Semiorder Database for Complex Activity Recognition in Multi-Sensory Environments

Shailendra Bhonsle, Amarnath Gupta, Simone Santini, Ramesh Jain
Visual Computing Lab., Dept. of Electrical and Computer Engineering,
University of California, San Diego
E-mail: bhonsle@cs.ucsd.edu, gupta@sdsc.edu, {ssantini, jain}@ece.ucsd.edu

Abstract

A prototype semiorder database used for activity recognition in multi-sensory monitoring environments is described. Activities are spatio-temporal compositions of events, which are a type of atomic semantic units for such compositions. The focus is on the temporal composition of activities from events in the presence of bounded duration of temporal uncertainty in an event occurrence. Such temporal uncertainty forces the concurrency between event occurrences to be intransitive. Under certain assumptions, a subclass of partial orders, known as semiorders, models such intransitive concurrency appropriately. The semiorder database stores events and their semiorder temporal order of occurrences. A semiorder data model and the corresponding query language that embeds a semiorder pattern language are the main constituents of the semiorder database. We demonstrate this database and queries for activity recognition in a real time environment. The demonstration also includes a transducer subsystem for detection of events.

1. Introduction

The usefulness of databases in activity recognition in multi-sensory monitoring environments has lately been recognized [3]. A multi-sensory environment may consist of sensors as complex as multiple visual sensors. For example, multiple cameras may produce multi-perspective video of an observed environment, like a parking lot, or a segment of a highway. A visual signal processing subsystem detects and extracts information about different *domain-independent features* of observed objects. These objects may be detected mobile objects or spatial regions of interest. Some examples of domain-independent features are centroid, velocity, three-dimensional bounding box, and color of detected mobile objects.

One of the goals of such environments is to detect *activities* of interest that occur in the environment. An activity is defined as a complex fusion of multi-sensory observed data over *any* temporal and spatial extent. It is useful to consider an activity to be a complex spatio-temporal composition of events, which are a type of atomic semantic units for multi-sensory information produced in the environment. Events occur at specific points in time, and finite bounded intervals of temporal uncertainties are associated with their occurrences. Events are detected using specialized algorithms from the domain-independent feature information produced in the multi-sensory environment. For example, a car hitting a wall is an activity that can be decomposed into events of "car entering the region of the wall", "car stopping", and concurrent "occurrence of a loud noise". A database is used to store events, their spatial and other attributes, and also their temporal order. The usefulness of the database approach comes through the use of an appropriately defined query language. This makes it possible to *detect* and *retrieve* exponentially many user defined activities of interest on the same base set of events. In the following subsections, we briefly describe the semiorder data model, an algebraic query language that embeds a semiorder pattern definition language, and the design of the current semiorder database prototype.

2. Semiorder data model and query language

In our research, we primarily focused on database requirements for dealing with temporal aspects of composition of activities from events. Spatial parameters associated with events and activities are defined through appropriate use of *attributes* of events. Consider the binary relation *occurs_before* between occurrences of events A and B, and define two events to have occurred concurrently if they are not related by this relation. Association of temporal uncertainty intervals with occurrences of events forces the concurrency between event occurrences to be *intransitive*. Such intransitive concurrency between events is not modeled appropriately by *temporal sequence* data models and their *weak-order* extensions, which either do not cater for concurrency, or are capable of only dealing with transitive concurrency. Assumptions of a fixed uncertainty interval for events in a given set of events and that the probability of an event occurrence is uniformly distributed over this interval give rise to a special subclass of partial orders known as

0-7695-0506-6/00 $10.00 © 2000 IEEE

semiorders. Semiorders provide the right level of abstraction to model intransitive concurrency between event occurrences, and they represent the natural evolution of temporal sequence based compositions of activities from events in the presence of temporal event occurrence uncertainties. The model and its assumptions are described in detail in [2].

In our database-centered approach, we adapted the notion of semiorders to design a *data model* for storage of events and their temporal order. The corresponding algebraic semiorder query language provides useful operations on sets of semiorders that populate an instance of a semiorder database. Furthermore, this language embeds a semiorder pattern definition language to define semiorder constants for use in conjunction with *morphism* class of operations of the query language. The data model, the query algebra, and the semiorder pattern definition language are described in detail in [2, 1].

Apart from providing semiorder specific operations (like *disjoint union (concurrence), rearrange,* and *flatten*), and relational operators on unordered (flattened) sets of events, the language provides *morphism, iteration,* and *aggregation* classes of semiorder operations. All these operations return sets of semiorders. Morphism operators, for example suborder isomorphism, are used to extract an activity from the database. The restricted set of iteration operators in the currently defined language provides means to iterate over semiorders, for example to find the set of nodes at the first level of a semiorder. Aggregation operators include "scalar" and "ordered" aggregation. Ordered aggregation operators return results which are semiorders. Morphism operators work in conjunction with semiorder patterns, which are succinct representations of sets of semiorder constants, and are defined in a semiorder pattern definition language [2].

3. Semiorder database system design

The overall system architecture comprises of a visual signal processing subsystem, a transducer subsystem for detection of events, and the semiorder database subsystem for activity recognition. The database prototype is generic and can be used in other domains, for example in retrieval of appropriate visual information in visual information management systems. The system is implemented using a commercially available Java based object-oriented database [4]. The semiorder database prototype is implemented in Java and works across different platforms. The schema definition as well as events belonging to many *semiorder schemas* are stored in the native object-oriented database. The schema is defined using a schema definition language, and is parsed to automatically generate needed schema and event classes.

The query environment consists of a query parser, a validator, an optimizer, an execution engine, and a query result visualization engine. The validator extracts the schema and validates the query against the schema definition. The current optimizer only decides *the set of data to be extracted from the native database.* It arranges *select* conditions in query expressions so that an optimal set of event data is extracted from the native database. The execution engine applies appropriate semiorder algebra operators on this set of extracted data. The returned results always consist of sets of semiorders, and are visualized and navigated through using semiorder graphs. Figure 2.1 shows a screenshot of the query result visualization engine. We have a separate video database in our domain of application. The semiorder database query environment provides for retrieval of video sequences for recognized activities.

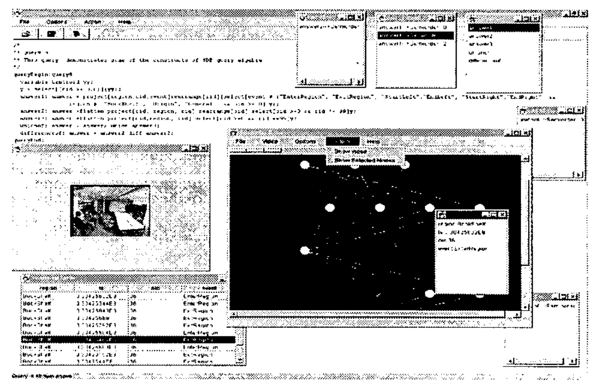

Figure 2.1: The query environment for the semiorder database

4. Conclusions

We demonstrate the current prototype implementation of the semiorder database during this conference,. The setup consists of a real-time signal processing module, event detection transducer, and the semiorder database. The monitored environment consists of a segment of a highway, whose video is played in real-time through a video recorder. Interesting activity queries include various traffic patterns involving single and multiple objects. The video segments corresponding to these activities are retrieved from another real-time video database.

5. References

[1]. S. Bhonsle et. al., "Complex visual activity recognition using a temporally ordered database", 3rd International Conference on Visual Information Systems, Amsterdam, June 1999.

[2]. S. Bhonsle, "Semiorder model for temporal composition of activities from events in multi-sensory information management systems", Ph.D. dissertation, Department of computer science and engineering, University of California San Diego, under preparation, December 1999.

[3]. P. Kelly et. al., "An architecture for multiple perspective interactive video", ACM Multimedia 95, pp. 201-212, 1995.

[4]. ObjectStore Inc., "PSE-Pro for Java users manual", 1999.