Knowledge-Based Constraint Satisfaction for Spatial Reasoning

<author info omitted for review process>

Abstract

This system addresses issues in reasoning intelligently over spatial descriptions to produce representations of plausible solutions. Specifically, it looks at coupling a semantic-network-based explicit representation of a natural language description with an ontology of implicit background knowledge. The ontology contains generalized rules for interpreting what objects are and how they should and should not be interpreted alone and in spatial interrelationships. In addition, the linguistic issues of underspecification and uncertainty in spatial semantics and pragmatics are considered.

Introduction

Given a simple English description of a real-world scene, say, *a dog is in front of a house and near a tree*, anyone can easily formulate some kind of corresponding mental image. The description itself contributes only a small part. In fact, most of the details come from a commonsense understanding of the components in the scene and how they can and cannot be laid out in a realistic manner. Performing similar spatial reasoning is the goal of this system, which takes a formal representation of a scene description and produces one or more solutions that specify corresponding locations and orientations for its objects. Such solutions can directly support applications in natural language processing like text understanding, question-and-answer systems, user-friendly animation and graphical rendering tools, etc.

Reasoning over spatial layouts is a difficult task for a computer. As Herskovits (1986) concludes, "[a] computational treatment … will require much greater sophistication than naive representation theory would lead us to expect." What makes the problem especially difficult is that computers lack the vast storehouse of knowledge that people possess and the amazing abilities to reason intelligently over it. This system addresses these problems, as well as the linguistic and knowledge-representation issues of underspecification, or the lack of complete details in any description, and uncertainty, or the wide range of valid interpretations for it.

Despite the potential usefulness of this work, very few related systems exist. CarSim (Dupuy et al. 2001) focuses on graphically rendering the results of vehicle collisions based on accident reports. WordsEye (Coyne and Sproat 2001) leans more toward depicting appropriate poses for actions. Neither appears to address the linguistic and cognitive side of text understanding strongly. In fact, most systems that do spatial layout take a purely geometric approach and do not rely on knowledge at all (Xu, Stewart, and Fiume 2002, Yamada 1993).

Semantic Network

The scene description is defined by a semantic network of concepts, attributes, and relations, which map closely to its nouns, adjectives, and prepositions, respectively. This paper addresses only concepts and relations. In particular, the concepts are limited to concrete (i.e., non-figurative) entities that would typically be found in a zoo. Aside from the obvious visual appeal, plants and animals exhibit a wide variety of important spatial characteristics. The relations are limited to binary constructs such as *X in-front-of Y*. Almost all English spatial prepositions, which have been studied heavily, fall into this category (Freeman 1975, Bennet 1975, Herskovits 1986, Hill 1982, Talmy 1983, and Hawkins 1984).

As in most related systems (except Dupuy et al. 2001), scenes are fabricated by hand rather than input from existing sources. They must also be static, which is a common limitation due to the complexity of dynamic movement, time dependencies, etc. (Adorni, Di Manzo, and Giunchiglia 1984). Finally, only two-dimensional reasoning is supported because most scenes do not actually require the expressive power of true three-dimensional reasoning (Xu, Stewart, and Fiume 2002).

The semantic network serves as a formal representation of the *explicit* knowledge specified in the description. For example, Figure 1 depicts the statement *the dog is to the left of the cat, near the cat, and facing away from the cat*. Each node specifies a *concept instance* that refers to a unique entity in the description. Unique labels distinguish between multiple instances of the same type; e.g., cat_1, cat_2, *Fido*, etc. Each link specifies a binary *relation* that refers to a spatial context formed between its source and target concept instances.

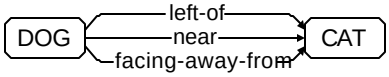

Figure 1: Semantic network

Knowledge Base

The explicit knowledge in the semantic network supplies only the syntactic framework for interpretation. Nothing in it defines the context-independent semantics of what a dog and cat are, for instance, or the context-dependent pragmatics of what it means for one to be in front of the other, etc. For humans, this *implicit* knowledge comes from an acquired understanding of the world. The knowledge base provides some of this background by formally defining how the concept instances and relations are interpreted in context. It addresses the problem of underspecification by augmenting the explicit syntactic knowledge with implicit semantic knowledge. The mechanism is simple: each concept instance in the semantic network has a link to a corresponding *concept definition* in the knowledge base.

The knowledge base is an ontology structured as an inheritance hierarchy of concept definitions that specify a framework for their stereotypical interpretation. Analogous to the way the animal kingdom organizes species based on their shared morphological characteristics and behaviors, the knowledge base organizes its concept definitions by shared spatial characteristics and behaviors. For example, one branch may contain concept definitions that are best approximated spatially as a sphere and support a particular interpretation of the `in-front-of` relation, whereas another branch may do so with a cylinder and a different interpretation.

Concept definitions lower in the hierarchy are inherently more specific than the more general concept definitions above them and inherit their contents. For example, in biological terms, a dog is a canine and inherits its interpretation framework, and likewise for a canine, which is a mammal, and for a mammal, which is an animal, and so on. Multiple inheritance permits a concept definition to derive from more than one branch, although the possibility of conflicts must be considered (e.g., it cannot be approximated as both a sphere and a cylinder). Each concept definition directly specifies or indirectly inherits five components: properties, fields, constraint rules, inference rules, and contexts.

Properties

A property is a straightforward slot-filler construct that assigns a value to a variable. Other components in a concept definition (or in another concept definition linked to it by a relation) can be contingent on this value or even on the presence or absence of the property itself. The most common property is the boolean `has-canonical-front`, which specifies whether a concept definition is interpreted as having a generally accepted front side and therefore is capable of facing something else. This behavior is shared by most animals, for instance but not by plants.

Fields

A field defines the geometry of a two-dimensional plane for space within and around a concept instance (Schirra and Stopp 1993; Gapp 1994). It is projected onto the cells of a polar grid that situates the concept instance in the center as shown in Figure 2. The grid can be oriented such that the arrow always faces outward from the face of a concept instance (if applicable) or always faces north (assumed to be the back of the computer screen). Although various shapes can be projected onto this grid, only wedges and rings have so far proved necessary. Face-oriented wedges define areas like front, back, left, right, etc., which depend on where the concept instance is facing. North-oriented wedges define areas like north, south, east, west, etc. and never rotate. Rings define distances from the center like adjacent, near, far, etc.

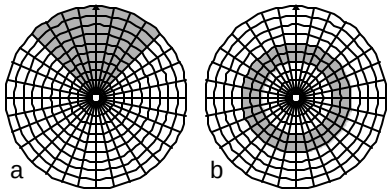

Figure 2: Field geometry (top view)

The geometry of a field is used primarily for *determining* whether the position of another concept instance satisfies a relation between it and the concept instance possessing the field. For example, the relation `in-front-of` is typically bound to the *front* field of a concept instance. Thus, for any concept instance to be inferred as "in front of" this concept instance, it must appear within this field.

Overlaid on the geometry is a probabilistic topography used for *generating* a position of another concept instance such that it satisfies the interpretation of a relation. The most effective topography seems to be a normal distribution through the core of the field as shown in Figure 3, where darker shading indicates higher probability. This mechanism appears to provide for a realistic distribution where central positions are more likely than those at the periphery.

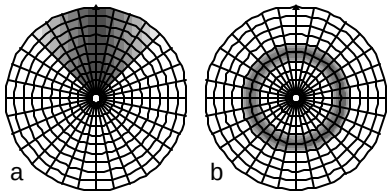

Figure 3: Field probability topography

Constraint Rules

The interpretation of a relation comes from the context formed by the two concept instances it links. A constraint rule defines this interpretation by specifying how each concept instance must interact with respect to their fields. The only rule within the scope of this paper, `FIELD-MUST-CONTAIN`, requires that the target concept instance appear somewhere within a specific field of the source concept instance. For example, the relative distance relation

```
(RELATION near
 (FIELD-MUST-CONTAIN ?b.field-near ?self))
```

applied to *dog near cat* requires that *dog* (bound to `?self`) appear in the *near* field of *cat* (bound to `?b`). The variable

`?self` always binds to the concept instance linked to the concept definition containing the rule. Likewise, `?b` always binds to the other concept instance of the relation.

Constraint rules may also be conditionally evaluated based on the properties of either concept instance. For example,

```
(RELATION facing
 (PROPERTY-IS-TRUE
  ?self.has-canonical-front
  (FIELD-MUST-CONTAIN
   ?self.field-front ?b)))
```

would apply the constraint rule for *dog facing tree*, but not for *tree facing dog* because only *dog* has a true value for property `has-canonical-front`. Thus, a tree, lacking a canonical front, cannot be constrained to face anything.

Relations conditionally dependent on the existence of a canonical front are the most complex. For example, `in-front-of` has two disjoint constraint rules:

```
(RELATION in-front-of
 (PROPERTY-IS-TRUE ?b.has-canonical-front
  (FIELD-MUST-CONTAIN ?b.field-front ?self))
 (PROPERTY-IS-FALSE
  ?self.has-canonical-front
  (FIELD-MUST-CONTAIN
   ?b.field-south ?self)))
```

If the concept instance bound to `?b` is linked to a concept definition that possesses a canonical front, then its front field (e.g., Figure 2a) is evaluated in the constraint rule because it always projects outward relative to the direction it is facing. Otherwise, its south field is evaluated. This mechanism addresses the difference between a local and global frame of reference (Claus et al. 1988). For local, such as *dog in-front-of cat*, the interpretation is solely in terms of the two concept instances. For global, such as *dog in-front-of tree*, it additionally involves the position and orientation of the viewer of the scene (Herskovits 1986). To this effect, the *viewer* is assumed to reside in the south and face the center, which corresponds to looking at the scene on a computer screen. Thus, *dog in-front-of tree* is actually interpreted as *dog between tree and viewer*.

Inference Rules

A constraint rule specifies a template for positioning and orienting a pair of concept instances based on a relation between them. An inference rule is the opposite of this: it determines which relations hold between any pair of concept instances in the solutions generated for the constraint rules. This serves the purpose of inferring implicit relationships that were not stated in the scene description. For example, if a solution to *wolf near tiger* results in *wolf* being north of *tiger* as well, then *wolf* (bound to `?self`) would be located in the *north* field of *tiger* (bound to `?any`), thereby inferring the relationship *wolf north-of tiger*:

```
(IS-IN-FIELD ?self ?any.field-north
 (INFER-RELATIONSHIP north-of ?self ?any))
```

The variable `?any` binds with all concept instances one at a time in the semantic network. If the `IS-IN-FIELD` dependency is satisfied for any pairing of concept instances, then the inferred relationship is bound to them. For example, the evaluation of a semantic network containing the set of concept instances *wolf*, *tiger*, and *tree* would be the set {{*wolf*, *tiger*}, {*wolf*, *tree*}, {*tiger*, *wolf*}, {*tiger*, *tree*}, {*tree*, *wolf*}, {*tree*, *tiger*}}, where the first element of each pair is `?self` and the second is `?any`. As in constraint rules, `?self` always binds to the concept instance linked to the concept definition containing the rule.

Again, conditional evaluation is supported. For example, if *wolf* is oriented such that its front is opposite *tiger*, then *tiger* (bound to `?any`) would be found in the *back* field of *wolf* (bound to `?self`), thereby inferring the relationship *wolf facing-away-from tiger*:

```
(PROPERTY-IS-TRUE ?self.has-canonical-front
 (IS-IN-FIELD ?any ?self.field-back
  (INFER-RELATIONSHIP
   facing-away-from ?self ?any)))
```

The nearly identical inference *tree facing-away-from wolf* would not be made because the concept definition for *tree* specifies that it does not have a canonical front.

Contexts

The conditional evaluation presented so far depends entirely on the properties of the two concept instances in a relationship. This mechanism guides the reasoning by evaluating only applicable rules. It primarily addresses the context-independent semantics of what each instance is and what it can or cannot support. It does not strongly address the context-dependent pragmatics of interaction between instances. This role is played by contexts, which specify for each concept definition how it should be interpreted in a specific relationship with another concept definition. For example, the relationship *X under Y* generally means *X is under the bottom side of Y*. The majority of concept definitions can inherit this default interpretation. However, if *Y* is a tree, for instance, it is more appropriate to override the interpretation to read *X is under the top of Y* (where *top* loosely refers to the canopy). Thus anything under a tree is interpreted as being under its canopy, not under its base. Of course, the original context could be preserved for anything that really belongs there, say worms.

Contexts can be defined tightly between specific concept definitions (e.g., a woodpecker in an oak tree) or loosely between categories of concept definitions (e.g., any kind of bird in any kind of tree). This supports a powerful yet concise generalization capability that cleanly handles both the majority interpretation and various exceptions.

Constraint Propagator

A constraint rule is merely a template that restricts a possible solution. It is the responsibly of the constraint propagator to satisfy all applicable constraint rules simultaneously by

generating positions and orientations for all concept instances in a semantic network. This collection of results, called a solution set, is not unique. In fact, an effectively infinite number of solution sets can be consistent with a scene description. This lack of preciseness in natural language descriptions greatly complicates automated text understanding. Since all solution sets can be considered equally valid, this system uses a probabilistic approach to cull the solution sets to those containing the most likely positions and orientations. The implication (yet to be demonstrated) is that higher-probability solution sets are perceived as generally more acceptable (or less disputable) and can thus be taken as a loosely defined "default" interpretation.

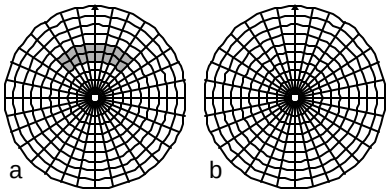

Figure 4: Intersected *front* and *near* fields

In Venn-diagram style, constraint rules for each relation are evaluated as the intersection of their contributing fields. For example, in *dog in-front-of cat and dog near cat*, *dog* must be simultaneously located within both the *front* field and the *near* field of *cat*. The intersection of these respective wedge-shaped and ring-shaped fields in Figure 2 limits the possible positions of dog to the shaded area in Figure 4a. The probability topography of each field in Figure 3 is joined in the intersection as well, thereby making the darkest area in Figure 4b the most likely position for interpretation of the combined relation `in-front-of-and-near`.

The final step in the spatial reasoning is to apply all available inference rules to all pairings of concept instances in the semantic network. If desired, the resulting inferences can then be inserted back into the semantic network, thereby augmenting it with an explicit, commonsense spatial understanding of its contents. The caveat is that different solution sets may produce different, possibly incompatible, inferences. Augmenting identical clones of the semantic network (one per solution set) solves this problem, but it is beyond the scope of discussion.

Results and Discussion

The fuzzy, qualitative nature of spatial relations hinders a formal, quantitative analysis of the performance of this approach in its current stage of development. Nevertheless, preliminary results suggest that it is quite effective. The knowledge base contains over 70 concept definitions that are representative of various animals, plants, and simple structures (e.g., park benches, cages, etc.). The large number of combinations prevents exhaustive testing, but for representative pairings, this approach has been shown to handle both constraints and inferences for the following relations (among others outside the scope of this paper):

- The relative position relations `in-front-of`, `in-back-of`, `left-of`, `right-of`, `in-front-left-of`, `in-front-right-of`, `in-back-left-of`, and `in-back-right-of` in both local and global frames of reference.
- The relative position relations `north-of`, `south-of`, `east-of`, `west-of`, `northwest-of`, `northeast-of`, `southwest-of`, and `southeast-of` for cardinal directions, which are independent of frame of reference.
- The relative distance relations `inside-of`, `adjacent-to`, `near`, `midrange-from`, `far-from`, and `at-the-fringe-of`.
- The relative orientation relations `facing`, and `facing-away-from`.

Evaluation is performed manually to determine whether the results are consistent with a scene description. For scenes with relatively few concept instances, typically less than 10, solution sets to constraints are always generated correctly. For more complex scenes, the most common problem is the failure to find any solution set that satisfies all the constraint rules simultaneously. This reflects a limitation in the constraint propagator, not in the underlying knowledge representation. Near-future modifications to it are expected to improve the results. Finally, regardless of the scene complexity, inferences are always generated correctly.

References

Adorni, G.; Di Manzo, M.; and Giunchiglia, F. 1984. Natural Language driven Image Generation. In Proceedings of COLING-84, 495-500. Stanford, CA.

Bennet, D. 1975. *Spatial and Temporal Uses of English Prepositions. An essay in Stratificational Semantics*. London: Longman.

Claus, B.; Eyferth, K.; Gips, C.; Hörnig, R.; Schmid, U.; Wiebrock, S.; and Wysotzki, F. 1988. Reference Frames for Spatial Inference in Text Understanding. In Freksa, C.; Habel, C.; and Wender K., eds. Spatial Cognition--An interdisciplinary approach to representing and processing spatial knowledge 1404:214--226.

Coyne, B.; and Sproat, R. 2001. WordsEye: An Automatic Text-to-Scene Conversion System. In Proceedings of SIGGRAPH-01, 487-496. Los Angeles, CA.

Dupuy, S.; Egges, A.; Legendre, V.; and Nugues, P. 2001. Generating a 3D Simulation of a Car Accident from a Written Description in Natural Language: the CarSim System. In Proceedings of the Workshop on Temporal and Spatial Information Processing, 1-8. Toulouse, France.

Freeman, J. 1975. The Modeling of Spatial Relations. *Computer Graphics and Image Processing* 4:156-171.

Gapp, K. 1994. Basic Meanings of Spatial Relations: Computation and Evaluation in 3D Space. In Proceedings of AAAI-94, 1393-1398. Seattle, WA.

Hawkins, B. 1984. The Semantics of English Spatial Prepositions. Ph.D. diss., University of California, San Diego.

Herskovits, A. 1986. *Language and Spatial Cognition: An interdisciplinary Study of the Prepositions in English*. Cambridge: Cambridge University Press.

Hill, C. 1982. *Up/down, front/back, left/right. A contrastive study of Hausa and English*. In Weissenborn, J. and Klein, W., eds. *Here and There. Cross-Linguistic Studies of Deixis and Demonstration*. Amsterdam: John Benjamins.

Schirra, J.; and Stopp, E. 1993. ANTLIMA--A Listener Model with Mental Images. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, 175-180. Chambery, France.

Talmy, L. 1983. *How Language Structures Space*. In Pick, H. and Acredolo, L., eds. *Spatial Orientation: Theory, Research, and Application*. New York: Plenum Press.

Xu, K.; Stewart, J.; and Fiume, E. 2002. Constraint-Based Automatic Placement for Scene Composition. In Proceedings of the Conference on Human-Computer Interaction and Computer Graphics, 25-34. Calgary, Canada.

Yamada, A. 1993. Studies on Spatial Description Understanding Based on Geometric Constraints Satisfaction. Ph.D. diss., University of Kyoto.