MITTEILUNG NR. 125

NATURAL LANGUAGE ACCESS TO IMAGE SEQUENCES: EVENT RECOGNITION AND VERBALIZATION

BERND NEUMANN

IFI-HH-M 125/84

SEPTEMBER 1984

FACHBEREICH INFORMATIK
UNIVERSITÄT HAMBURG
SCHLÜTERSTR. 70
D-2000 HAMBURG 13

This report is based on a contribution to the First International Conference on Artificial Intelligence Applications, Denver, Colorado, December 5-7, 1984

Zusammenfassung

Der Beitrag behandelt das Problem, auf den Inhalt von Bildfolgen zuzugreifen, insbesondere Aspekte der höheren Bilddeutung und des natürlichsprachlichen Zugriffs auf bewegte Vorgänge. Es werden Verfahren vorgestellt, die im System NAOS für TV-Bildfolgen von Verkehrsszenen implementiert sind. Eine natürlichsprachliche Anfrage über eine Objektbewegung, z.B. einen Überholvorgang, wird in eine Tiefenstruktur überführt, in der die Tiefenkasus des Verbs spezifiziert sind. Hieraus werden Prädikate abgeleitet, die auf einer propositionalen Szenenbeschreibung instantiiert werden müssen. Die auf diese Weise realisierte Ereigniserkennung stellt einen Prozeß der höheren Bildeutung dar, der sich von bisher bekannten Verfahren insbesondere aufgrund der Behandlung von Zeitintervallen unterscheidet. Das domänenspezifische Wissen ist in transparenten Datenstrukturen codiert und kann an andere Aufgabenstellungen angepaßt werden.

NATURAL LANGUAGE ACCESS TO IMAGE SEQUENCES: EVENT RECOGNITION AND VERBALIZATION

Bernd Neumann

Fachbereich Informatik, Universität Hamburg
Schlüterstr. 70, D-2000 Hamburg 13, FRG

ABSTRACT

This paper addresses the problem of obtaining and accessing high-level interpretations of temporal image sequences. It presents an approach which has been implemented in the system NAOS for the domain of traffic scenes. Natural language questions pertaining to object motions are transformed into a deep case frame centered around the verb of locomotion. The deep case structure is mapped into predicates involving 'event models' which capture the verb meaning. Event recognition is performed by a hierarchical matching scheme (BARROW et al. 72) using constraint satisfaction for time intervals. The methods can be applied to other tasks by adapting the domain knowledge which is represented in terms of transparent data structures.

1. INTRODUCTION

As progress in computer vision paves the way for applications of increasing complexity, the problem of accessing computer vision results becomes important. This is not yet the case for most existing vision systems where the output may be as simple as 'pass' or 'fail' or is given in terms of recognized objects and their positions. In the last years, however, problems of increasing complexity have been tackled where the results may not be represented in such a simple manner. In this contribution we shall consider the analysis of time-varying imagery which may be required for various applications, e.g. determining cloud motion or environmental changes, tracking moving objects and computing their 3D shape and 3D

trajectories, classifying heart motion, etc. Time-varying imagery is usually represented in terms of an image sequence with images taken at regularly spaced time intervals. The results of motion analysis could in priciple be given by shape and position measurements for each instance of time. But this is clearly insufficient for most purposes because of two reasons. First, it is difficult if not impossible for humans to extract useful information from vision system output of this form, particularly if the image sequence consists of hundreds of images, and second, the concepts which one is really interested in - e.g. particular motion trajectories - are not recognized by the vision system. Hence we have to solve two problems: extend the scope of a vision system to high level concepts and provide means to communicate results to human users.

This paper addresses both issues raised above. It presents methods for recognizing certain high-level concepts called 'events' and explores a particularly convenient way of accessing the vision system output: natural language. Image sequences are viewed as data in a large database. Natural language queries provide database access. Examples are taken from the domain of street traffic. A typical retrieval question is

"Did a yellow VW overtake a truck in front of the FBI?".

It will be shown that the deep case structure of this question can be mapped into predicates involving scene components which in turn can be computed from the underlying image sequence. Question answering amounts to instantiating such predicates. This contribution extends previous work on natural language access to pictorial databases (HUSSMANN and SCHEFE 84, WAHLSTER et al. 83) in several respects.

(i) High-level concepts of interest are defined as transparent data structures in an extendable knowledge base.

(ii) Effective techniques are provided for recognizing events which are conjunctions of predicates extending over time.

(iii) The deep case structure of natural language questions can be used to control event recognition and retrieve events.

The processes which will be discussed have been implemented in a system called NAOS. On the vision side NAOS expects that object recognition has been achieved and the 3D scene geometry is available, thus low-level vision problems are by-passed. On the language side NAOS connects to a parser and a generator developed for the project HAM-ANS (HOEPPNER et al. 83).

2. EVENT RECOGNITION

In this section we shall show how to recognize interesting occurrences in an image sequence. It is assumed that the image sequence has been analyzed up to an intermediate level where the 3D locations of all objects of interest are known for each instance of time. Also class membership, color and shape features have been determined. The data are represented in a relational format as shown below.

```
(CLASS VW1 VW)
(COLOR VW1 YELLOW)
(LOCATION VW1 (20 100 8) (4 1 0) 1)
(LOCATION VW1 (40 105 8) (4 1 0) 2)
(LOCATION VW1 (60 110 8) (1 0 0) 3)
etc.
```

Each relational tupel is a proposition about the scene, hence relations can also be viewed as predicates. The arguments of the LOCATION predicate are object ID, 3D position and orientation vector, and instance of time. Because of the prevailing geometrical contents of the data this level of representation is termed 'geometrical scene description' (GSD).

We are interested in recognizing specific patterns of motion, for example one car overtaking another. In general, such a pattern - called an 'event' - may be any subspace of four-dimensional space-time. Events can be defined to meet a particular purpose. In TSOTSOS 80, for example, events describe leftventricular heart motion. In NAOS events are occurrences in a traffic scene which can be described by verbs of locomotion.

Events are organized into classes according to the verb which is associated

with the event. Event classes are defined by event models. An event model is the conceptual entity which specifies what we are looking for in a scene.

The notions of event models and events are analogous to object models and objects or other established concepts in knowledge representation. Event models are generic descriptions which are usually part of the knowledge base. Events are instantiations thereof and refer to a particular scene. In the following we shall present the event models used in NAOS in more detail.

Event models consist of a head, which is a predicate about a scene, and a body, which specifies how to verify the predicate. The following is the model for 'overtake' events.

```
Head:   (OVERTAKE OBJ1 OBJ2 T1 T2)

Body:   (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 T3 T4)
        (RECEDE OBJ1 OBJ2 T4 T2)
```

Event model predicates are written in the same relational notation as the input data, except that the arguments are usually variables which must be instantiated by matching the event model to the data.

The variables T1 to T4 are time variables denoting interval boundaries. Time intervals are different from other data in that they are represented by constraints rather than fixed instances. Constraints arise from durative predicates such as MOVE. Durativity means that a predicate, if true for a certain interval, is also true for all subintervals:

$$(P \ldots T1\ T2) \Rightarrow (P \ldots T1'\ T2') \text{ for } T1 \leqslant T1' < T2' \leqslant T2$$

If a durative predicate is matched against suitable data, the time variables are constrained according to this inequality rather than instantiated to particular values. As time variables typically occur in more than one predicate of an event model, constraints accumulate. Hence the need to perform feasability tests and to compute solutions of the resulting system of linear inequalities arises.

In MALIK and BINFORD 82 linear programming, in particular the SIMPLEX method, is proposed to obtain the desired results for a similar problem. In NAOS, a much simpler procedure is employed. It is based on an inequality net which is maintained for all time variables. Each variable has a current minimum and maximum value and is linked to other variables according to the inequalities. If a new inequality is encountered, new links are added, and the new constraints are propagated along the links, lower bounds upwards and upper bounds downwards. Whenever a minimum surpasses a maximum, the inequalities are inconsistent. Otherwise minimum and maximum are valid bounds and provide the desired solution.

For the following example we shall assume that all 'move' events have been computed in an initialization step and entered into the database. (This is the usual procedure in NAOS.) Consider the data

```
(MOVE CAR1  1 30)
(MOVE CAR2  7 13)
(MOVE CAR2 20 35)
(BEHIND CAR1 CAR2 15 27)
```

and the list of predicates (taken from the event model 'overtake')

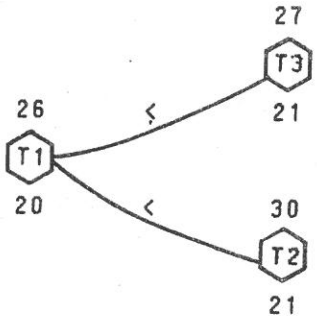
```
(MOVE OBJ1 T1 T2)
(MOVE OBJ2 T1 T2)
(BEHIND OBJ1 OBJ2 T1 T3)
```

One possible instantiation would give rise to the inequalities:

$$1 \leq T1 < T2 \leq 30$$
$$20 \leq T1 < T2 \leq 35$$

$15 \leq T1 < T3 \leq 27$

The corresponding inequality net exhibits the resulting minimal and maximal values as shown below.



We believe that this incremental constraint propagation method is a key element for dealing with temporal concepts in scenes. It reflects the fact that the basic building blocks of interesting concepts are scene properties extending over some time interval, i.e. durative properties. Taken together, they give rise to systems of inequalities as shown above, and to concepts which need not be durative, e.g. 'overtake' or 'stop', but whose interval boundaries also obey constraints.

We now describe the event recognition strategy used in NAOS. It is assumed that a certain set of predicates has to be instantiated, e.g. the body of the event model 'overtake'.

The basic techniques for event recognition are hierarchical matching and backtracking search. The scheme used in NAOS is particular in several ways as will become apparent. The key process is EVENTEVAL which tries to instantiate a set of predicates with the goal of making all predicates conjunctively true.

EVENTEVAL set of predicates:

- SELECT predicate from set.
- GENERATE all instances.
- Select instance and TEST for compatability.
- Backtrack if not compatible, else

- EVENTEVAL remaining predicates.

The following steps are carried out in the GENERATE component:

GENERATE all instances of a predicate:

- Generate all instances of non-instantiated arguments except time variables.
  Each combination of such instances defines a predicate 'pattern'.
- Skip predicate pattern if generated before.
- EVENTEVAL body if predicate is composite, else EVAL body.

GENERATE cycles through all patterns of a predicate by substituting possible instances for non-instantiated variables except for those denoting time intervals. There are provisions for avoiding duplicate computations by keeping a history of all patterns which have been tried before. Evaluation is either carried out by a recursive call of EVENTEVAL or by EVAL which deals with primitive predicates. Primitive predicates (as opposed to composites) cannot be broken down into constituents, they are defined as procedures. Each evaluation of a pattern generates all time intervals for which the pattern is true. EVAL can be characterized as follows.

EVAL primitive predicate:

- Compute all maximal time intervals for which the predicate is true.
- Enter instances into database.

The computations of EVAL are carried out using data of the GSD or facts of the knowledge base. In its simplest form the computation of a primitive predicate is a direct retrieval from the GSD (e.g. CLASS or COLOR). From the structure of EVENTEVAL and GENERATE one can see that event recognition proceeds in a doubly recursive manner: by recursively instantiating a list of predicates and by recursively decomposing predicates.

3. NATURAL LANGUAGE ACCESS

So far we have discussed processes and representations which can be considered integral parts of a vision system. Data provided by a fictitious vision front-end has been interpreted in terms of high-level conceptual units called events. We now consider the task of accessing the data by natural language queries. The idea of treating an image sequence as a data-base and implementing natural language access has also been pursued in the HAM-ANS project (HOEPPNER et al. 83). This contribution presents processes which map from the deep case structure of a question into predicates of the event recognition formalism developed in NAOS. The parser and generator which connect deep case structures with surface strings are borrowed from HAM-ANS. Consider the question:

"Did a yellow VW overtake a truck in front of the FBI?"

The parser produces a deep case structure which specifies the linguistic representations for each of the verb's deep cases. For example, "a yellow VW" is the agent, "a truck" is the objective, and "in front of the FBI" is the locative. (FBI is the German abbreviation for Computer Science Department). What does the deep case structure express about a scene in terms of predicates computable from the GSD? What does it mean to be "in front of" some object while overtaking?

We call the relevant body of knowledge which connects linguistic case fillers with the corresponding scene-oriented (i.e. geometrical) notions 'deep case semantics'. This knowledge is represented by case frame models as illustrated below for the example 'overtake'.

Case frame model 'overtake':

```
(VERB "overtake")
(OVERTAKE OBJ1 OBJ2 T1 T2)

(AGENT AGT-EXP)
(REF AGT-EXP OBJ1)
```

(OBJECTIVE OBJ-EXP)
(REF OBJ-EXP OBJ2)

(LOCATIVE LOC-EXP)
(LOC-REF LOC-EXP (LOC-PREP OBJ1 LOC-OBJ T1 T2))

(TENSE TNS-EXP)
(TIME-REF TNS-EXP T1 T2)

Each case description of the case frame model consists of two parts: a declaration of an identifier (or constant in case of the verb) for the case expression on the language side, and a predicate (in general a list of predicates) relating the case expression to the scene data. For each case expression occurring in a question the corresponding predicates have to be verified in the scene. The heart of the deep case semantics are the predicates REF and LOC-REF. They will now be described in more detail.

REF relates a natural language expression for a noun phrase or pronoun (in the format understood by the surface string generator or delivered by the parser) to possible candidates in the scene. This step is called dereferentiation. For example, a set of yellow VWs {VW1, VW3, ... } may be determined as the range of OBJ1 (referring to the case frame model 'overtake' and a fictitious scene with more than one yellow VW). If no such objects exist, an answer such as "There is no yellow VW in the scene" will be generated. In general, the second step is a quantization test on referents. Consider the question

"Did the BMW overtake two trucks?".

If the BMW has not been previously mentioned, the definite article implies that there is exactly one BMW in the scene. Also, there must be at least two trucks. If the quantization test fails, an appropriate answer will be generated.

LOC-REF is analogous to REF with the difference that an abstract location instead of an object is to be related to a NL expression. The locative case and other spatial deep cases such as source, path and goal are often

misconstrued as referring to names of places or objects. With scene data as a referential data base the spatial deep cases can be defined concisely as follows. The locative case is the union of all positions of the agent during the event, i.e. the volume swept out by the agent's trajectory. Similarly, source and goal are spatial volumes corresponding to the object's initial and final location.

While dereferentiation of objects is performed before event recognition, all other constraints of a question are evaluated in connection with the event model corresponding to the verb of the question. This is now shown for the first of the two examples above. The deep case semantics of the verb, of course, are given by (OVERTAKE OBJ1 OBJ2 T1 T2) and the corresponding event model. From the locative expression "in front of the FBI" the appropriate scene constraint (IN-FRONT-OF OBJ1 LOC-OBJ T1 T2) is generated by calling LOC-REF, while BUILDING1 (which is the FBI) is bound to LOC-OBJ. TIME-REF generates constraints for the interval boundaries. From the past tense of the question one gets

$$T\text{-}PAST\text{-}BEG \leq T1 < T2 \leq T\text{-}PAST\text{-}END$$

where the boundary values are fixed time marks. This initializes the temporal constraint satisfaction scheme described earlier. Hence, in summary, the following query predicates are obtained:

(OVERTAKE OBJ1 OBJ2 T1 T2)
(IN-FRONT-OF OBJ1 LOC-OBJ T1 T2)

with suitable ranges attached to the variables.

Event recognition now takes place using EVENTEVAL. The result is a list of all possible instantiations of the query predicates. Time variables are constrained to certain ranges, all other variables are instantiated with fixed tokens.

The next step is another quantization test, in this case on the number of events. NAOS can handle quantizations of considerable complexity, e.g.

"Did three pedestrians cross at least two streets?"

This is evaluated by counting the number of pedestrians and crossed streets in the crossing events returned by EVENTEVAL. Each pedestrian must cross at least two streets.

For answer generation, the instantiated events are used to fill a case frame using the deep case semantics and the case fillers available from the question. A surface string is generated by passing the deep case frame to a generator written by BUSEMANN 84. In the current state of implementation, only simple answers can be given, e.g.

"Yes, a yellow VW overtook a truck in front of the FBI."

More advanced techniques, e.g. cooperative answer generation (WAHLSTER et al. 83) can be added using the same framework.

4. CONCLUSIONS

This contribution has addressed the problem of information retrieval from temporal image sequences. It has been assumed that images have been processed up to the level of recognized objects. This is the intended scope of many vision systems currently under development. In general, this level of representation is inadequate for describing the contents of time-varying scenes as it is neither suitable for human interpretation nor complete in the sense that interesting temporal concepts are recognized. The system NAOS which has been described offers both: recognition of high-level concepts and user access via natural language queries. The system has been applied to data describing a traffic scene. The events which can be retrieved are object motions corresponding to natural language questions involving verbs of locomotion. About 50 verbs are currently considered in NAOS. Not all of them can be implemented in the way described in this paper, as there are some which refer to more than the scene geometry, for example 'yield' or 'wait'. To recognize such events, it is necessary to generate expectations about the development of the scene in a larger context. NAOS is currently extended accordingly.

ACKNOWLEDGEMENTS

Hans-Joachim Novak has contributed to project NAOS as the main collaborator of the author. The project is partially supported by the Deutsche Forschungsgemeinschaft.

REFERENCES

Barrow et al. 72
Some Techniques for Recognizing Structures in Pictures
H.G. Barrow, A.P. Ambler, and R.M. Burstall
in: J.K. Aggarwal, R.O. Duda, and A. Rosenfeld (eds.), Computer Methods in Image Analysis, IEEE Press, 1977, 397-425

Busemann 84
Surface Transformations During the Generation of Written German Sentences
S. Busemann
Report ANS-27, Research Unit for Information Science and AI, Hamburg, 1984

Hoeppner et al. 83
Beyond Domain Independence: Experience with the Development of a German Language Access System to Highly Diverse Background Systems
W. Hoeppner, T. Christaller, H. Marburger, K. Morik, B. Nebel, M. O'Leary, W. Wahlster
IJCAI-83, 1983, 588-594

Hussmann and Schefe 84
The Design of SWYSS, a Dialogue System for Scene Analysis
M. Hussmann, P. Schefe
in: L. Bolc (ed.), Natural Language Communication with Pictorial Information Systems, Springer, 1984, 143-202

Malik and Binford 82
Representation of Time and Sequences of Events
Jitendra Malik and Thomas O. Binford
in Proc. of a Workshop on Image Understanding, Palo Alto, California, September 15-16, 1982

Tsotsos 80
A Framework for Visual Motion Understanding
J.K.Tsotsos
TR CSRG-114, University of Toronto, 1980

Wahlster et al. 83
Over-Answering Yes-No Questions: Extended Responses in a NL Interface to a Vision System
W. Wahlster, H. Marburger, A. Jameson, S. Busemann
IJCAI-83, 1983, 643-646