MITTEILUNG 131

A RELATIONAL MATCHING STRATEGY FOR TEMPORAL EVENT RECOGNITION

Hans-Joachim Novak

IFI-HH-M-131/85

January 1985

This report is based on a talk which was held at the GWAI-84.

ZUSAMMENFASSUNG

Relationale Beschreibungen werden bisher in der Bildverarbeitung vorzugsweise zur Objekterkennung eingesetzt. In dieser Arbeit wird eine Erweiterung des Relationalvergleichs vorgestellt, die der Erkennung zeitlich veränderlicher Vorgänge, sogenannter Ereignisse (events), dient. Mit diesem erweiterten Relationalvergleich ist es möglich, sowohl die zeitliche Ausdehnung von Ereignissen als auch deren inhärente Eigenschaften wie Durativität und Nicht-Durativität korrekt zu behandeln.

A RELATIONAL MATCHING STRATEGY FOR TEMPORAL EVENT RECOGNITION

Hans-Joachim Novak
Universität Hamburg, Fachbereich Informatik
Schlüterstr. 70, 2000 Hamburg 13
West Germany

Abstract

In this contribution a relational matching strategy is described. It allows to handle temporal information like the beginning and end of events, durativity and non-durativity correctly. The implementation and use of this strategy for temporal event recognition in the NAOS system is discussed.

Introduction

Relational descriptions for picture processing were first proposed by BARROW and POPPLESTONE [1]. Their aim is object recognition. The idea is to describe the image in terms of relations between image regions and to compare these descriptions with predefined relational models of objects. The result is the best match between the description and the model.

In [2] the matching process is described in detail. Especially the idea of hierarchical syntheses is put forth. The models are structured hierarchically and recognition proceeds by first finding the smaller substructures and then checking combinations of these to recognize larger substructures. This approach is shown to be computationally more efficient than matching large structures.

In summary, relations are used to describe both the image and the object models and matching is used to compare both descriptions. These ideas were first used for object recognition in single images.

Leaving the single image paradigm and turning to image sequences we are especially interested in time-varying properties of an image sequence called events.

In our case events are 'meaningful' parts of a scene (image sequence) insofar as an event may be verbalized using a verb of locomotion. Events are organized around locomotion verbs [3].

Event recognition starts when object recognition has been achieved. Thus a level of representation is assumed where the scene can be characterized in terms of objects and relations between them. The events are represented by event models. Event recognition proceeds by matching event models against the scene description. This is in analogy to the approach for object recognition described earlier. However, it will be shown that the matching process has to be extended in order to cope with problems arising from the nature of time varying events.

The overall goal of the NAOS system is the verbal description of the motions of objects in a traffic scene (cf. [4]). The scope of the present paper is the recognition of the events.

This paper describes a matching strategy for the recognition of temporal events which is implemented in the NAOS system. Therefore the representation of the scene is described first, second, the representation of event models is shown and in the last chapter the matching process and its neccessary extensions for recognizing temporal events are discussed.

Scene representation

Assuming a stationary camera, a scene consists of two parts, namely the non-moving objects (i.e. streets, buildings, etc.) and the moving objects. The stationary background (the non-moving objects) is recognized using a detailed street-model. The recognition of the form and trajectory of the moving objects builds on special processes as described e.g. in [5]. Presently our scene-analysis system cannot classify the moving objects like cars, pedestrians and cyclists. Therefore this classification is done interactively. A detailed description of the processes necessary to automatically construct the scene representation is contained in [6].

In our case the scene representation consists of the two parts:

- stationary background (instantiated street model)
- moving objects.

This representation - called geometrical scene description (GSD) - is an object centered representation associating all relevant information of an object with that object. In particular the GSD contains:

per <u>single image</u> of the scene

- time
- list of the objects
- viewer position and -orientation
- illumination

per <u>object</u>

- identity (in the sequence)
- 3D-form and -appearance
- 3D-position and -orientation
- class membership
- color
- functional features (e.g. the front of an object)

Without going into detail a section of the GSD is shown below. The LOCATION-entry has the form:

```
(LOCATION <internal name> <position> <orientation> <time1> {<time2>})
```

where the position is given by x, y and z coordinates of the object's center of mass and the orientation is a vector describing the direction into which the front of the object points.

```
(CLASS VW1 VW)
(COLOR VW1 YELLOW)
(CLASS TRUCK1 TRUCK)
(CLASS BUILDING1 BUILDING)
(NAME BUILDING1 "Dept. of CS")
(LOCATION BUILDING1 (100 -60 70) ( 0 1 0 ) 1 40)
(LOCATION VW1 ( -100 70 8 ) ( 4 1 0 ) 1)
(LOCATION VW1 ( -80 75 8 ) ( 4 1 0 ) 2)
          ↓
          ↓
          ↓
(LOCATION VW1 ( 875 50 8 )( 1 0 0 ) 31)
(LOCATION VW1 ( 880 50 8 )( 1 0 0 ) 32 40 )
(LOCATION TRUCK1 ( 50 50 15 )( 1 0 0 ) 1)
(LOCATION TRUCK1 ( 69 50 15 )( 1 0 0 ) 2)
```

The GSD contains a complete geometrical description of the original scene and is the basis for the event recognition process. In the next paragraph the representation of event models is described.

Event models

Due to the purpose of our system - verbal description of the motions of objects in a traffic scene - events are grouped around motion verbs. Events in our system are 'move', 'stop', 'accelerate', 'overtake', etc. Once an event is recognized it is known which verb may be used in a natural language description of the event. As an example the event model for 'overtake' is given below:

```
(OVERTAKE  OBJ1 OBJ2 T1  T2)
  (MOVE OBJ1  T1  T2)
  (MOVE OBJ2  T1  T2)
  (APPROACH OBJ1  OBJ2  T1  T3)
  (BEHIND OBJ1  OBJ2  T1  T3)
  (BESIDE OBJ1  OBJ2  T3  T4)
  (IN-FRONT-OF  OBJ1  OBJ2  T4  T2)
  (RECEDE OBJ1  OBJ2  T4  T2)
```

Informally the above event model may be read as follows. If OBJ1 overtakes OBJ2 in the time interval from T1 to T2 the following conditions must hold: Both objects move in the interval. In a subinterval from T3 to T4 which is within (T1 T2) the objects are beside each other. Before this OBJ1 approaches OBJ2 and afterwards OBJ1 recedes from OBJ2.

In general an event model consists of several relations (in the following often called propositions). Each relation itself consists of a relation identifier, e.g. MOVE, one or more variables, e.g. OBJ1, OBJ2, and time variables denoting the interval during which the relation is valid, e.g. T1 and T2. It is implicitly assumed that T1 < T2 if in a proposition T1 occurs left of T2, e.g. (MOVE OBJ1 T1 T2).

Three types of propositions are distinguished: primitive, composite and special. Primitive propositions are directly evaluated by specialized procedures using the GSD. MOVE is an example for a primitive proposition.

Composite propositions like OVERTAKE consist of several propositions which may themselves be composite again. Special propositions are used to evaluate temporal expressions like 'during' which do not directly refer to the GSD.

Propositions are evaluated by generating values for their variables so that the proposition is true. For composite propositions to be true the conjunction of the propositions they consist of must be true.

Event recognition

In this paragraph a detailed description of the matching strategy for temporal event recognition is given.

There are two major differences to the relational matching scheme described in [2]. First, in the beginning the GSD does not contain relations which could be directly matched against event models as the latter describe 'higher level concepts' which have to be computed from the basic ones contained in the GSD. The second major difference arises from the temporal dimension of events. If for example we know that the relation (MOVE CAR1 10 25) holds it might be necessary to verify that CAR1 moves in the interval from 12 to 20. A literal match of the pattern (MOVE CAR1 12 20) against (MOVE CAR1 10 25) will be unsuccessful. The time variables of the MOVE event are not independent but are interval boundaries and must be treated accordingly.

In the following a matching process is described which can handle time variables as required. First the overall evaluation strategy is explained.

In general a list of propositions must be evaluated to recognize an event, e.g.

```
((OVERTAKE OBJ1 OBJ2 TBEG TEND)
 (IN-FRONT-OF OBJ1 "Dept. of CS" TBEG TEND)),
```

which can be paraphrased as "Which object overtakes another one in front of the Department of Computer Science?".

Note, that we could as well want parts of the OVERTAKE event to be

IN-FRONT-OF our department by choosing different time variables for the IN-FRONT-OF proposition. Choosing T3 and T4 for instance, would imply that we want OBJ1 to be in front of the department while it is beside OBJ2 (see event model OVERTAKE above).

If for a specific event model or proposition there is no instance in the GSD, the model or proposition has to be evaluated. This is done by finding all instances and storing them in the GSD. Composite event models are evaluated recursively, primitive ones by specialized procedures. In general, lists of propositions are evaluated recursively.

The evaluation of a list of propositions may be viewed as a tree search. For an effective search it is necessary to evaluate the proposition with the highest branching-factor first. Consider the OVERTAKE event model. If more objects move and fewer objects approach other objects it is more efficient to evaluate the APPROACH proposition first. We distinguish between an intrinsic and an effective branching-factor. The intrinsic branching-factor of a proposition is an estimate of its probability of being true for arbitrary but fixed values of its variables. The intrinsic branching-factor is domain dependent and arises from experience and introspection. It is associated with the relation identifier. The effective branching-factor is computed by first multiplying the number of possibilities for assigning values to the variables of the proposition and then subtracting the intrinsic branching-factor from the reciprocal of this value. This is done at evaluation time. For event recognition, the proposition with the highest effective branching-factor is evaluated first, if it cannot be instantiated the process may stop at once neglecting the rest of the propositions. The effective branching-factor is recomputed after each evaluation of a proposition in order to take care of newly instantiated variables.

Event recognition is a two phase process embedded in a backtracking control structure. In the first phase all instances of a proposition are generated and added to the GSD. In the second phase the first instantiation is chosen and it is tested whether for the instantiated variables the remainder of the propositions can also be instantiated. Composite propositions are therefore expanded and the resulting list of propositions is evaluated. All composite propositions are thus finally reduced to primitive ones.

Note that there may be several instances of a proposition due to different time intervals, e.g. (MOVE CAR1 5 15) and (MOVE CAR1 25 40). Backtracking ensures that all these instances are tested for compatibility with the remaining propositions.

The event recognition is successful if for concrete values of the variables the conjunction of the propositions is true. It fails if a proposition cannot be instantiated in particular if it is not temporally compatible to the others.

The description of the overall evaluation strategy ends here. Next it is shown by means of an example that traditional relational matching is not sufficient for temporal event recognition.

Let us look closely at the evaluation of a list of propositions consisting of ((MOVE OBJ1 T1 T2) (MOVE OBJ2 T1 T2)). Possible values for the variables OBJ1 and OBJ2 are CAR1 and CAR2, CAR3 respectively. Let us further assume that at the end of the first phase for the first proposition the following instantiation has been found and added to the GSD: (MOVE CAR1 15 65).

In the second phase compatible instances have to be found. Therefore two backtrack-loops are constructed for each proposition. One ensuring that the non-time variables take all possible values and the other one ensuring that for each value all instantiations are tested.

In the above example the instantiation of the first proposition is chosen and it is tested whether the second proposition can be instantiated and has a compatible instantiation. The variable OBJ2 of the second proposition is therefore bound to CAR2 and all instances are generated and added to the GSD. Let us assume them to be (MOVE CAR2 3 12) and (MOVE CAR2 67 75). Each of these instantiations is tested for temporal compatibility in turn. In the traditional relational matching paradigm the values of the time variables are therefore tested for equality. As this fails OBJ2 is now bound to CAR3, all instantiations are generated and it is again tested for temporal compatibility. Note that for an instantiation (MOVE CAR3 10 70) this test would again fail although the instantiation (MOVE CAR1 15 65) exhibits that both objects move in a common interval.

The reason for the above failure is the lexical incompatibility of the values for the time variables T1 and T2. The role of the time variables as boundaries of a durative event is not considered. Without extensions the paradigm of relational matching cannot be used for our purposes.

We will now describe an extension to relational matching which allows to treat time variables correctly.

Two basic types of events must be distinguished, durative and non-durative events. An event which is valid in the interval (T1 T2) is durative if it is also valid in each subinterval (T3 T4) with $T1 \leq T3 < T4 \leq T2$. In our system all primitive events (propositions) are durative whereas certain composite events like OVERTAKE are non-durative. A special kind of a non-durative event where one boundary is fixed, is a timepoint event, e.g. STOP. The fixed timepoint in this case is given by the first time where the object does not move. For durative events the match between the pattern (MOVE CAR3 15 65) and a date (MOVE CAR3 10 70) should succeed as the time interval of the pattern is included in the interval of the GSD entry. This implies that time variables should not be instantiated but rather be restricted in their possible values. Hence the match should lead to the inequality: $15 \leq T1 < T2 \leq 65$.

The time variables in a proposition have to be interpreted as boundaries of the interval in which the proposition is valid. A single match of a model against a GSD entry leads to a restriction of the possible values of the time variables which can be written as an inequality. Further matches with the same and also newly introduced time variables lead to a system of linear inequalities. If this system has a feasible solution the propositions are temporally compatible. In [7] it is proposed to use the SIMPLEX algorithm of linear programming to find such feasible solutions. We propose a simpler algorithm which also accounts for durative and non-durative events.

In the implementation of the event recognition scheme each time variable has associated with it a minimum and a maximum value. When starting the recognition procedure these values are initialized to the beginning and end of the scene. Furthermore, each variable carries two lists, one containing all the variables which are greater ("upper variables") and the other one containing all the variables which are smaller ("lower variables").

For durative events the time variables T1 and T2 may be interpreted as minimum and maximum of the interval in which the proposition is valid. Hence we use for instantiations the notation: (<durative event>...<min T1><max T2>) e.g. (MOVE CAR1 15 65).

For non-durative events more than two time variables are necessary (see e.g. event model OVERTAKE) and the interval boundaries T1 and T2 lie within certain boundaries themselves. Therefore we use for instantiations the notation (<non-durative event>.. ..(<min T1><max T1>) (<min T2><max T2>)) e.g. (OVERTAKE CAR1 CAR2 (1 13) (15 20)).

A special kind of non-durative events are inchoative and resultative events like START and STOP where one interval boundary is fixed i.e. has equal minimum and maximum values. Therefore we use for instantiations the notations (<inchoative event> ...T1 (<min T2> <max T2>)) and (<resultative event> ...(<min T1><max T1>) T2) e.g. (START CAR1 12 (13 20)).

Consider a match of the pattern (MOVE CAR3 T1 T2) with (MOVE CAR3 10 70). The notation of the instantiation implies that it is a durative event, therefore 10 is interpreted as the minimum value for T1 and 70 as the maximum value for T2. T1 has as upper variables (those being greater) T2 and no lower variables and T2 has as lower variables T1 and no upper variables. According to the type of the event, durative, the procedure TIMETEST propagates the minimum value of T1 upwards and the maximum value of T2 downwards according to the following algorithm (due to Neumann):

1. add T2 to the set of upper variables of T1 and T1 to the set of lower variables of T2;

2. if the minimum (maximum) of T1 (T2) is greater (less) than its present maximum (minimum), the propositions are incompatible;

3. if the minimum (maximum) of T1 (T2) is less (greater) or equal to its present minimum (maximum), retain the present minimum (maximum);

4. if the minimum (maximum) is greater (less) than the present minimum (maximum) but smaller (greater) or equal than the

present maximum (minimum) use it as the new minimum (maximum);

5. do the above steps for all upper (lower) variables of T1 (T2) with the new present minimum +1 (maximum -1); if it fails for one variable, the propositions are incompatible.

The words in parentheses are for propagating the maximum. The addition (subtraction) of one in step 5 above ensures T1 < T2. In the beginning, the present minimum (maximum) are the initialized values (see above). Note, that according to the overall control structure all entries are subject to subsequent backtracking.

After running the above procedure for the example we have established as minimum and maximum 10 and 69 for T1 and 11 and 70 for T2. Consider the next match of the pattern (MOVE CAR1 T1 T2) against (MOVE CAR1 15 65). It is easily verified, that using the above mechanism the new values for the minimum of T1 and the maximum of T2 will be 15 and 65. In comparison, consider the next match to be between the pattern (MOVE CAR1 T1 T2) and the GSD entry (MOVE CAR1 1 9). The algorithm shows the instances to be incompatible and backtracking ensues.

We will now give an example of a non-durative event, namely STOP. The event model STOP is a composite one consisting of the primitives MOVE and STAND:

```
    (STOP OBJ T1 T2)
      (MOVE OBJ T1 T2)
      (STAND OBJ T2 T3).
```

When the system has to recognize all STOP events it first initializes the minimum and maximum values for the time variables T1, T2 and T3 to the first and last image of the sequence (e.g. 1 and 50). Then the generation phase starts.

Consider the GSD to contain the following entries after generating all instantiations of the first proposition: (MOVE CAR1 38 45) and (MOVE CAR1 29 34).

The first instantiation (MOVE CAR1 38 45) is then taken and according to the above algorithm minimum and maximum values for T1 and T2 are established of 38 and 44 for T1 and 39 and 45 for T2. The (STAND CAR1 T1 T2) instantiations are then generated. Assume the only instantiation to be (STAND CAR1 34 38). Due to the algorithm of TIMETEST, 34 is interpreted as minimum value of T2 and therefore not compatible to the already existing value of T2. Backtracking ensues and the second MOVE instantiation is taken establishing 29 and 33 as minimum and maximum values for T1 and 30 and 34 for T2. It can easily be verified that the above STAND instantiation is compatible and (STOP CAR1 (29 33) 34) is entered into the GSD.

The difference between non-durative events on the one hand and durative events on the other hand is that in the first case a match leads to the propagation of the minimum and maximum value of each time variable according to the above scheme. For durative events only the minimum of T1 and the maximum of T2 are propagated upwards respectively downwards.

Conclusion

A relational matching strategy for the recognition of temporal events has been described. An extension to the traditional relational matching scheme was introduced which handles all temporal relations between events correctly. Especially it accounts for the role of durative and non-durative events. The overall control structure of the recognition process is backtracking.

Among other work on temporal relations and temporal logic (cf. [7], [8],[9], [10], [11], see also the bibliography given in [12]), ALLEN 81, just to cite one, as well notes the difference between durative and non-durative events. He is mainly concerned with keeping the temporal order of events when the present (now) is changing and therefore proposes an interval based representation.

Most of the above mentioned approaches deal either with the ontology of time or the question whether time should be represented using time points or intervals, or constructing a history of events or with the question which temporal inferences can be made from natural language input. In the NAOS system we pursue the question how to recognize temporal events given a representation of real-world scenes and models of the events. This lead to the use of relational matching and the described extension.

The recognition scheme is implemented in LISP/FUZZY on a DECsystem 10 at the Fachbereich Informatik in Hamburg and is running under TOPS 10.

Acknowledgements

This work was done in the NAOS project which is partially supported by the Deutsche Forschungsgemeinschaft. I would also like to thank B. Neumann whose comments improved the paper. I am responsible for residual confusion.

References

[1] Barrow,H.G., Popplestone,R.J., Relational Descriptions in Picture Processing. In: Meltzer, D., Michie, D. (eds.), Machine Intelligence 6, University Press, Edinburgh, 1971, 377-396

[2] Barrow, H.G., Ambler, A.P., Burstall, R.M., Some Techniques for Recognizing Structures in Pictures. In: Aggarwal, J.K., Duda, R.O., Rosenfeld, A. (eds.), Computer Methods in Image Analysis, IEEE Press, 1977, 397-425

[3] Neumann, B., Novak, H.-J., Event Models for Recognition and Natural Language Description of Events in Real-World Image Sequences. IJCAI-83, 724-726

[4] Novak, H.-J., On Verbalizing Real-World Events: An Interface of Natural Language and Vision. In: Neumann, B. (ed.), GWAI-83, Informatik Fachberichte 76, Springer, Berlin/Heidelberg/New York, 100-107

[5] Dreschler, L., Nagel, H.-H., Volumetric Model and 3D-Trajectory of a Moving Car Derived from Monocular TV-Frame Sequences of a Street Scene. In: Computer Vision, Graphics and Image Processing 20 (1982), 199-228

[6] Neumann, B., Towards Natural Language Description of Real-World Image Sequences. GI-12. Jahrestagung, Informatik Fachberichte 57, Springer, Berlin/Heidelberg/New York, 1982, 349-358

[7] Malik, J., Binford, T.O., Representation of Time and Sequences of Events. In: Proc. of a Workshop on Image Understanding, Palo Alto, California, September 15-16, 1982

[8] Bruce, B.C., A Model for Temporal Reference and its Application in a Question Answering System. Artificial Intelligence 3 (1972), 1-25

[9] Kahn, K.M., Mechanisation of Temporal Knowledge. Technical Report MAC-TR-155 (September 1975) AI-LAB, MIT, Cambridge/MA, 1975

[10] Allen, J.F., An Interval-Based Representation of Temporal Knowledge. IJCAI-81, 221-226

[11] McDermott, D., A Temporal Logic for Reasoning About Processes and Plans. Cognitive Science 6 (1982), 101-155

[12] Bolour, A., Anderson, T.L., Dekeyser, L.J., Wong, H.K.T., The Role of Time in Information Processing: A Survey. In: SIGART (May 1982)