To appear in:
AI Review Journal, Vol. 10, No. 3-4, 1995

Picture Semantics For Integrating Text And Diagram Input

Raman Rajagopalan
Department of Computer Sciences
University of Texas at Austin
Austin, Texas, 78712
raman@cs.utexas.edu

Abstract.
The saying 'a picture is worth a thousand words' exemplifies the great value of pictures in describing a scenario. Pictures convey spatial information in a compact form, allowing textual descriptions to concentrate on the non-spatial (henceforth, *contextual*) properties of objects. The difficult task in integrating text and diagrammatic input to a system is to establish *coreference* - matching object references in the text to objects in the diagram.
We show that the coreference problem can be greatly simplified if limited contextual information can be provided directly in diagrams. We present a methodology, the Picture Semantics description language, for associating contextual information with objects drawn through graphical editors. Then, we describe our implemented research tool, the Figure Understander, which uses this methodology to integrate the differing information in text and graphically-drawn diagrammatic input into a single unified knowledge base description. We illustrate the utility of our methods through examples from two independent domains.

Key words: Picture Semantics, Coreference, Diagram and Text Integration, Spatial Reasoning

1. Introduction

Maps, schematic diagrams, and pictures accompanying stories are three examples of the many ways in which pictorial information is used in human communication. Larkin and Simon describe the benefits of using both diagrams and text to describe a scene in their classic paper, "Why a Diagram is (Sometimes) Worth 10,000 Words" (Larkin *et al.*, 1987). Pictures can be used to *compactly* describe the spatial properties of objects in a scene. This allows any accompanying textual descriptions to concentrate on the non-spatial domain-specific and problem-specific (henceforth, *contextual*) properties of objects. To clarify our terminology, the rectangular object in the diagram of Figure 1 has the *spatial* property that it is rectangular, and the *contextual* property that it is a magnetic field.

1.1. The coreference problem

Diagrams and text typically describe the same object in different ways. Figure 1 shows the diagram and text statement for a Physics problem (Resnick *et al.*, 1988). The diagram objects are lines, polygons, and circles. The text problem statement uses domain-specific terms, such as 'disk' and 'track', to describe the same objects. Each form of input, diagram and text, contains unique information. In Figure 1,

– **Diagram:**

754 / Faraday's Law of Induction

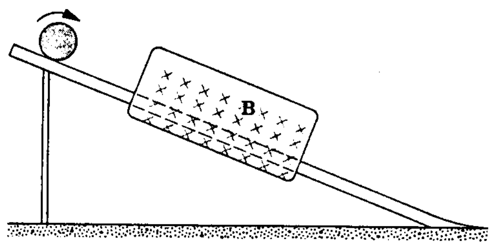

Figure 13 Question 7.

– **Text:** Figure 13 shows an inclined wooden track that passes, for part of its length, through a strong magnetic field. You roll a copper disk down the track. Describe the motion of the disk as it rolls from the top of the track to the bottom.

Fig. 1. A problem from Resnick and Halliday's (1988) textbook, *Fundamentals of Physics*.

the diagram shows the position and shape of the circular object, but it is the word 'disk' in the text that reveals that it is a solid object, and it is the word 'copper' that reveals that it is a conducting object.

The *coreference* problem (Novak *et al.*, 1993) is to associate object references in text with the appropriate diagram objects. Solutions to the coreference problem are important because the unique information in each form of input, diagrams and text, can be integrated only after the mappings between diagram and textual objects are identified.

To solve the coreference problem, human readers can make use of a large "commonsense" knowledge base to quickly visualize the spatial properties of an object mentioned in a text description and identify the appropriate object(s) in a picture. Similar strategies have been adopted in previous work to automate the ability to solve the coreference problem. One approach is to use model libraries which map domain-specific terms to a spatial model for the object (Novak *et al.*, 1993), such as the model that a 'car' is a rectangle with four circles attached to it. Another approach is to use an image database to directly associate a picture with a textual term (Rowe *et al.*, 1993; Srihari, 1994).

Both approaches face problems with the completeness of the model library or image database and ambiguities in assessing the quality of a match against a given input. Unfortunately, a given object or collection of objects in a diagram may map to several models in a model library, or objects in the input may have a different shape or orientation, and thus appear to be different, than objects in an image database.

For example, in interpreting the diagram in Figure 1, a human reader can easily recognize that the sloping portion of the triangular base forms the 'track', that the vertical piece forms the support for the track, and that the horizontal line is the ground. In a model library, a possible mapping for the word 'track' is that it is a line object in a diagram.

In the text of Figure 1, there is only one reference to a 'line' object - the track. The diagram contains three line objects - the track, the support, and the ground. In this case, we would either need a model for a 'supported track' in the library which would include all three line objects in the diagram, or the underlying system would have to perform complex reasoning about the information in the text to associate only the sloping line with the word 'track'.

1.2. Overview of our goals and solution

In previous work, researchers have addressed the problem of automatically interpreting text and pictorial input *intended for human use*, such as Srihari's interests in identifying persons in newspaper photographs based on clues in an associated text caption (Srihari, 1994), or Novak and Bulko's (1993) interest in solving textbook Physics problems using the exact text problem statement and associated diagrams that are available to the human reader. These problem assumptions do not allow any automation-friendly modifications in either the diagram or text input, thus preserving the full complexity of the coreference problem, and making the solutions difficult to implement and use.

For example, in previous work, it is assumed that the pictorial input is provided in visual (raster) form, such as a camera image. In such cases, it becomes necessary to perform visual scene interpretation (Ballard *et al.*, 1982) simply to identify the pictorial objects of interest. There can be many ambiguities. In Figure 1, for example, it may be non-trivial to infer that the X's form a pattern to be associated with the rectangular object, or to infer that the label 'B' is attached to the rectangular object, as opposed to the track or to the diagram as a whole.

1.2.1. *Our goal*

Our goal is to provide a simpler methodology which allows the combined use of text and diagrams to compactly provide the spatial and contextual properties of a scene as input to a given system (Rajagopalan *et al.*, 1994; Rajagopalan, 1994b). This need was forced on us as we pursued another goal, the development of a problem solver for the magnetic fields domain which performs qualitative reasoning about changes in both spatial and non-spatial properties to solve problems (Rajagopalan, 1994a). As we considered how to input a description of the initial state, we found that special purpose text forms were too cumbersome for providing a complete description of the spatial properties, and that diagrams were a far more natural method of providing the same information. However, we still required text input to describe certain non-spatial properties of the initial state, including changes that may be taking place. Thus, we encountered the coreference problem.

1.2.2. *Our approach*

We have developed a novel solution to the coreference problem based on two observations: (1) Annotations are often used in drawings to allow human readers to associate non-spatial contextual information with objects. For example, Figure 1 contains two means to identify the rectangular object as a magnetic field: the label, B, and the pattern of Xs; (2) Graphical drawing tools provide several methods, such as colors, filling patterns, and brush types, to highlight objects.

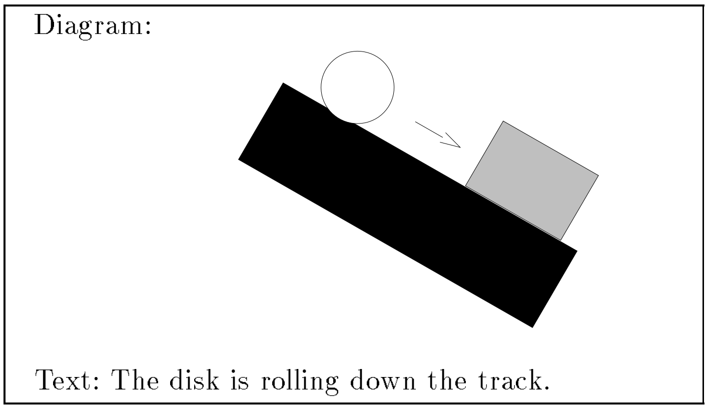

Fig. 2. Input given to the Figure Understander for the textbook problem shown in Figure 1.

Our solution to the problem of compactly providing both spatial and non-spatial information as input to a given reasoning system has been to automate the ability to associate contextual information with the use of highlighting colors, patterns, and brush types for objects in diagrams drawn through graphical drawing tools. We have a developed a special description language, the *Picture Semantics*, for this purpose.

1.2.3. *Advantages of our approach*

Our approach has two favorable consequences for the coreference problem: (1) The need for additional text input can sometimes be completely eliminated, thus trivially solving the coreference problem. (2) When the need for additional text input persists, our approach allows constraints on both the spatial and contextual properties of objects to be used when matching textual and diagram objects.

Our solution is straight-forward to implement and use. The assumption that the pictorial input is given through a graphical editor greatly reduces the complexity of diagram interpretation - anything drawn as a distinct object in the diagram can be readily identified as a distinct object from the output of the graphical editor. Further, properties such as brush types, filling patterns, and colors can be directly associated with objects, thus eliminating a source of ambiguity present in interpreting visual input in raster form.

The remainder of this paper is organized as follows. In section 2, we describe the Picture Semantics language for describing contextual properties directly within a diagram, and illustrate its use in solving the coreference problem for the magnetic fields problem described in Figure 1. In Section 3, we describe the Figure Understander, our implemented tool for automatically interpreting and integrating the information in text and diagram input. In Section 4, we illustrate the utility of our approach through additional examples involving the interpretation of dialogue about an urban scene.

2. The Picture Semantics description

The diagram and text input to the Figure Understander for the problem of Figure 1 is given in Figure 2. The cruder diagram and greatly simplified text form

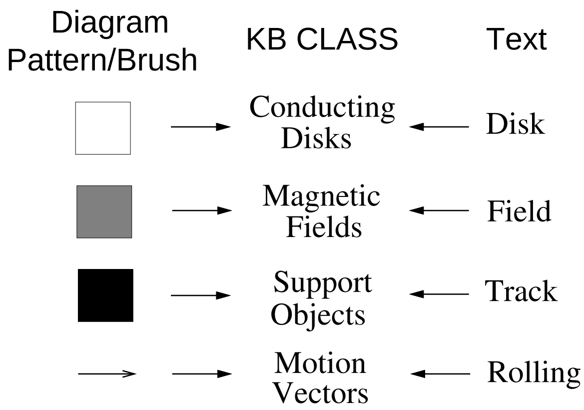

Fig. 3. Diagram and text semantics for Figure 2.

are sufficient to capture all the necessary input information for the problem. This simplification is possible because of the power of our Picture Semantics description, illustrated in Figures 3 and 4, which allows contextual properties to be associated with the use of brush types, filling patterns, and colors in diagrams.

Incidentally, text labels could also be used in the Picture Semantics if the graphical editor provides a means to directly attach the label to other diagram objects. Otherwise, the task of associating the text label with a diagram object can itself be a challenging problem (He *et al.*, 1994) particularly when the diagram contains overlays. Since the graphical editor we use, the `InterViews` drawing editor, treats text as a separate object, we shall not consider text labels any further in this paper.

In our approach, we first process the diagram before processing the text input. The result is a partial knowledge base with instantiations for each diagram object under the appropriate knowledge base class. For the diagram in Figure 2, we use the Picture Semantics form to provide the following associations between diagram objects and knowledge base (KB) classes, as illustrated in Figure 3: white objects are conducting disks, gray objects are magnetic fields pointed into the page, black objects are supports, and vectors with the light brush type used in the diagram are motion vectors.

In our application, we always provide a vector in the diagram to provide the direction of motion of objects, intrinsic front directions, and the directions of any forces acting on the objects. In the example of Figure 2, the diagram does not specify which object is in motion, since the vector is floating in free space. This information is obtained from the text input, "the disk is rolling down the track".

2.1. Solving the coreference problem in Figure 2

After processing the text input in Figure 2, we learn that there must be a mobile, conducting disk in the diagram which is in motion (from 'disk is rolling'). Further, we learn that there is contact between the disk and a supporting track (from 'down the track'). From our application-wide assumption described above, we also infer that there must be a motion vector present.

The coreference problem in processing the diagram and text in Figure 2 is to identify the conducting disk, motion vector, and supporting track in the diagram, even though the diagram primitives are only lines, circles, and polygons.

The problem is quickly solved by use of the Picture Semantics description, since, when processing the diagram, we have already extracted the information that there are instances of conducting disks, magnetic fields, supporting objects, and motion vectors in the diagram. Thus, once the text processing identifies the need for finding a conducting disk, motion vector, and supporting track, we can directly search against the knowledge base for diagram objects with these contextual properties.

Since there is only one object of each KB class in the diagram, the matching process becomes straightforward, and we can assert the direction of motion specified by the vector into the KB description of the disk. In more complex scenes, when there are multiple instances of a given object type in the diagram, spatial constraints in the text, such as the clue that the conducting disk and supporting track must be in contact, can be used to isolate the specific instances of interest. In Section 4, we present an urban scene interpretation example where there are multiple instances of each object type.

2.2. Features of the Picture Semantics description

Figure 4 shows a partial Picture Semantics description for our magnetic fields problems. There are four main blocks. Under `:patterns-for-2d-objects` and `:brush-types-for-vector-objects`, the user can assign a meaningful label, such as the label 'mobile-conducting-disk' assigned to the solid-white pattern, for each of the color, filling pattern, and vector types allowed by the graphical editor used to draw the diagram.

The second block (`:object-groups`) allows the user to organize individual labels into groups and to specify the root to be used for the names of the knowledge base instantiations of diagram objects of this group. For example, the group 'magnetic-fields' is defined to include diagram objects with a solid-gray pattern and those with a horizontal line pattern. The first diagram object encountered with this pattern will have a knowledge base instantiation with the name 'F'. The second will be named F-1, the third F-2, and so on.

The third block specifies the knowledge base assertions to be made for each individual object type (`:individual-object-semantics`) and for groups as a whole (`:object-group-semantics`). All contextual information is expressed in terms of predicates in the Algernon knowledge representation language (Crawford *et al.*, 1991).

For example, under `:object-group-semantics`, we have specified that all diagram objects of the group 'conducting-disks' are members of the knowledge base class 'conducting-disks', and that they have the property of being solid objects. Under `:individual-object-semantics`, we specify that diagram objects with a solid-white pattern (label 'mobile-conducting-disks') have the additional property that they are mobile.

The fourth block (`:special-procedures`) specifies user-defined, special-purpose procedures to be run after diagram processing is completed. For example, the special procedure 'connect-vectors-to-objects' associates a vector in the diagram with the object which contains the tail of the vector whenever an unambiguous match can be found. The vector in Figure 2 is intentionally not attached to the

```
(defpict pictures-for-mag-fields
    :patterns-for-2d-objects
      ; Graphical Editor Pattern      ; Label For Object
      ((solid-gray                    field-into-page)
       (horizontal-line               horizontal-field)
       (solid-black                   track)
       (solid-white                   mobile-conducting-disk)
       (light-cross-mesh              immobile-conducting-disk))

    :brush-types-for-vector-objects
      ; Graphical Editor Brush Type   ; Label For Object
      ((vl-s-vector-out               mot-dir-vect))

    :object-groups
      ; Label For Group  ; Root Name For KB objects  ; Objects In Group
      ((magnetic-fields        F     (field-into-page horizontal-field))
       (conducting-disks       D     (mobile-conducting-disk
                                      immobile-conducting-disk))
       (support-objects        S     (track))
       (motion-vectors         MV    (mot-dir-vect)))

    :object-group-semantics
      ((conducting-disks ; Group Label
       ; Properties of group to be asserted into the KB.
        ((member conducting-disks ?obj) ; membership in KB class.
         (is-solid ?obj TRUE)))
       (magnetic-fields
         ((member steady-uniform-magnetic-fields ?obj)
          (source-of-field ?obj permanent)))
       (motion-vectors
         ((member motion-vectors ?obj))))

    :individual-object-semantics
      ((conducting-disk  ; Group Label
        (mobile-conducting-disk  ; Object Label
         ; special properties of object type to assert into the KB.
         ((mobility-of ?obj mobile))))
       (magnetic-fields
        (field-into-page
         ((field-direction-of ?obj '(0 0 -1))))))

    :special-procedures (connect-vectors-to-objects))
```

Fig. 4. A partial Picture Semantics description for our magnetic fields problems.

conducting disk. However, we could have eliminated the need for text input completely by moving the tail of the vector into the disk, allowing the fact that the disk is moving to be extracted directly from the diagram!

3. The Figure Understander

The Figure Understander is an implemented research tool for integrating the information in text and *graphically drawn* diagrammatic input into a single knowledge base description. The intended use of the Figure Understander is to support applications, such as map understanding systems, story understanding systems, or automated problem solving systems, that will benefit from the ability to *compactly* describe the spatial state of the world through a diagram, and yet may require additional text-based input to describe the domain properties of the diagram objects and any dynamic changes that are taking place.

The input to the Figure Understander includes a diagram produced through an object-oriented drawing editor (currently, the `Postscript` file for a diagram produced using the `Interviews` drawing editor), a Picture Semantics description for the diagram, one or more sentences of descriptive text, and an Algernon (Crawford *et al.*, 1991) knowledge base. It outputs an integrated description of the information in each form of input, diagram and text, into the knowledge base.

3.1. Diagram processing

The strength of the Figure Understander lies in its diagram interpretation capabilities. As the diagram is processed, a knowledge base (KB) object is created for each diagram object. Both the contextual information given in the Picture Semantics and the qualitative spatial properties of interest are attached to the KB description of the object. The spatial properties extracted include the spatial extent (size, shape, and position), connectivity, and orientation of individual objects, and topological and directional spatial relations between objects. For example, the connectivity of a polygon is stored by creating knowledge base structures for all the edges and vectors, and storing the connections between the edges and vertices. Incidentally, the Figure Understander also contains heuristics to allow for human errors in drawing the diagrams, such as adjusting for an unintentional gap of a few pixels between objects intended to be touching each other.

3.1.1. *The spatial model*

The Figure Understander extracts and reasons with a *qualitative* internal description of the initial spatial state. Our spatial representation, described in greater detail in (Rajagopalan, 1994a), has been used to solve problems involving static scenes as well as problems involving translational and rotational motion. For the purposes of this paper, we will assume that all objects are 2-dimensional and are confined to the XY plane.

Spatial relations between objects are evaluated using 'ideal meanings', as defined by Herskovits (1985). More specifically, for problems involving static scenes or only translational motion, we use a minimum bounding rectangle, similar to Chang, et al., (1987) and Jungert (1992), to approximate the spatial extent of each object. Two objects are assumed to satisfy a spatial relation if their bounding rectangles satisfy the relation.

The minimal bounding rectangle is drawn around the extremal edges or vertices of an object: its rightmost (rm), leftmost (lm), topmost (tm), and bottommost (bm) vertices or edges, as shown in Figure 5. Then, as shown in Figure 6, we define eight regions around the bounding box of each object. The relative positions of two objects are determined based on the regions around the bounding box of a reference object that are intersected by bounding box of the second object.

We model both directional (e.g., right-of, left-of) and topological (e.g., next-to, adjacent, intersecting) spatial relations between objects in terms of *qualitative inequality constraints* on the X-coordinate values of the rightmost and leftmost

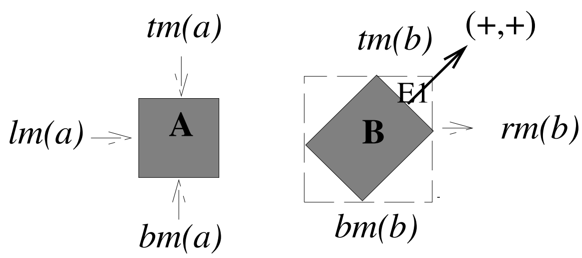

Fig. 5. Modeling spatial extent and orientation using extremal points.

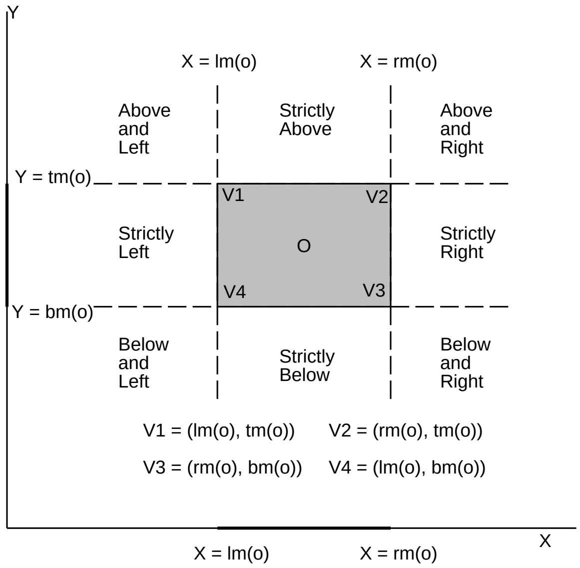

Fig. 6. Regions defined by a rectangular bounding box with vertices V1, V2, V3, and V4.

extremal edges of their bounding boxes, and the Y-coordinate values of the topmost and bottommost extremal edges of the bounding boxes. The inequality relations that define the relative positions of objects in an input diagram are extracted automatically by the Figure Understander.

The list of binary spatial relations recognized by the Figure Understander is given in Figure 7. While space limitations do not permit a formal definition to

`Right-of`	`Strictly-right-of`	`Touching`	`Adjacent`
`Left-of`	`Strictly-left-of`	`Touching-right`	`Adjacent-right`
`Above`	`Strictly-above`	`Touching-left`	`Adjacent-left`
`Below`	`Strictly-below`	`Touching-above`	`Adjacent-above`
`Ahead-of`	`Strictly-ahead-of`	`Touching-below`	`Adjacent-below`
`Behind`	`Strictly-behind`	`Disconnected`	`Overlapped`
`Intersecting`	`Intersecting-right`	`Intersecting-left`	`Intersecting-top`
`Equals`	`Encloses`	`Enclosed-in`	`Intersecting-bottom`

Fig. 7. The spatial relations recognized by the Figure Understander.

.

be presented for all of these spatial relations, here are three examples. For two bounding boxes, A and B:

- Left-of(A,B) iff $rm(A) < lm(B)$, where $rm(A)$ is the X-coordinate value of the rightmost edge of A, and $lm(B)$ is the X-coordinate value of the leftmost edge of B.

- Aligned-X(A,B) iff $tm(A) > bm(B) \wedge bm(A) < tm(B)$, where $tm(A)$ and $tm(B)$ are the Y-coordinate values of the topmost edges of A and B, and $bm(A)$ and $bm(B)$ are the Y-coordinate values of the bottommost edges of A and B. The region defined by the Aligned-X relation contains the points where motion in the X direction could cause an intersection between A and B.

- Strictly-left-of(A,B) iff $Left\text{-}of(A, B) \wedge Aligned\text{-}X(A, B)$.

We model the orientation of an object in two ways: (1) the surface normal direction of each distinct surface (edge) of the object, as used by Nielsen (1988), and (2) the angle between the plane of the object and each of the planes formed by the coordinate axes. The latter is required only for reasoning about rotation and will not be discussed further here.

For (1), we can obtain the surface normal direction of any edge or arc of a convex object from knowledge of the *identities* of the vertices and edges which form the extremal points/surfaces of the object. For example, if an edge or arc lies between the topmost and rightmost extremal points/surfaces, such as edge E1 in Figure 5, then its surface normal direction will have positive X and Y components. We similarly model all 8 qualitative orientations in 2-dimensional space as presented by Freksa (1992) and Mukerjee and Joe (1990).

The distinction in our work is that while the 'idealized' minimum bounding rectangles are used as the basis for describing the relative positions of objects, for reasoning about orientation, we maintain knowledge of the actual orientation of the surfaces of an underlying object. This is necessary to model such properties as the direction of current flow along a particular surface of an object, the directions of magnetic forces acting on an object, and the intrinsic front direction of an object.

Further details of the advantages and limitations of our spatial model, its extensions for reasoning about translational and rotational motion, and details of its implementation within the framework of existing qualitative modeling and simulation systems (Kuipers, 1994) may be found in our AAAI-94 article (Rajagopalan, 1994a).

3.2. Text processing

The task of attaching semantic information given in any accompanying text sentences to diagram objects requires one to first locate the appropriate diagram objects to which the information is to be attached. We require that the text must provide sufficient information to *uniquely* locate the diagram objects to which the information is to be attached.

The Figure Understander performs text processing after processing the diagram. Since the Picture Semantics provides the conceptual class for each diagram

```
(defparameter *semantic-functions*
   '((disk (lambda ()
            (let ((?v1 (make-variable))
                 `(:the ,?v1 (member conducting-disks ,?v1)))))

     (track (lambda ()
             (let ((?v1 (make-variable))
                  `(:the ,?v1  (member support-objects ,?v1))))) ))
```

Fig. 8. Examples of semantic interpretation functions used to obtain the conceptual class for textual objects.

```
Interpreting text sentence:
 (THE DISK IS ROLLING DOWN THE TRACK)

 The following Algernon path shall be asserted:

((:THE ?V1 (MEMBER CONDUCTING-DISKS ?V1) (MOBILITY-OF ?V1 MOBILE))
 (:THE ?V2 (MEMBER SUPPORT-OBJECTS ?V2) (TOUCHING ?V1 ?V2))
 (:THE ?V3 (MEMBER MOTION-VECTORS ?V3) (VECTOR-DIRECTION-OF ?V3 ?VECT))
 (VELOCITY-VECTOR ?V1 ?VECT))
```

Fig. 9. Algernon query and assert form generated for the text in Figure 2.

object, if we can obtain the conceptual class of each text object, text and diagram objects could be matched based on their conceptual class. We use domain-dependent semantic interpretation functions, as illustrated in Figure 8, to provide the conceptual class for any nouns or pronouns that may be used in the text as reference objects. In Figure 8, the function `make-variable` is used to return a variable name, such as ?V1 or ?V2, to be substituted into the `:the` from.

Algernon's (Crawford *et al.*, 1991) definite description facility `(:the)` is used to connect object references in text to objects in the knowledge base (KB). The form `(:the ?v1 (member conducting-disks ?v1))` asks Algernon to find a *unique* object, to be bound to the variable ?v1, that is a member of the knowledge base class `conducting-disks`. If it is the case that there is more than one conducting disk in the knowledge base, we can use any additional constraints, such as spatial relations, obtained from processing the text to narrow the search of the knowledge base.

The text processing is accomplished by first parsing a sentence through an ATN grammar, and then interpreting the parse output through a domain-dependent semantic interpreter. For each supported sentence type, the semantic interpreter examines the subject, verb, direct object, etc., and assembles a query-and-assert form in the syntax of the Algernon language for locating the appropriate diagram objects in the knowledge base (query) and attaching any new information given in the text about the object (assert).

Figure 9 shows the Algernon form generated after processing the sentence 'The disk is rolling down the track'. There are three queries (:the) and one new assertion. The queries search for diagram objects using both spatial properties and con-

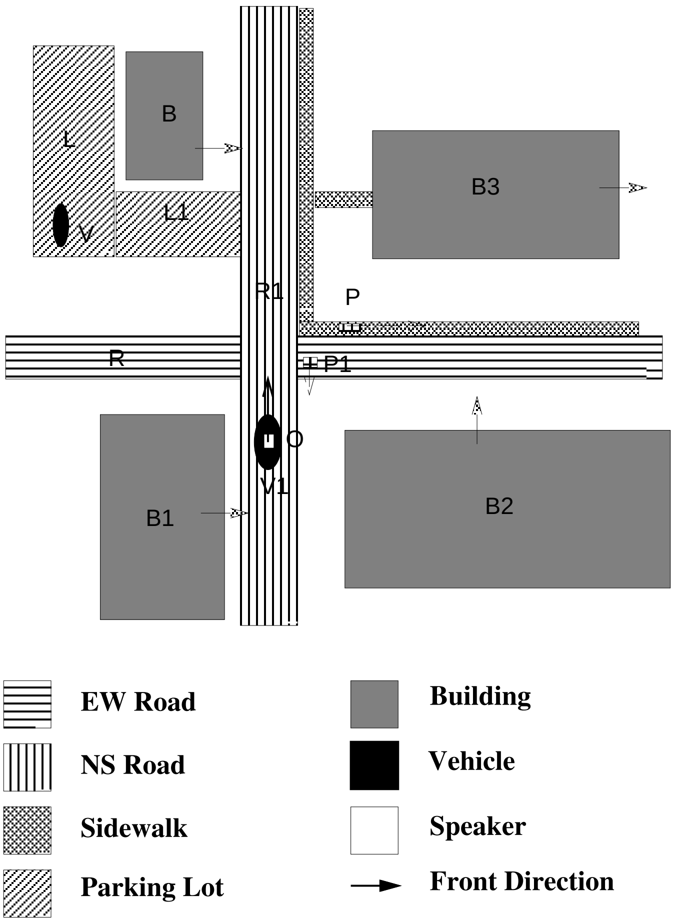

Fig. 10. An urban scene - buildings, streets, vehicles, and pedestrians. The labels identify the internal names of the knowledge base objects created for each diagram object. The key shows the domain class for each diagram object.

textual properties. The first form, `(:the ?v1 (member conducting-disks ?v1) (mobility-of ?v1 mobile))`, is a query to locate a *unique* diagram object which is a member of the knowledge base (KB) class 'conducting-disks', and which is also a mobile object. The second query is to locate a member of the KB class 'support-objects' such that the conducting disk and support object are touching each other. The third query searches for a motion vector and retrieves its direction. The assert then assigns the direction of the motion vector as the velocity vector of the rolling disk.

4. A scene interpretation example

In Section 2, we discussed the use of the Picture Semantics description to provide contextual information, such as the knowledge base (KB) class for each diagram object, directly within the diagram. We showed that when each diagram object is of a unique KB class, the solution to the coreference problem of matching text and diagram objects becomes straightforward. In this section, we consider solution strategies for the coreference problem when there are multiple objects of the same

```
(defpict roads-and-buildings
    :patterns-for-2d-objects
      ;Graphical Pattern  Domain Object Type
      ((solid-white          speaker)
       (solid-black          vehicle)
       (solid-gray           building)
       (slash                parking-lot)
       (horizontal-line      NS-road)
       (vertical-line        EW-road)
       (wide-rect-mesh       pedestrian)
       (light-cross-mesh     sidewalk))

    :brush-types-for-vector-objects
      ;Graphical Brush Type, Domain Object Type
      ((vl-s-vector-out      int-front-vect))

    :object-groups
      ; Domain-Class-Name    Abb-Name   Member-Object-Type
      ((sidewalks               S         (sidewalk))
       (speakers                O         (speaker))
       (pedestrians             P         (pedestrian))
       (roads                   R         (EW-road NS-road))
       (buildings               B         (building))
       (int-front-vectors       IF-V      (int-front-vect)))

    :object-group-semantics
      ; Semantics given as Algernon predicates.
      ((buildings    ((member buildings ?obj)))
       (roads        ((member roads ?obj)))
       (pedestrians  ((member people ?obj)))
       (speakers     ((member speakers ?obj))))

    :individual-object-semantics
    ; format: (class-name (obj-type semantics) (obj-type semantics) ..)
      ((roads
        (EW-road ((direction-of ?obj EW)))
        (NS-road ((direction-of ?obj NS)))))

    :special-procedures ((connect-vectors-to-objects))  )
```

Fig. 11. Example of a partial Picture Semantics for urban scene interpretation.

contextual class in the diagram. Essentially, in such cases, the Figure Understander expects additional constraining information in the text to uniquely identify the appropriate diagram object. One useful source of additional constraints is the use of spatial relations in the text.

Figure 10 shows an input diagram for an urban scene that consists of many lines, rectangles, and ellipses. To assist the reader, Figure 10 includes a key which visually presents the Picture Semantics for the diagram. The Picture Semantics form shown in Figure 11 provides the contextual knowledge that the objects are buildings, streets, vehicles, and pedestrians. The goal for this example is to process the text sentences in Figure 12 to attach additional contextual properties to descriptions of the diagram objects.

1. My name is Raman.
2. The name of the building that is to my right is Welch Hall.
3. The name of the building that is in front of Welch Hall is Taylor Hall.
4. The name of the building that is behind Taylor Hall is the Aerospace-Engineering building.
5. The name of the building that is to the right of the Aerospace-Engineering building is the Biology building.
6. I am traveling on Twenty-Fourth street.
7. Speedway street intersects Twenty-Fourth street.
8. Ben is crossing Speedway street.

Fig. 12. A collection of descriptive sentences for the scene depicted in Figure 10.

The text sentences in Figure 12 make heavy use of spatial relations to assist the reader in locating the diagram objects of interest. By convention, the text interpretor for the urban scene understanding application interprets references to the directional spatial relations right of, left of, ahead of, and behind as strictly right of, strictly left of, strictly above, and strictly behind. For example, to interpret sentence 2, "The name of the building that is to *my right* is Welch Hall", the Figure Understander uses the fact that the word 'my' refers to the speaker, locates the only speaker (O) in the diagram, and then uses the spatial relation 'building *strictly to the right* of the speaker' to locate the only building (B2) that satisfies the 'strictly right of' relation with respect to the speaker and labels that as 'Welch Hall'.

4.1. Intrinsic/Deictic frames of reference

Retz-Schmidt (1988) discusses that in human use of spatial prepositions, extrinsic, intrinsic, or deictic frames of reference could be used. The Figure Understander reasons about spatial relations with respect to the extrinsic frame of reference of the diagram. However, to allow reasoning about deictic and intrinsic frames of reference, the Figure Understander's spatial reasoning module allows queries about directional spatial relations to carry a frame of reference argument, and automatically determines an equivalent relation to prove in terms of the extrinsic frame of reference of the diagram. In Figure 10, the vectors define the front direction for each object. Currently, the Figure Understander only considers intrinsic and deictic frames of reference that are orthogonal to that of the diagram.

In general, when processing text sentences, it is possible that there can be ambiguity concerning whether spatial relations are specified with respect to an intrinsic or deictic frame of reference. To avoid this ambiguity, the Figure Understander follows the convention that only objects mentioned explicitly in text sentences can serve as the frame of reference. Thus, in processing sentence 2, the deictic frame of reference of the speaker is used as the frame of reference, based on the use of the phrase 'my right'. However, in processing sentences 3-5, the intrinsic frames of reference of Welch Hall, Taylor Hall, and the Aerospace-Engineering building, respectively, are used instead, since those sentences contain such phrases as 'in front of Welch Hall' and 'behind Taylor Hall'.

4.2. Annotated demonstration

In the remainder of this section, we shall demonstrate the combined use of contextual and spatial constraints in text to solve the coreference problem in processing the diagram in Figure 10 and the text in Figure 12. All computer generated output is shown in typewriter font.

```
> (parse-figure "speedway-24th.ps"       ; input diagram
'((my name is Raman))    ; input text
'roads-and-buildings)    ; picture semantics form

************************************************
***             Parsing The Figure           ***
************************************************

Extracting spatial information from the figure.
Executing user defined special procedures.
Interpreting text sentence:
 (MY NAME IS RAMAN)

 The following Algernon path shall be asserted:
((:THE ?V1 (MEMBER SPEAKERS ?V1))
 (NAME-OF ?V1 'RAMAN))
```

The Figure Understander is invoked through the `parse-figure` function. The diagram is provided in the form of a PostScript file produced by the InterViews drawing editor. The input also includes a list of the sentences to be processed and the picture semantics form to be used. As the Figure Understander processes the diagram, it creates knowledge base objects for all the diagram objects, and stores both the spatial information about the objects and the semantic information given through the picture semantics.

After processing the diagram and the text sentence 'My name is Raman', database entries will exist for all the objects in the diagram, and their internal names will correspond to the abbreviations given in the picture semantics. The internal names for the larger objects in the scene are given as labels in Figure 10.

Let us now query the entire set of two dimensional objects in the knowledge base to identify the objects with bindings for the name-of relation. This is accomplished through Algernon's `ask` function. A list of bindings for the variables (symbols with the initial character, `?`) used within the `ask` function is retrieved using the `:collect` argument.

```
algy> (ask '((member convex-2-d-objects ?x)
             (name-of ?x ?y)) :collect '(?x ?y))
  ((O RAMAN))
```

Note that the only object with a domain-specific name is the speaker, O, whose name is Raman. Next, we shall proceed to process sentences 2-5 from Figure 12. Below, we show the Algernon query-and-assert form generated for the sentence 'The name of the building that is behind Taylor Hall is the Aerospace-Engineering building.' The query is to find two objects: first, to find a convex 2-dimensional object, `?V1`, whose name is `'(Taylor Hall)`, and second, to find a building `?V2` that is strictly behind the object `?V1`. The query further specifies that the 'strictly behind' relation is to be interpreted in terms of the intrinsic front direction of object ?V1 if it has one. If object ?V1 does not have an intrinsic front direction, then the query will be interpreted in terms of the extrinsic frame of reference of the diagram. If such a building ?V2 is found, the assert says to name the building as the Aerospace Engineering building.

In processing this query and assert form, since the intrinsic front direction of Taylor Hall points to the right, as given by the vector attached to object B3 in Figure 10, the relation 'behind' is determined with respect to this new frame of reference to identify the building labeled B in Figure 10 as the Aerospace Engineering building.

```
algy> (interpret '(The name of the building that is behind Taylor Hall
                   is the Aerospace-Engineering building))

 The following Algernon path shall be asserted:

((:THE ?V1 (MEMBER CONVEX-2-D-OBJECTS ?V1) (NAME-OF ?V1 '(TAYLOR HALL)))
 (:THE ?V2 (MEMBER BUILDINGS ?V2)
           (:BIND ?VECT (GET-INTRINSIC-FRONT-DIRECTION-IF-ANY '?V1))
           (GET-DIRECTIONAL-RELATION-USING-ALTERNATE-FRAME-OF-REFERENCE
             STRICTLY-BEHIND ?V2 ?V1 ?VECT))
 (NAME-OF ?V2 '(AEROSPACE-ENGINEERING BUILDING)))
```

Set of named objects after processing sentences 2-5 in Figure 12:

```
algy> (ask '((member convex-2-d-objects ?x) (name-of ?x ?y))
           :collect '(?x ?y))
  ((B (AEROSPACE-ENGINEERING BUILDING)) (B1 (BIOLOGY BUILDING))
   (B2 (WELCH HALL)) (B3 (TAYLOR HALL)) (O RAMAN))
```

Now let us proceed to name the streets. Below, we show the Algernon query forms generated for sentences 6 and 7 in Figure 12. Notice that there are no attempts to identify an intrinsic frame of reference since the spatial relations to be proven, enclosed-in and intersecting, are not directional relations. The 'ask' query at the end reveals that streets R and R1 have been named.

```
algy> (interpret '(I am traveling on Twenty-Fourth street))

 The following Algernon path shall be asserted:

((:THE ?V1 (MEMBER SPEAKERS ?V1))
 (:THE ?V2 (MEMBER ROADS ?V2) (ENCLOSED-IN ?V1 ?V2))
 (NAME-OF ?V2 '(TWENTY-FOURTH STREET))
 (ACTION-OF ?V1 'TRAVELING))

algy> (interpret '(Speedway street intersects Twenty-Fourth street))

 The following Algernon path shall be asserted:

((:THE ?V1 (MEMBER ROADS ?V1) (NAME-OF ?V1 '(TWENTY-FOURTH STREET)))
 (:THE ?V2 (MEMBER ROADS ?V2) (INTERSECTING ?V2 ?V1))
 (NAME-OF ?V2 '(SPEEDWAY STREET)))
```

Set of named roads in the KB after processing sentences 6-7 in Figure 12:

```
algy> (ask '((member convex-2-d-objects ?x)
            (isa ?x roads) (name-of ?x ?y)) :collect '(?x ?y))
((R (SPEEDWAY STREET)) (R1 (TWENTY-FOURTH STREET)))
```

Let us now consider the last sentence in Figure 12, 'Ben is crossing Speedway Street'. This sentence names one of the pedestrians after uniquely identifying the street that the pedestrian is crossing. Note that in addition to the name of the street, Speedway, we could have instead specified the direction of the street, NS or EW, since that is also a known property (given in the picture semantics) that distinguishes the two streets in Figure 10. Since it can be inferred that Ben is not the speaker, the query looks for a person other than the speaker.

```
algy> (interpret '(Ben is crossing speedway street))

 The following Algernon path shall be asserted:

((:THE ?V1 (MEMBER ROADS ?V1) (NAME-OF ?V1 (:QUOTE (SPEEDWAY STREET))))
 (:THE ?V2 (MEMBER PEOPLE ?V2)
          (:UNP (ISA ?V2 SPEAKERS)) (ENCLOSED-IN ?V2 ?V1))
 (NAME-OF ?V2 (:QUOTE BEN)))

algy> (ask '((member people ?x) (name-of ?x ?y)) :collect '(?x ?y))
  ((P1 BEN) (O RAMAN))
```

5. Discussion

This paper has addressed the question "Are there any simple, but unrestrictive, methods that allow diagrams and text to be used as computer input?". The goal is to take advantage of the power of diagrams for describing a spatial state (Larkin *et al.*, 1987; Narayanan *et al.*, 1994) and yet retain the ability to use text-based input to describe other contextual properties of interest. This goal requires the development of solutions to the *coreference problem* - matching object references in text to objects in the diagram.

Previous work in this area has resulted in the development of methods for automating the ability to understand text and diagrammatic input intended for *human communication* (He *et al.*, 1994; Novak *et al.*, 1993; Rowe *et al.*, 1993; Srihari, 1994). This assumption makes the coreference problem particularly complex and difficult for several reasons: (1) There can be hidden assumptions on the background knowledge of the human reader which may be not be available to an automated reasoning system; (2) Even the task of independently understanding each form of input may be difficult. Understanding pictorial input in visual (raster) form requires visual scene interpretation (Ballard *et al.*, 1982), and processing text intended for human-to-human communication can require a sophisticated natural language processing system to resolve the ambiguities that may be present.

We have described a simpler methodology, the Picture Semantics description, and an implemented tool, the Figure Understander, which allows multimedia input to a knowledge base in the form of a graphically-drawn diagram and a collection of text sentences to be integrated into a single unified description.

By using graphical editors to provide diagrammatic input, we gain two advantages over systems that have to deal with visual input. First, graphical editors directly identify the objects of interest by providing a *geometric description* of the objects in the scene. If the additional constraint that diagram objects must be in one-to-one correspondence with the objects to be created in the knowledge base is enforced, there is no need for visual scene interpretation.

Second, graphical editors allow additional properties to be attached to objects such as patterns, colors, and brush types. Our system uses a special input form, the Picture Semantics, to associate contextual knowledge with these graphical properties. This approach provides a simple and elegant solution to the coreference problem by allowing text and diagram objects to be matched based on both spatial and domain-specific contextual information.

5.1. Applicability of our solution

We impose strict restrictions on the contents of both the diagram and the text. For example, unlike the work of Srihari (1994), we do not consider the possibility that there may be object references in the text for which there is no corresponding object in the diagram. To avoid ambiguity in establishing coreference, we also require that there is at least one unique object in the diagram which can be matched against a text object based on its Picture Semantics description alone. We further assume that it will be possible to use spatial relations in the text between such uniquely identifiable objects and the non-unique diagram objects to isolate any specific non-unique diagram object.

The reader may ask, 'Aren't these requirements too restrictive?'. The answer is yes, if, as with the work of Srihari (1994), Rowe (1993), and Novak (1993), the goal is to automatically process text and diagram input intended for human use, such a problem statement in a textbook, or a newspaper photograph. That is why their solutions to the coreference problem are necessarily more complex.

The solutions provided in this paper are intended for users who simply wish to take advantage of diagrams to compactly input the many spatial properties of a spatial scene, but who need another means, such as text, to provide additional domain-specific information about the objects in the diagram. In these situations, it is quite reasonable to assume that the user has a great interest in providing ambiguity-free input descriptions.

Just as we are making use of the Figure Understander to process diagram and text descriptions of a problem in our work on qualitative reasoning about the effects of magnetic fields (Rajagopalan, 1994a), the work of Hari Narayanan, et al. (1994) and He, et al. (1994) could also benefit from the simple approach presented in this paper for using diagrams and text as computer input.

Further, although the focus of this paper has been on our solutions for the coreference problem, the reader may find the diagram processing and spatial reasoning capabilities of the Figure Understander to be worthy of further study. The Figure Understander automatically chooses bounding shape abstractions for objects in diagrams, can choose a frame of reference for a diagram to maximize the alignments of vectors and object surfaces with the coordinate axes, and automatically compensates for simple human errors in drawing a diagram, such as slight misalignments between objects. The Figure Understander also automatically extracts a qualitative description of a spatial scene from an input diagram, including the shapes (if the shape is simple) of objects, the connectivity (in terms of edges and vertices) of polygons, the relative positions of objects, and the orientation of objects.

Although the Figure Understander reasons about a static input scene, we have also developed methods for qualitatively reasoning about simultaneous dynamic changes in both the spatial and non-spatial properties of a physical system. The interested reader may consult (Rajagopalan, 1994a) or (Rajagopalan, 1995) for further details.

5.2. Related issues

This paper has focused on a very specific problem - providing an efficient method for using text and diagrams as computer input. Some related research issues include the generation of explanations using both text and graphical input (Feiner *et al.*, 1990), generating a pictorial description from a text description of a scene (Gapp, 1994; Latecki *et al.*, 1992; Olivier *et al.*, 1994), and generating a text description from a visual image of a scene (Maass, 1994).

The solutions to these issues focus less on associating pictorial objects with text objects, and instead, concentrate on the modeling and use of spatial relations. For example, Gapp (1994), Latecki and Pribbenow (1992), and Olivier, et al., (1994) have all pointed out that the interpretation of distance-based topological relations, such as near and far, and those relating two paths, such as along, can be highly context dependent. They have defined both heuristic and quantitative methods for evaluating the applicability of such topological relations to a set of objects in a given spatial scene.

Acknowledgements

I wish to thank Professor Benjamin Kuipers, Paul McKevitt, and the anonymous reviewers for their many suggestions in developing and clarifying the ideas presented in this paper. This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-8904454, IRI-9017047, and IRI-9216584, and by NASA contracts NCC 2-760 and NAG 9-665.

References

Ballard, D.H. and Brown, C.M. (1982). *Computer Vision*, Prentice Hall.

Chang, S. K., Shi, Q. Y., and Yan, C. W. (1987). Iconic Indexing by 2-D Strings *IEEE Transactions on Pattern Recognition and Machine Intelligence*, **9**:413-428.

Crawford, J. and Kuipers, B. (1991). Algernon - A Tractable System for Knowledge-Representation. In Working Notes of *The AAAI Spring Symposium on Implemented Knowledge Representation and Reasoning Systems*, Palo Alto, CA: American Association for Artificial Intelligence.

Feiner, S. and McKeown, K. (1990). Coordinating Text and Graphics in Explanation Generation. In Proceedings of *The Eighth National Conference on Artificial Intelligence*, 442-449, Boston, MA: American Association for Artificial Intelligence.

Freksa, C. (1992). Using Orientation Information for Qualitative Spatial Reasoning. In Frank, A., Campari, I., and Formentini, U. (Eds.) *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, 162-178, Springer-Verlag: Berlin.

Gapp, K. (1994). Basic Meanings of Spatial Relations: Computation and Evaluation in 3D Space. In Proceedings of *The Twelfth National Conference on Artificial Intelligence*, 1393-1398, Seattle, WA: American Association for Artificial Intelligence.

He, S., Abe, N., and Kitahashi, T. (1994) Assembly Plan Generation by Integrating Pictorial and Textual Information in an Assembly Illustration. In Working Notes of *The AAAI-94 Workshop on Integration of Natural Language and Vision Processing*, 66-73, Seattle, WA: American Association for Artificial Intelligence.

Herskovits, A. (1985). Semantics and Pragmatics of Locative Expressions. *Cognitive Science* **9**:341-378.

Jungert, E. (1992). The Observer's Point of View: An Extension of Symbolic Projections. In Frank, A., Campari, I., and Formentini, U. (Eds.) *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, 179-195, Springer-Verlag: Berlin.

Kuipers, B. (1994) *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge.* MIT Press: Cambridge, MA.

Landau, B. and Jackendoff, R. (1993). "What" and "Where" in Spatial Language and Spatial Cognition". *Behavioral and Brain Sciences* **16**:217-265.

Larkin, J. and Simon, H. (1987). Why a Diagram is (Sometimes) Worth 10,000 Words. *Cognitive Science* **11**:65-99.

Latecki, L. and Pribbenow, S. (1992). On Hybrid Reasoning for Processing Spatial Expressions. In Proceedings of *The Tenth European Conference on Artificial Intelligence*, 389-393, Vienna: European Coordinating Committee for Artificial Intelligence.

Maass, W. (1994). From Visual Perception to Multimodal Communication: Incremental Route Descriptions. *AI Review Journal*, **8**.

McKevitt, P. (Ed.) (1994) *Working Notes of the AAAI Workshop on Integration of Natural Language and Vision Processing.* American Association for Artificial Intelligence: Menlo Park, CA.

Mukerjee, A. and Joe, G. (1990). A Qualitative Model for Space. In Proceedings of *The Eighth National Conference on Artificial Intelligence*, Boston, MA: American Association for Artificial Intelligence.

Narayanan, N. Hari (Ed.) (1992). *Working Notes of the AAAI Spring Symposium Series, Symposium: Reasoning with Diagrammatic Representations.* American Association for Artificial Intelligence: Menlo Park, CA.

Narayanan, N. Hari, Suwa, M., and Motoda, H. (1994). How Things Appear to Work: Predicting Behaviors from Device Diagrams. In Proceedings of *The Twelfth National Conference on Artificial Intelligence*, 1161-1166, Seattle, WA: American Association for Artificial Intelligence.

Nielsen, P. (1988). A Qualitative Approach to Mechanical Constraint. In Proceedings of *The Seventh National Conference on Artificial Intelligence*, 270-274, Saint Paul, MN: American Association for Artificial Intelligence.

Novak, G. S., and Bulko, W. (1993). Diagrams and Text as Computer Input. *Journal of Visual Languages and Computing* **4**:161-175.

Olivier, P., Maeda, T., and Tsujii, J. (1994). Automatic Depiction of Spatial Descriptions In Proceedings of *The Twelfth National Conference on Artificial Intelligence*, 1405-1410, Seattle, WA: American Association for Artificial Intelligence.

Rajagopalan, R. (1994a). A Model for Integrated Spatial and Dynamic Reasoning about Physical Systems. In Proceedings of *The Twelfth National Conference on Artificial Intelligence*, 1411-1417, Seattle, WA: American Association for Artificial Intelligence.

Rajagopalan, R. (1994b). The Figure Understander: A Tool for the Integration of Text and Graphical Input to a Knowledge Base. In Proceedings of *The Sixth IEEE International Conference on Tools with Artificial Intelligence*, 80-87, New Orleans, LA: IEEE Computer Society.

Rajagopalan, R. (1995). *Qualitative Reasoning about Dynamic Change in the Spatial Properties of a Physical System.* Ph.D. Diss., Department of Computer Sciences, University of Texas at Austin, Austin, TX (forthcoming).

Rajagopalan, R. and Kuipers, B. (1994). The Figure Understander: A System for Integrating Text and Diagram Input to a Knowledge Base. In Proceedings of *The Seventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, 211-220, Austin, TX: International Society of Applied Intelligence.

Resnick, J., and Halliday, D. (1988) *Fundamentals of Physics*, John Wiley and Sons: New York.

Retz-Schmidt, G. (1988). Various Views on Spatial Prepositions. *AI magazine* **9**:95-105.

Rowe, N. and Guglielmo, E. (1993). Exploiting captions in retrieval of multimedia data. *Information Processing and Management* **29**:453-461.

Srihari, R. (1994). Use of Captions and other Collateral Text in Understanding Photographs. *AI Review Journal* **8**.