VEIL: Combining Semantic Knowledge with Image Understanding

Thomas A. Russ, Robert M. MacGregor and Behnam Salemi
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
{tar|macgregor}@isi.edu salemi@pollux.usc.edu

Keith Price and Ram Nevatia
USC Institute for Robotics and Intelligent Systems
{price|nevatia}@iris.usc.edu

Abstract

The VEIL system is pioneering two thrusts within the field of image understanding that until now have received relatively little attention. First, VEIL reasons at a semantic level about scene information, using a knowledge base that augments an original image understanding (IU) model (representing the output of a lower level IU system) with structural and functional information. VEIL can apply both spatial and temporal reasoning to detect event sequences that span multiple images. Second, VEIL provides users with an intelligent interface that relies on a library of domain-specific terms and queries to provide a domain-specific browsing and query facility. The library, implemented in Loom as an extensible ontology of domain specific definitions and queries, can be easily customized by a user to facilitate analysis activities.

1 Introduction

VEIL performs high level interpretation of scene images using the deductive capabilities of the Loom knowledge representation system [MacGregor and Bates 1987, Brill 1993],, and provides users with semantically-enriched browsing and editing capabilities. The VEIL experiments use a database of RADIUS Model Board 2 image site models stored in SRI's RCDE system. This database is augmented by a knowledge base stored in Loom that includes references to the underlying RCDE-object models, representations of functional and structural knowledge not contained in the RCDE model, and a library of high-level spatial reasoning functions. The Loom knowledge base also contains abstract definitions for objects and events. Using this architecture as a base, VEIL supports queries that search within an image to retrieve concrete or abstract objects; or that search across images to retrieve images that contain specific objects or events. An event in VEIL is comprised of entities and/or subevents that collectively satisfy a set of temporal and spatial constraints. VEIL can scan a sequence of images and detect complex events. In the example presented in this paper, VEIL finds Field Training Exercise events consisting of four subevents occurring in distinct images.

VEIL is implemented as a modular architecture wherein all communication between Loom and the underlying IU system is mediated by RCDE protocols and data structures [RADIUS Manual 1993],. In the future, it would be practical for us to incorporate multiple IU systems into the VEIL architecture. Also, VEIL could be exported to other sites along with RCDE, allowing other IU researchers to connect their systems to VEIL. Thus, VEIL provides a generic means for extending a RCDE-based IU system to include semantic processing of image data. Use of VEIL also promotes the use of explicit, declarative domain models. We forecast that this approach will be a key enabling technology when it becomes time to interconnect image understanding systems with other knowledge-intensive systems and applications.

2 Underlying Technology

We are extending the semantics of the information that is captured by an image understanding program by associating domain-level information with images. We use the following terminology. The *image* means the digital input data. For our examples these are photographs. The *site model* is a geometric model of objects found in a particular image. Such objects can be roughly divided into objects representing terrain features, structures and vehicles. A *domain model* is a

This research project was supported by the Department of Defense Advanced Research Projects Administration under contract number F30602–93–C–0064.

semantic model of items of interest in the domain. This includes buildings and vehicles as well as abstract notions such as the function of objects, groups of objects, convoys and field training exercise events.

2.1 RADIUS

Our experiments use the forty RADIUS Model Board 2 images of a hypothetical armored brigade garrison and exercise area. A site model common to all forty images was provided by the University of Maryland. This RCDE-object site model was used with only minor modifications in our work.[1] We augmented the common site model with vehicle models for a subset of ten images. Vehicles were identified by a graduate student and their location noted in a file. Vehicle model objects are needed for VEIL's event processing, but the source of the models is irrelevant. A suitable automatic vehicle detector could be substituted for our manual methods.

2.2 Loom

We use Loom, an AI knowledge representation language in the KL-ONE family, to provide the infrastructure for semantic reasoning. Loom provides the following benefits:

- Declarative language. Information is encoded in an easy-to-understand format. This makes it easy to comprehend and extend the model.
- Well-defined semantics. The meaning of language constructs is well-defined. The meaning of the terminology is well-established and validated by over 15 years of AI research into description logics [Brachman 1979, Brachman *et al.* 1983].
- Expressivity. Loom is one of the most expressive languages in its class.
- Contexts. Assertions (facts) about multiple images can be accessed at the same time. This is a key feature used in recognizing events.

2.2.1 Definitions

Loom reasons with *definitions*, which equate a term with a system-understood description in terms of necessary and sufficient conditions. This allows useful flexibility in reasoning for a recognition domain. Combined with a hierarchy of concepts one is able to make assertions that precisely capture the amount of information available. When details are not known, one is not forced to overcommit in making entries to the knowledge base. As more information becomes available it can be added incrementally, improving the picture of the world. If enough additional information is added, Loom's classifier automatically recognizes an instance as belonging to a more specific concept.

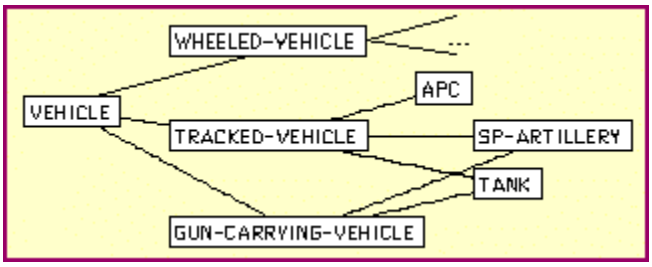

Fig. 1. Vehicle Hierarchy of the Domain Model

We will illustrate how this works using the fragment of the domain model for vehicles shown in Figure 1. Suppose that the first pass of processing is able to identify some group of pixels in the image as a vehicle. Details about the type of vehicle are not yet known, so the information is entered as a "vehicle V1 in location X." With further processing, it may be determined that the vehicle has tracks. This information can be added, to the knowledge base, allowing the classification of the vehicle as a tracked vehicle. The classifier is able to perform this inference because the definition of a "tracked-vehicle" is a vehicle with a drive type of tracks. Since V1 now satisfies the definition, Loom automatically concludes that it is of type Tracked-Vehicle. If an appropriate type of gun is detected, V1 may finally be recognized as a tank.

By using definitions, Loom can make the inferences licensed by the definitions automatically. This service frees applications built on top of Loom from needing to implement their own inference mechanism.

Since Loom's definitions are true equivalences, they can be used to reason in both directions. The example above illustrated using the components of a definition to perform a recognition task—synthetic reasoning. One can also assert the presence of higher level objects and then use the definitions to identify components that should be present.

For example, a particular SAM unit may be known to deploy with a radar vehicle and three launchers. If such a unit is asserted to exist in a scene, Loom concludes that there are three launchers present, even if they are not identified. The definition can then be used as a guide to what other objects should be present. It can be used to drive the reasoning. This type of reasoning was used in another part of the VEIL project that identified runways [Price *et al.* 1994].

2.2.2 Contexts

Loom has a context mechanism that allows one to maintain distinct assertion sets. Loom's contexts are

[1] The modifications were to ensure a consistent composite grouping of buildings which were represented in the site model as multiple cubes. Several of such complex-structure buildings were already present as composite objects. We manually rounded out the site model to assure consistency in the modeling.

organized hierarchically, which allows inheritance. Siblings are separate, so this allows information about different scenes to be kept separate, but in the same lisp image. The query language (see below) is able to perform queries across contexts, so one can make comparisons and look for particular patterns.

Augmenting this flexibility is the fact that Loom contexts are themselves first-class objects. That means that assertions and annotations about the context themselves can be represented in the Loom formalism and be used to select appropriate contexts. This capability was added to Loom version 3.0 in response to the needs of the VEIL project. For example, if one had a context associated with a particular image, one could annotate the context with information such as sun angle, time of day, camera location, etc. This information is available for image retrieval purposes. At the end of this paper, we will discuss the use of this context mechanism in event detection.. Event detection will involve searching for a sequence of images (contexts) that fulfill the criteria for a given event. This uses the ability of Loom to have several image descriptions in memory simultaneously as well as the ability to formulate and execute queries that cover several images.

2.2.3 Query Mechanism

Loom includes a general query facility that is able to find objects based on their name, type or value of role (relation) fillers.

Queries for Particular Objects: Specific objects can be queried for in images. Examples include looking for all buildings, all headquarters, all tanks, etc. These queries allow a seamless use of collateral information in the RCDE system.

Queries for Relationships: In addition to queries that relate to single objects, one can also query about relationships. Examples include finding all buildings with an area of more than 5000 square feet, locating all tanks on roads, or finding headquarters that are near barracks, etc.

Loom queries are not restricted to single images, but can extend across images. This type of query is used in the event detection example below.

3 The Domain Model

A prototype knowledge base containing domain concepts was created for use in VEIL. The type of knowledge encoded in this domain model ranged from the concrete to the abstract. Loom models for concrete, visible objects such as roads, buildings and vehicles (see Fig. 2) are linked to geometric objects in the RCDE site model. Collateral information about objects in a scene, such as "building B44 is a brigade headquarters", is associated with the Loom instance representing the building. Abstract concepts such as groups, functional roles and events are used to augment reasoning about the concrete objects.

The main example that we use in VEIL is the concept of a group of vehicles.[2] Abstract entities can be specialized based on their characteristics. VEIL defines a convoy as a group of vehicles with at least 65% of them on a road. Additional constraints can be added such as requiring a minimum number of vehicles (i.e., >4). Loom's flexible knowledge representation easily supports specialization such as defining a convoy of tanks.

The definition of a convoy combines information that is present in the Loom level (such as group membership) with information that is inherently geographic (such as the location of vehicles on roads). Loom's forte is symbolic reasoning. Determination of geographic location is geometric reasoning that is best handled using RCDE model structures. Accordingly, we have developed several representative and interesting geometric predicates and linked them to Loom relations. Reasoning is performed at the appropriate level and the results integrated by Loom.

3.1 Linking the Domain and Site Models

At the domain model level, the geometric information about the objects is not directly available. Instead, reasoning is focused on the function and wider role of the objects. At the geometric level, information about the location and size of objects is either directly available or computable from directly available information. For example, the location of a particular cube object is readily available and its volume can be easily computed using information stored about the length of the sides of the cube.

We have implemented several functions at the geometric level which are linked to Loom relations. Table 1 summarizes the basic relations. The most fundamental predicate is the one that returns locations. Given the three-space location of objects we implemented directional relations (north, northeast, etc.), We have also implemented computations for the area and volume of the most common geometric objects used in the site models.

Loom relations were linked to these functions. This enables Loom queries to seamlessly exploit both the semantic information contained in Loom's domain model as well as the geometric information from the underlying site model. An example of such a composite query is to find "all vehicle storage sheds with a floor area greater than 5,000 square feet.":

[2] In the current implementation, groups are created by humans. Future work extending our ideas would involve providing tools for moving this into a semi-automated task.

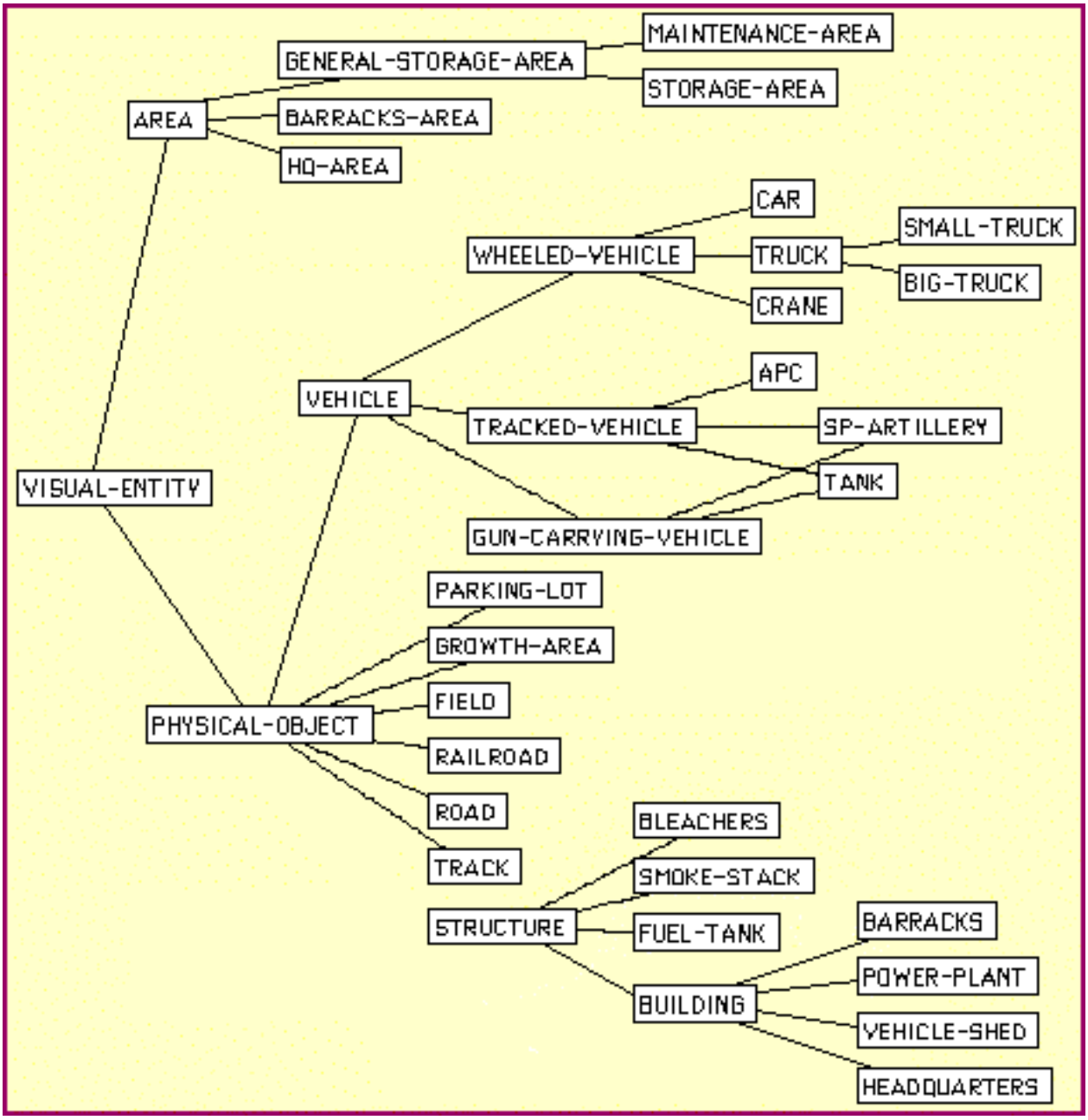

Fig. 2 Domain Model for Visible Objects in VEIL

```
(retrieve ?shed
   (and (vehicle-shed ?shed)
        (> (area ?shed) 5000)))
```

The concept *vehicle-shed* and the relation > are domain level operators. The relation *area* is a domain level relation that is linked to a site model level function.

3.2 Geometric Relations

Geometric relations can also be computed between objects. We implemented a containment test (contains-point-p), which tests to see if a given three-space point is contained in a 3 dimensional object (or located over a 2 dimensional object). This predicate is used in queries and concept definitions to locate vehicles that are on roads.— for example in the concepts of vehicles in a convoy.

One of the more interesting relations that we have investigated is the "is-near" relation. This is a subjective relation adopted from the nearness predicate in Abella's Ph.D. thesis [Abella 95]. Her studies found that a psychologically valid implementation of nearness was influenced by the size of the objects in

RCDE	Geometric Function				
Object Type	Location	Contains-point-p	Is-Near	Area	Volume
CUBE-OBJECT	X	X	X	X	X
CYLINDER	X	X	X	X	X
HOUSE-OBJECT	X	X	X	X	X
3D-CLOSED-CURVE	X	X	X	X	
3D-RIBBON-CURVE	X	X	X	X	
COMPOSITE-OBJECT	X	X	X	X	X
Others	X				

Table 1. Geometric Functions and RCDE Objects

question. In other words, the larger the object, the farther away one could be in absolute distance while still being considered near. She developed a function that computes an "extended bounding box" for each object, based on the object's dimensions. When two bounding boxes intersect, the objects are "near".

We extended her formula to three dimensions. For buildings or vehicles, this yields appropriate results. The approach breaks down when the aspect ratio becomes very large. Extremely long, thin objects end up with very large bounding boxes because of the effect of their length on the size of the nearness boundary. Roads are a prime examples from the site model that we use. The length of a road influences how far away one can be from a road and still be considered *near*. This produces unintuitive results.

We therefore modified the algorithm for the case of long, thin objects. Objects with a large aspect ration disregard the long dimension when computing the nearness boundary. This modification produces appropriate results for our purposes., Figure 3 shows an image and the associated extended bounding boxes of a curved road and two buildings. Building 2 (on the right) satisfies the "near-to" relation with respect to the curved road, but Building 1 (on the left) does not.[3]

[3]The road is shown divided into bounding boxes segment-wise. The rectangular boxes are used for a rough test of nearness. A

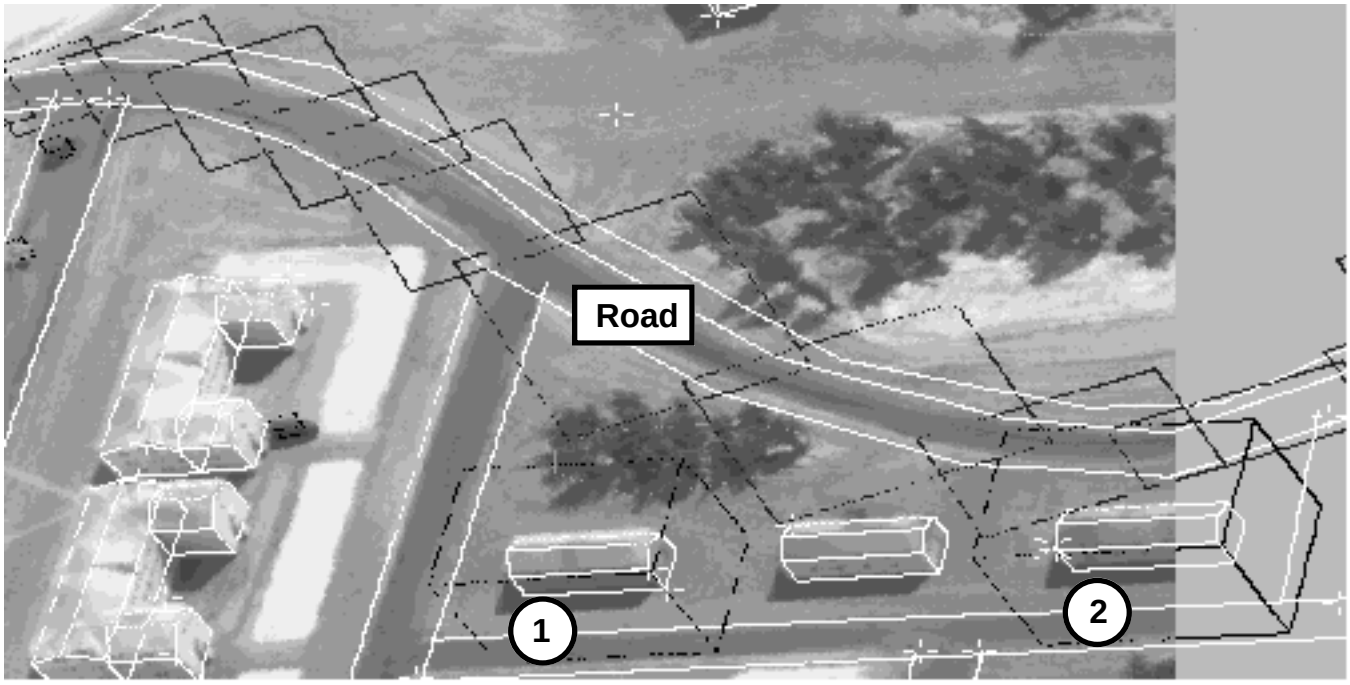

Fig. 3. Extended Bounding Boxes for Computing Nearness

Site model objects have white boundaries. Extended bounding boxes for selected objects are black.

```
(make-event
  :name 'field-training-exercise
  :case-roles '((armored-unit ?y))
  :components
    '((:scene ?s1 ?y (in-garrison ?y))
      (:scene ?s2 ?y (convoy ?y))
      (:scene ?s3 ?y (deployed-unit ?y))
      (:scene ?s4 ?y (convoy ?y)))

  :constraints '((before+ ?s1 ?s2)
                 (before+ ?s2 ?s3)
                 (before+ ?s3 ?s4))))
```

```
(retrieve (?S1 ?S2 ?S3 ?S4 ?Y)
          (and (within-context ?S1
                    (In-Garrison ?Y))
               (within-context ?S2
                    (Convoy ?Y))
               (within-context ?S3
                    (Deployed-Unit ?Y))
               (within-context ?S4
                    (Convoy ?Y))
               (before+ ?S1 ?S2)
               (before+ ?S2 ?S3)
               (before+ ?S3 ?S4))
```

Fig. 4. Event Definition and Corresponding Loom Query

4 Event Detection

In this section we describe how we define events — objects that satisfy constraints both within and across images. We also outline how VEIL is able to locate such events in its database.

4.1 A definition language

An event is a sequence of *scenes* which satisfy certain criteria. Some of the criteria apply within scenes whereas other criteria describe the relationship between different scenes. Accordingly, we defined a language that allows these constraints to be specified in a natural way. The scenes in an event are described separately, specifying any criteria that apply within a single scene. A set of global constraints is then used to specify the conditions that must hold between scenes. The most common cross-scene constraint is that of order. A sequence of scenes implies that there is an ordering to the scenes.

4.2 Sample event definitions

Figure 4 shows an event definition named "Field-Training-Exercise" and its associated Loom query. The event consists of four scenes involving an armored unit "?y". The scenes must include one with ?y "in-garrison", two scenes with ?y in convoy and one with ?y deployed. In addition, the scenes are constrained temporally by the :constraints field. Translating this into English, we are looking for a sequence of scenes showing an armored unit in a garrison, then moving in convoy, then deployed in a training area and finally in convoy again. A set of images showing this evolution is shown in the example below.

4.3 Example of Event Detection

Figure 5 shows a master view of the ten images we used in our experiments. An example of a field training exercise event is highlighted. Figure 6 shows a close-up of the field training exercise with the objects participating in the event highlighted. A colored box is drawn around the group of vehicles in each image. (In these figures, the box has been enhanced for better black-and-white printing).

4.4 How it's done

The Loom query in figure 4 is used to extract those scenes which meet the event criteria. This involves satisfying the conditions for each individual scene (such as finding a group that is in a garrison area in a scene) and also satisfying the cross-scene constraints (such as being in a particular temporal order). The Loom query for the event shown above is represented as follows:

The result of this query will be a set of tuples. Each tuple consists of the four scenes (images) and the group that satisfies the query. The display in the example was created automatically from one such match.

Because of Loom's named definitions, the query for finding events is quite compact and reasonably readable. This shows the power of having a domain-specific language: even complex criteria can be expressed in a concise and natural manner.

5 Current Status

The current VEIL model has been tested using ten images from the RADIUS Model Board 2 image set. It is integrated with the RCDE code and uses the RCDE graphics interface for user interaction and display purposes. The figures in this paper are screen shots.

6 Future Work

There are several directions for extending our research. One major direction would be to improve the matching algorithm used to find events. The current match relies on using Loom's general query mechanism. While this provides flexibility, the logic-based query language does not take advantage of special features of

more sophisticated test which implements a smooth envelope is used for the final comparison.

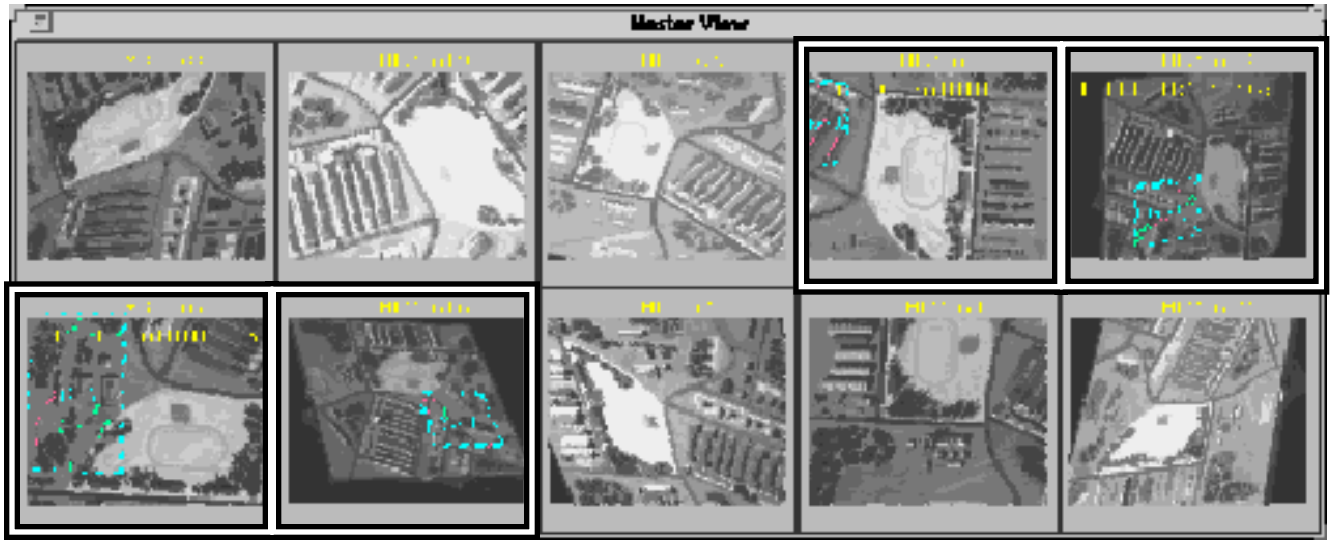

Fig. 5. Field Training Exercise Event Found in an Image Sequence

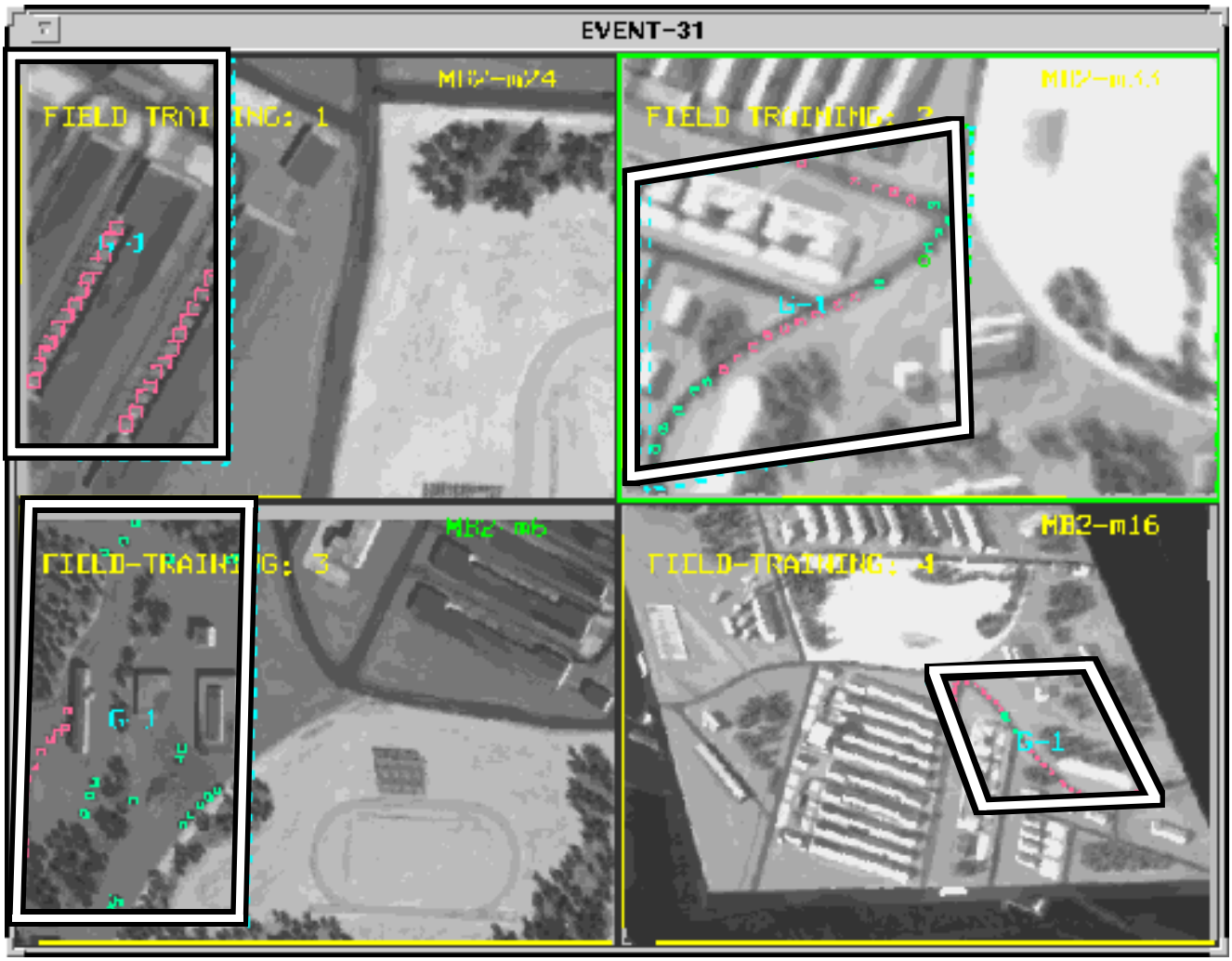

Fig. 6. Close-Up View of Field Training Exercise

the event matching that can be used to increase efficiency. For example, there is no direct exploitation of the fact that a sequence is being looked for. Additional enhancements would be to modify the event matching language to allow inexact matches. This can take the form of partial matches, matches to key features but with missing elements, or a more general probabilistic matching scheme.

A sub-problem of the general matching task is associating groups from one image with those from a different image. (In the current work, such matching is done by hand). An interim position would be to use a credulous matcher, although that would need to be refined in order to scale well. The preferred approach would be to develop a compatibility score for matches between groups in one image and groups in another image. This score would be based on factors such as the size of the group, the composition of the group (i.e., with or without tanks), as well as heuristic reasoning based on other elements that are visible in an image. With a more sophisticated matcher, a list of candidate image sequences can be identified and ranked as to the closeness of the match.

Computer support for assigning individual vehicles to groups is another area for further investigation. The group assignment problem involves identifying a collection of vehicle that are related in some interesting way. Geometric proximity is one important consideration, but it is not always he most important. Consider a convoy driving by a parking lot. Some vehicles in the convoy will be closer to parked vehicles than to other convoy vehicles, but the importance of being on the road should be given more weight in the group assignment process. A semi-automated grouping tool would be a useful addition to RCDE.

7 Conclusion

The bulk of work in IU research has been on developing algorithms that operate at the pixel level and are able to recognize geometric objects. Common examples are edge detectors, building detectors and vehicle detectors. In our work, we have been investigating the next stage of image understanding. In other words, we are concerned with the question of what sort of processing would we like to have happen once the low-level detectors have finished their work.

We feel that the next step involves reasoning with the aid of domain models—models of the world. This raises the level of abstraction of the interface between the image analyst and the computer system. Instead of operating at the level of pixels or geometric shapes, one would like to have the interface operate at a level that has the appropriate semantic content for the task at hand. This level would allow interaction in terms of headquarters rather than buildings, convoys rather than isolated vehicles. By raising the level of interaction, better use of an image analyst's time can be made.

By increasing the level of abstraction and allowing queries at that level, it becomes easier to select appropriate images for viewing out of a large library. By raising the level of abstraction, we are also able to describe events that cover multiple images naturally and locate them efficiently.

References

[Abella 1995] Alicia Abella., *From Imagery to Salience: Locative Expressions in Context*, Ph.D. thesis, Columbia University, 1995.

[Brachman *et al.* 1983] Ronald J. Brachman, Richard E. Fikes, and Hector J. Levesque, KRYPTON: A functional approach to knowledge representation, *IEEE Computer*, **16**(10):67–73, 1983. (Revised version reprinted in Ronald J. Brachman and Hector J. Levesque, editors. *Readings in Knowledge Representation*, Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1985., pp. 412–429.)

[Brachman 1979] Ronald J. Brachman, *On the Epistemological Status of Semantic Networks*, pages 3–50, 1979. (Reprinted in Ronald J. Brachman and Hector J. Levesque, editors. *Readings in Knowledge Representation*, Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1985. pp. 192–215)

[Brill 1993] David Brill, *Loom Reference Manual*, Version 2.0, USC/ISI, 4676 Admiralty Way, Marina del Rey, CA 90292, December 1993.

[MacGregor and Bates 1987], Robert MacGregor and Raymond Bates, The Loom knowledge representation language, Technical Report ISI/RS–87–188, USC/Information Sciences Institute, 1987.

[Price *et al.* 1994] Keith Price, Thomas Russ, and Robert MacGregor, Knowledge representation for computer vision: The VEIL project, ARPA Image Understanding Workshop, 1994.

[RADIUS Manual 1993], SRI International and Martin Marietta, *RADIUS Common Development Environment: Programmer's Reference Manual*, version 1.0 edition, July 1993.