

Reiter, R. (1980). Equality and domain closure in first-order data bases. *J.ACM, 27* (2), 235-249.

Reiter, R. (1987). Nonmonotonic reasoning. *Ann. Rev. Comput. Sci. 2,* 147-187, (in press).

Tsotsos, J.K. (1985). The role of knowledge organization in representation and interpretation of time-varying data: the ALVEN system. *Computational Intelligence, 1* (1), 16-32.

Wong, G. (1986). Depiction and domains in visual knowledge representation. M.Sc. Thesis, Dept. of Computer Science, Univ. of British Columbia.

Wos, L., Robinson, G.A. and Carson, D.F. (1965). Efficiency and completeness of the set of support strategy in theorem proving. *J.ACM, 12,* 536-541.

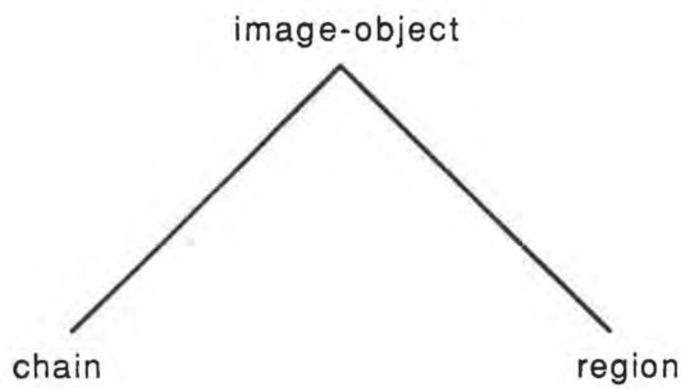


Figure 1. An image domain taxonomic hierarchy

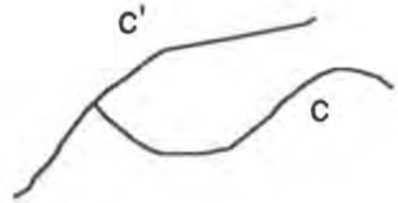
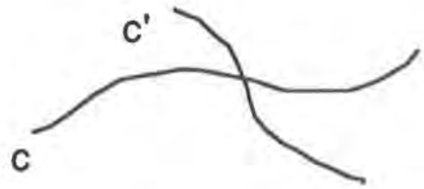
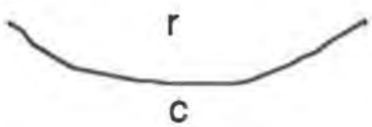
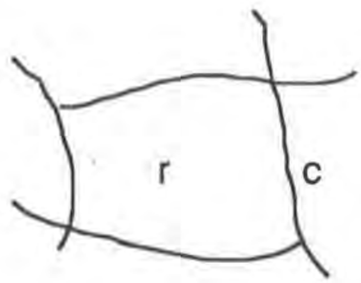
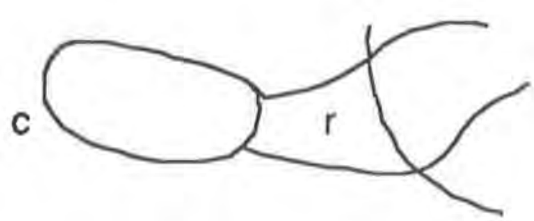
	Relation	Example
(a)	tee(c,c')	
(b)	chi(c,c')	
(c)	bounds(c,r)	
	bounds(c,r)	
(d)	closed(c)	
(e)	interior(c,r)	
(f)	exterior(c,r)	
	exterior(c,r)	

Figure 2. Relations in the image domain

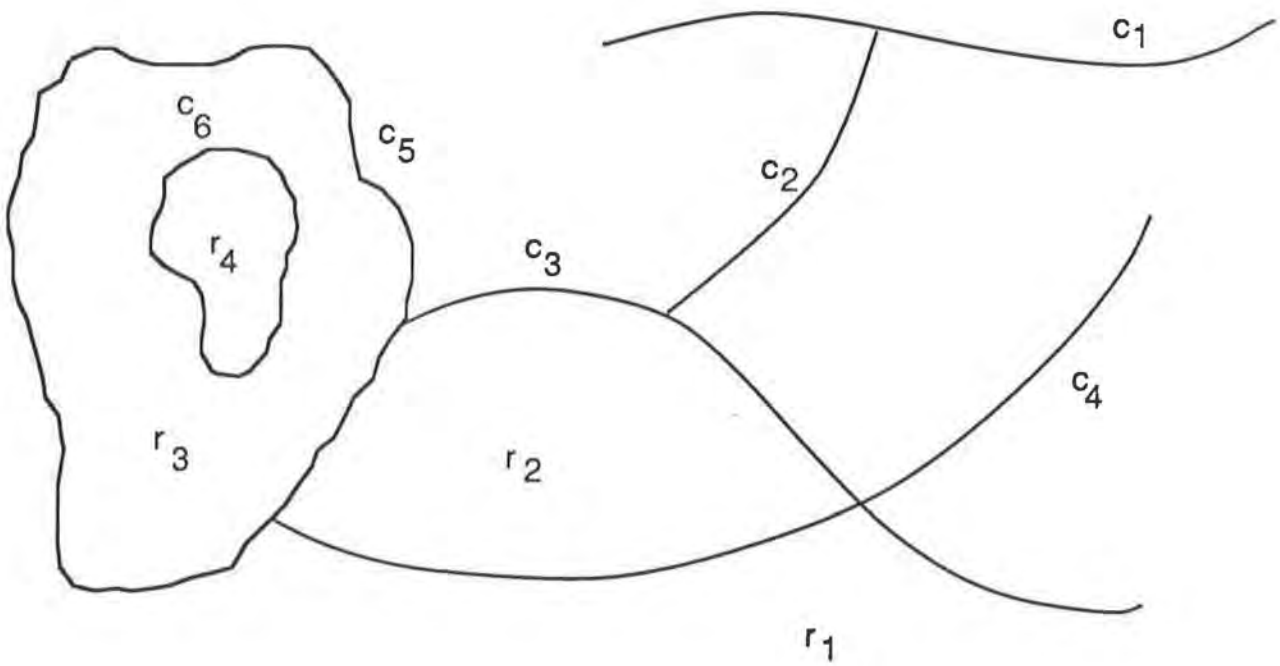


Figure 3. A sketch map

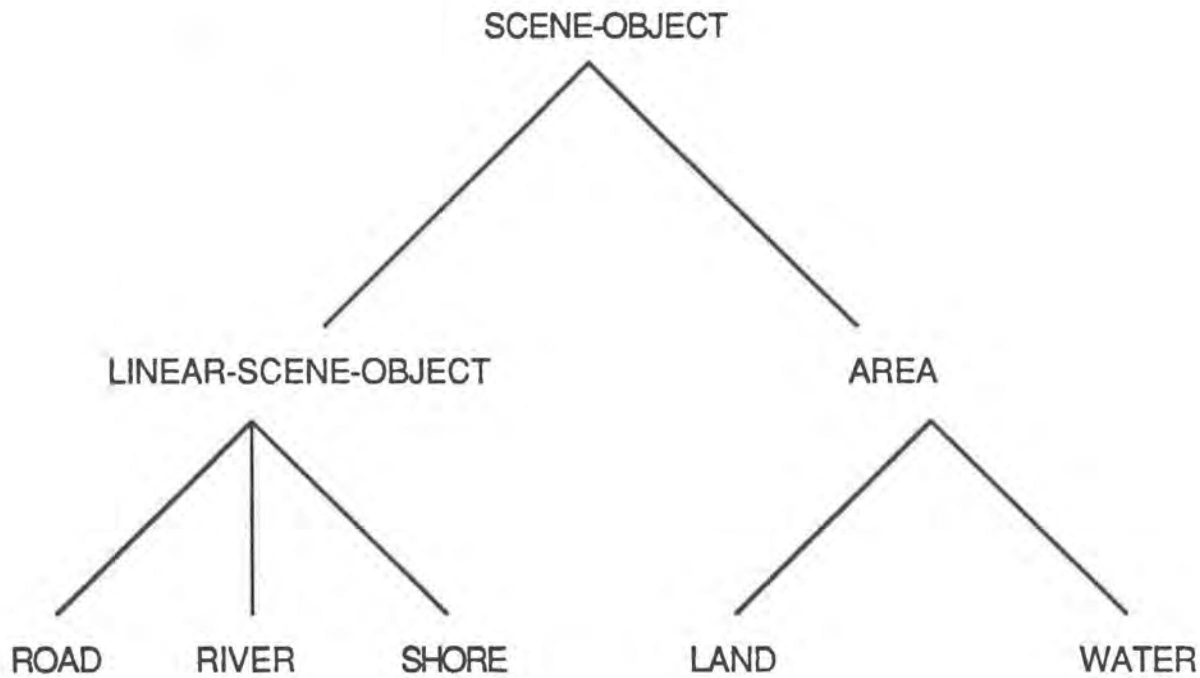


Figure 4. A scene domain taxonomic hierarchy

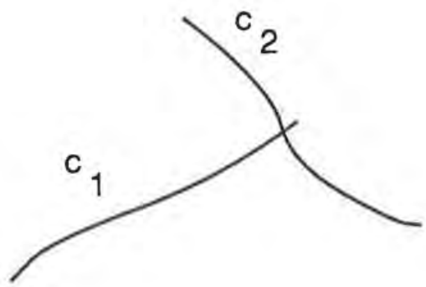


Figure 5. An image with two possible descriptions

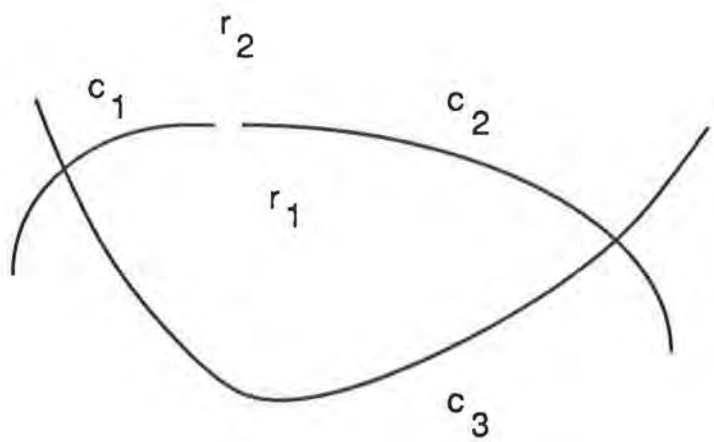


Figure 6. A broken chain?

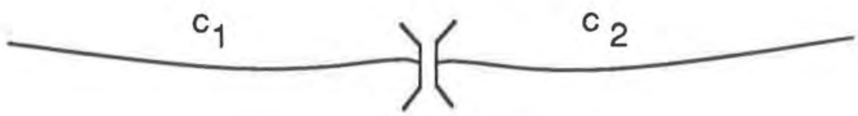


Figure 7. An occluding bridge

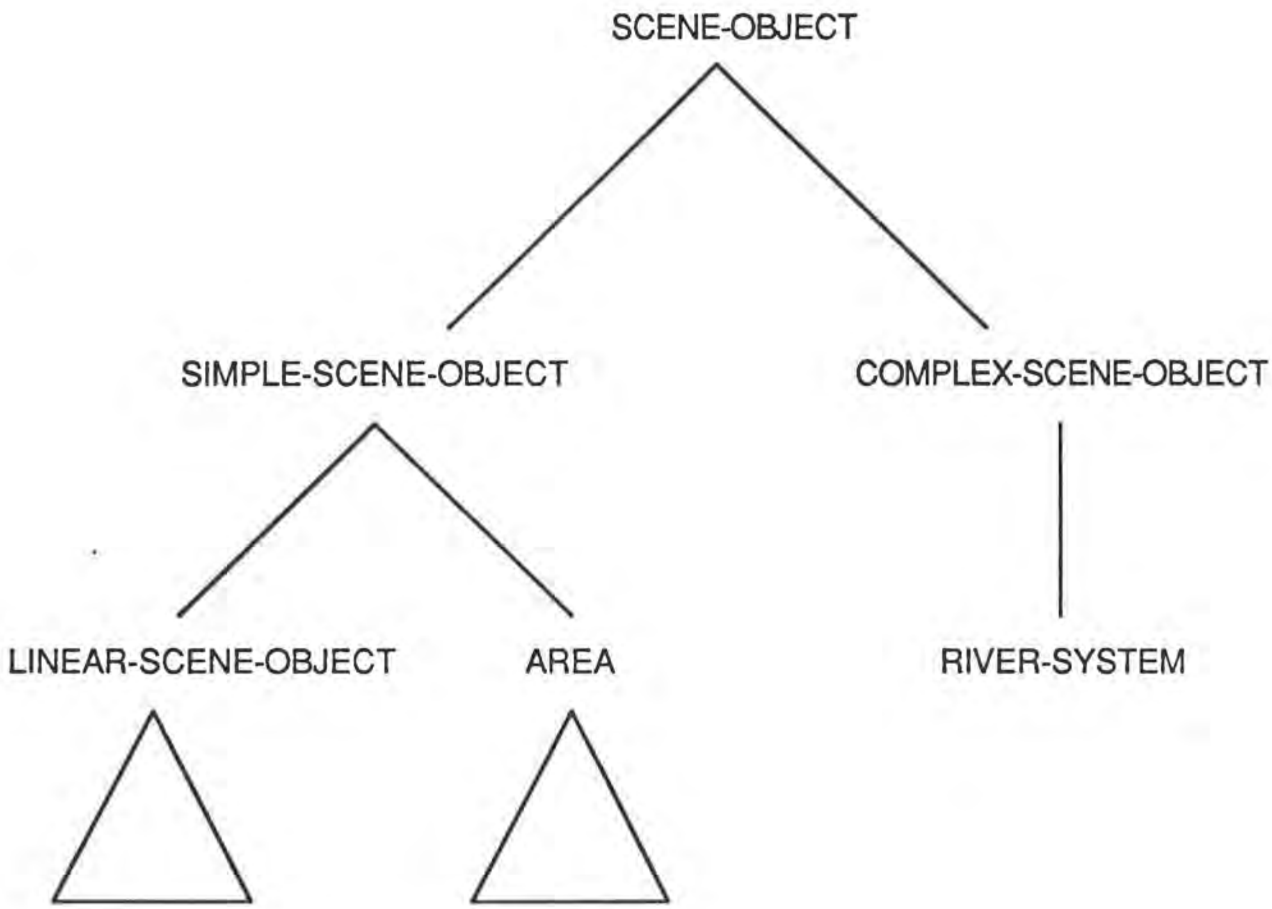


Figure 8. An expanded scene domain taxonomy

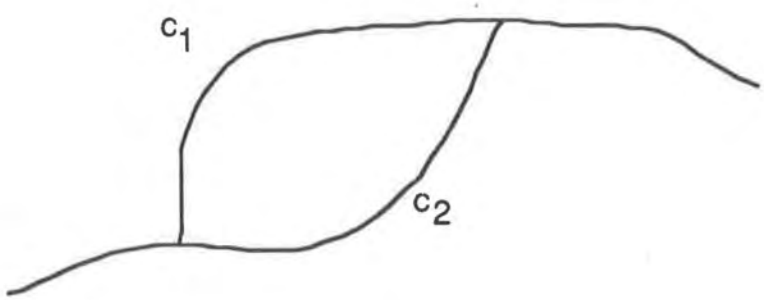


Figure 9. Two rivers?