A Review: Recognizing and Learning Events in Cognitive Vision Systems

Somboon Hongeng

Cognitive Systems Laboratory (KOGS)

Dept. of Computer Science

Hamburg University

D-22527 Hamburg, Germany

hongeng@kogs.informatik.uni-hamburg.de

November 2004

Zusammenfassung

Dieser Bericht enthält eine Übersicht über verschiedene methodische Ansätze zum Erkennen und Lernen von Ereignissen in Video-Filmen. Die Ansätze zur Ereigniserkennung werden grob in 1) statistische Mustererkennung und 2) symbolisches Schließen gegliedert. In einem statistischen Ansatz werden die Arten von Ereignismustern, die erkannt werden können, durch die jeweiligen probabilistischen Modelle bestimmt und können nicht leicht verallgemeinert werden. Ein logikbasierender Rahmen für symbolisches Schließen bietet dagegen einen allgemeineren und systematischeren Ansatz für die Repräsentation und Verwaltung einer umfangreichen Ereigniswissensbasis. Hier scheint ein Aggregat ein natürliches und nützliches Konstrukt zur Repräsentation höherer Konzepte wie Objektkonfigurationen, Vorgänge und Ereignisse zu sein. Bei einer Szeneninterpretation können taxonomische und kompositionelle Beziehungen zwischen Aggregatkonzepten ausgenutzt werden, wobei visuelle Evidenz und Kontextinformation einbezogen werden.

Konzepte für primitive Vorgänge, die die visuelle Evidenz für eine symbolische Szeneninterpretation darstellen, werden häufig durch Vektorquantisierung (VQ) gelernt. Bei VQ wird der Raum kontinuierlicher Objektmerkmale in eine endliche und kleine Zahl relevanter Prototypen diskretisiert. Alternativ kann ein Hidden-Markov-Model (HMM) verwendet werden, um zeitlich zusammenhängende qualitative Primitive zu repräsentieren und zu entdecken. Eine häufig beobachtete Folge von Aktionen kann mit einem HMM ebenfalls gelernt werden. Allgemeinere zeitliche Muster können mit einer Beschreibungssprache definiert werden, die Reihenfolgebeschränkungen zwischen Zeitpunkten oder Intervallen ausdrückt. Das Erlernen temporaler Muster wird häufig durch eine Suche vom Allgemeinen zum Speziellen realisiert. Logikbasierte Induktion (z.B. ILP- oder AMA-Algorithmen) bieten offenbar eine generische Lösung für das Problem, hierarchische Ereignismuster zu erlernen. Ansätze dieser Art kann man im Hinblick auf die Ausdrucksfähigkeit der verwendeten Beschreibungssprache und die Komplexität des Induktionsprozesses differenzieren.

Abstract

This paper reviews several computational frameworks for recognizing and learning events from a video stream. Approaches to event recognition are largely classified into 1) statistical pattern recognition, and 2) symbolic reasoning. In a statistical approach, the types of event patterns that can be recognized are governed by the choice of probabilistic models, and are difficult to generalize. A logic-based framework in symbolic reasoning provides a more general and systematic approach for representing and maintaining a large knowledge base of events. In symbolic reasoning, an aggregate seems to be a natural and useful construct for representing high-level concepts such as object configurations, occurrences and events. Scene interpretation can exploit the taxonomical and compositional relations between aggregate concepts while incorportating visual evidence and contextual information.

Primitive occurrences (which provide visual evidence for scene interpretation) are often learned by Vector Quantization (VQ). In VQ, the continuous space of object-level features is discretized into a finite and small number of relevant prototypes. Alternatively, a hidden Markov model (HMM) may be used to detect the coherency in qualitative primitives over a time interval. A commonly observed consecutive sequence of actions can also be learned by HMMs. Other temporal patterns can be expressed by a language that places ordering constraints on either the time-points or the intervals of events. Temporal pattern mining is often realized by a general-to-specific search technique. A logic-based induction (e.g., ILP and AMA-based algorithms) seems to provide a generic solution to the learning of hierarchical event models. These techniques are differentiated by the expressiveness of the languages used for representing the events and the complexity of inductive process.

1 Introduction

Recognizing and learning events taking place in a video stream are of key importance in cognitive vision systems. Events to be recognized may be simple motions of an agent acting alone (e.g., hand signing), or an agent acting upon an object (e.g., a hand picking up an object). Events can also be very complex, involving many agents (e.g., setting up plates and cutlery on a dining table). It is natural for humans to describe these events in terms of semantic concepts with regard to the types of agents, how they move (or how they are moved) in relation with some other agents. To enable an interaction with humans, a cognitive vision system must bridge the gap between the input video signal data and the event concepts. This is a particularly difficult task because the visual evidence is not always complete (e.g., in an evolving scene or in a partially visible scene) and the mapping between evidence and and event concepts is not one-to-one.

Approaches to event recognition in the last two decades can be largely classified into 1) statistical pattern recognition and 2) symbolic reasoning. In a statistical approach, it is assumed that there are underlying probabilistic models that generate visual patterns. The types of event patterns that can be recognized are often governed by the chosen probabilistic models. For example, a hidden Markov model (HMM) is a state-based random process that can be trained to recognize sequences of simple actions [16]. A HMM consists of a set of event states and probabilistic transitions among the states. Each state is assumed to uniquely generate a characteristic distribution of motion features. Therefore, it is less effective to train a HMM with composite event states, where the motion features may vary in an unpredictable way (which is common in an interaction of multiple agents). Some adaptations of the HMMs (e.g., a coupled HMM [14]) have been proposed recently, where an interaction of up to three agents can be modeled. It is often difficult to train these models because the parameter space becomes prohibitively large very quickly. Generalizing and reusing parts of the HMM models are also difficult.

In symbolic reasoning, a logic-based language is used to represent knowledge and background facts in a domain of interest. Non-logical symbols can then be interpreted based on deduction and entailment. Applied to dynamic scene understanding, a knowledge base of events must be constructed. Some expressive languages often allow events to be modeled in a hierarchical fashion, easily understood by human users. For example, simple events may be propositions on spatial-temporal characteristics of

motions of an agent. A more complex event may be composed of axioms that relate some spatial-temporal properties of these simple events. Given partial evidence, scene interpretation is performed by making hypotheses about potential models and invalidating hypotheses of unlikely events. Since event models are highly structured, deductions and entailments can be done efficiently.

One disadvantage of the symbolic reasoning approach is that event models must be carefully handcrafted. For example, one needs to write all axioms relating non-logical symbols to make sure that the proposition one wants to be true are entailed by and consistent with the knowledge base. Another disadvantage is that it is assumed that simple events can be abstracted from visual scenes reliably. When this is not the case, the system requires a strategy to handle conflicting evidence. Also, the hypothesis space of possible events to be explored is normally large and there is need for a control mechanism that guides the search. Event likelihood (probability) can provide such guidance and there has been some studies on the unification of statistical pattern recognition and symbolic reasoning approaches.

In a unified approach, simple events are abstracted from object motion based on a statistical method and are associated with a likelihood degree of the matching. Unlike in a traditional statistical approach, complex events (e.g., interactions between objects) are modeled explicitly by a spatio-temporal logic. During scene interpretation, probabilities of simple events can be combined subject to the defined spatio-temporal relations and can be used to gauge the likelihood of the hypothesis being made about the complex events.

Regardless of the choices of event representation, it is difficult to predict and handcraft all possible event models. Event patterns may change over time and new patterns may arise. The parameters of the statistical models of simple events may need to be modified to adapt to a new context. Event relations defined for the hierarchical models of complex events may need to be revised. For example, the spatio-temporal constraints between events in a model may need to be tightened or augmented to include a new event. It is evident that a scene interpretation system requires a mechanism for learning and adapting event models. Research in machine learning has focused on both symbolic and statistical learning techniques in many application domains (e.g., computer vision, database), but there is little research effort in event learning for dynamic scene interpretation.

In this paper, we review the state-of-the-art in event recognition and

learning. Several modes of learning (e.g., supervised, unsupervised) devised for various event representations (e.g., statistical and logical representations) will be discussed. Section 2 describes the construction of event models and the scene interpretation process. The review of the learning methods developed for each preocessing level is in Section 3. We conclude our review with a discussion in Section 4.

2 Dynamic Scene Interpretation

We present in this section the development of a scene interpretation system for understanding table-top events. In particular, we use the "table laying" scene described by Neumann et al. [13] as a guiding scenario. We focus on the requirements of various processing components and representations, which are common to the systems developed for other domains.

In a "table laying" scene (Figure 1), one observes objects such as plates and knives being transported to and arranged on the table over a certain time interval. The spatial configuration of the arrangement is subject to the type of meal being served (e.g., dinner, breakfast). Also, there exist temporal constraints, e.g., a saucer is always transported before a cup.

Figure 1: Snapshot of a table-laying scene.

It is evident that humans use diverse knowledge beyond the visual observation to describe an evolving scene in qualitative terms. For example, one can describe not only the primitive occurrences (e.g., "plate-laying",

"saucer-laying"), but also the fact that they are part of the global event "dinner setting" and that there are other missing object-laying events (e.g., a fork will *later* be placed to the *left* of the plate).

To equip a cognitive system with such capabilities, it is apparent that the system must know a typical set of primitive occurrences and how they are abstracted from quantitative visual observations (possibly through a learning process). The cognitive system also needs to prescribe how a complex event concept (i.e., a scene model) is defined from these primitive occurrences. These scene models are often handcrafted by knowledge engineers, but can also be acquired through a learning process. Finally, the system must be equipped with a scene interpretation mechanism, where some of these conceptual models are hypothesized from the available visual evidence and used to reason about the missing events or occurrences.

In this section, we briefly describe the conceptual framework, in which a high-level scene interpretation is determined by constructing a description of the scene in terms of concepts provided in a conceptual knowledge base. The survey of the learning mechanisms for both primitive occurrences and the conceptual event models is presented in Section 3.

2.1 Conceptual Knowledge Base

A conceptual knowledge base (CKB) consists of scene models, which are conceptual entities for high-level scene interpretations. To define a scene model in a CKB, one must choose an appropriate representational formalism. In [13], "aggregates" are introduced as representational units for object configurations, occurrences, episodes and other concepts which occur in high-level interpretations. The structure of an aggregate has several properties that are found to be natural and useful for high-level scene interpretations. First, the structure is composed of parts with relations between the parts, giving rise to a partonomy which is the hierarchical structure induced by part-of relations. Second, it supports a subsumption hierarchy (taxonomy), where a model may be a specialization of another model (or the other way around). Third, constraints can be specified qualitatively and concretely within an aggregate, which is useful for modeling spatial temporal relations in model definitions.

An aggregate can be represented using a frame-based model which contains the following information: 1) concept name, 2) taxonomical parent concepts, 3) parts, and 4) constraints between parts. Figure 2 shows a frame that describes an occurrence of placing a cover on a table. In the

parts section, local names and concept memberships of the visual phenomena (i.e., a table-top, a cover configuration, and the transport occurrences of the objects that are parts of the cover) are tied together to form a concept and satisfying certain constraints, which are expressed in the constraints section of the frame. Furthermore, there are time marks which refer to the beginning (tb) and ending (te) of the place-cover occurrence. In the constraints section, there are identity constraints that relate constituents of different parts to each other (e.g., the object of a plate transport occurrence must be identical to the plate in the cover configuration). There are also qualitative constraints on the time marks associated with sub-occurrences. Spatial constraints are expressed by a cover configuration which is modeled by another aggregate, e.g., a cup must be placed on top of the saucer. This example shows that an aggregate may have other aggregates as parts. Hence, a compositional hierarchy is induced.

name:	pc-cover
parents:	:is-a agent-activity
parts:	pc-tt :is-a table-top
	pc-tp1 :is-a transport with (tp-obj :is-a plate)
	pc-tp2:is-a transport with (tp-obj :is-a saucer)
	pc-tp3 :is-a transport with (tp-obj :is-a cup)
	pc-cv :is-a cover
time marks:	pc-tb, pc-te :is-a timepoint
constraints:	pc-tp1.tp-ob = pc-cv.cv-pl
	pc-tp2.tp-ob = pc-cv.cv-sc
	pc-tp3.tp-ob = pc-cv.cv-cp
	...
	pc-tp3.tp-te $\geq$ pc-tp2.tp-te
	pc-tb $\leq$ pc-tp3.tb
	pc-te $\geq$ pc-cv.cv-tb

Figure 2: A frame-based model for a place-cover. From [13]

The frame-based model is an expressive formalism for representing a concept and can be paraphrased into other formalisms such as Description Logics, which has well-founded reasoning services.

2.2 Primitive Occurrence Detection

Under a controlled environment, it is possible to detect and track a limited number of objects and obtain object-level descriptions (e.g., object classes, motion flows, 3D trajectories) from a video sequence. To perform

a high-level scene interpretation, one must map these quantitative object-level descriptions into primitive occurrences defined as a basis in the conceptual framework. In most systems, the models of primitive occurrences are constructed manually based on an extensive knowledge of the domain of interest. Also, one must know the significant primitive occurrences in advance.

In [13], the mapping is achieved through multiple layers of abstraction. First, perceptual primitives are computed from object features, providing the measurements (e.g., distance, angle) between object features. Then, qualitative primitives are defined and computed as predicates over perceptual primitives. For example, a predicate "nearness" can be defined by a hard threshold on distance. Finally, a primitive occurrence is defined as a conceptual entity that characterizes the coherency of a qualitative primitive over a time interval (e.g., object in motion, object at rest). The computation of perceptual primitives and qualitative primitives in [13] requires a manual construction of the models and a careful hand-tuning of the parameters (e.g., in choosing the thresholds or segmenting the primitive occurrences). In Section 3, we discuss various mechanisms for automatic (or semi-automatic) detection of primitive occurrences.

2.3 Scene Interpretation

In [13], scene interpretation is based on a hypothesis-and-test procedure. An observed scene is described in terms of instantiated scene models from the CKB. These instantiated models are called hypotheses and are maintained (by verifying their spatial-temporal consistency) in the interpretation base. Primitive occurrences detected from object-level descriptions provide partial evidence and will be the entry points for part-whole reasoning. In part-whole reasoning, hypotheses in the interpretation base are generated and connected based on the part-of links specified in the models. For example, hypotheses about the transports of a plate and a saucer are parts of the "place-cover" and may generate a "place-cover" hypothesis, if all constraints are satisfied and considered as strong evidence. The verification of an instantiated scene model consists of propagating the time marks of the updated primitive occurrences to incrementally constrain the appropriate time marks of all connected entities in the interpretation base. Spatial constraints are also propagated in a similar fashion in a 3D space.

In [13], the construction of the model base and the implementation of the scene interpretation process are realized by a configuration system

called KonWerk [7]. The representation language in KonWerk is object-oriented and supports frame-like representations. KonWerk provides a mechanism for constructing a configuration (or a hypothesized scene model) and has a dedicated constraint verification system. A more recent work by Neumann et al. [13] explores the implementation of the scene interpretation using Description Logics (DL). DL also offers logical inferences based on formal semantics, similar to the inferences in first-order predicate logic. Such a logic-based reasoning system is useful for maintaining the consistency of a large knowledge base.

3 Learning of Event Models

A major issue in implementing a scene interpretation system is where the knowledge of all primitive occurrence types and the structures of scene models in CKB comes from. In most cases, this knowledge is explicitly programmed by a human user. However, this can be a tedious and error prone process. Also, when the system is applied in a different setting or context, a different set of primitive occurrences and event structures will be required. In this section, we discuss some of the techniques for acquiring such knowledge (semi-) automatically from actual video data.

3.1 Learning Primitive Occurrences

Primitive occurrences are basically qualitative representations of the continuous space of object-level descriptions and can be obtained by discretizing the space into a finite and small number of relevant possibilities (or prototypes). The results of discretization may be, hence, different for one domain from another. One common method for learning discrete representations is Vector Quantization (VQ) [6].

VQ is a data compression technique originally used for approximating the probability density function of a vector variable $x(t)$ using a finite number of prototype vectors $c_i(t), i = 1, 2, ..., k$. It has been also applied in event learning as a method for acquiring prototypical spatio-temporal representations. Galata et al. [10] use a VQ algorithm as a method for learning the discrete interaction primitives between two vehicles in a traffic scene. As object-level descriptions, a feature vector F_{r_t} is used to describe the relative velocity and the spatial relationship between a reference car and another car that falls within its attentional window. The object interaction

primitives are abstracted from sequences of feature vectors $F_{r_0}, F_{r_1}, \ldots, F_{r_m}$, by replacing F_{r_t} with their nearest (in a Euclidean sense) prototype from a finite set of prototypical object interactions. Figure 3 illustrates the learnt primitive interactions by VQ for the traffic domain example application. These can be viewed as a qualitative discretization of the continuous relational space. This representation is obtained by maximizing the discernability given a granularity (i.e., the number of relations desired).

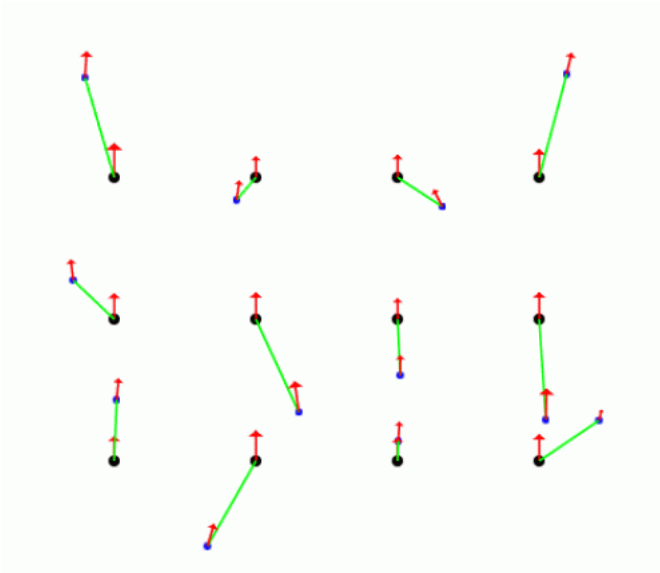

Figure 3: Learnt primitive interactions between a pair of vehicles in a traffic scene. From [10].

A limitation of most VQ techniques (specifically, the k-means algorithm) is that the final distribution of prototypes relies on the learning bias (e.g., the number of prototypes and their initial placement within the feature space). Choosing a wrong bias can result in sub-optimal distributions.

An alternative method for learning prototypes is based on detecting the coherency in qualitative primitives over a time interval. This is in contrast with the VQ, where the dynamics of a feature sequence is disregarded. A common observation is that an object moves in a task-oriented manner, during which time its motion is consistent. For example, consider a scene where CAR_1 is overtaking CAR_2. First, CAR_1 changes to a neighboring lane that is free of obstruction. Then, CAR_1 speeds up to position itself ahead of CAR_2. Finally, CAR_1 changes the lane back to be in front of CAR_2. Considering an overtaking action as a sequence of three types of

movements, a dynamic state-based model such as a hidden Markov model (HMM) can be used to learn the distributions of movement prototypes as well as the dynamics of the sequential actions (e.g., the time segmentation of motion prototypes). A limitation of the state-based model is that the number of prototypes must be given to obtain an optimal result (same as the VQ). In addition, it also requires that the video data must be segmented and labeled (e.g., “overtaking”, “following”).

3.2 Learning High-Level Scene Models

To our knowledge, there is very little amount of work in learning a dynamic scene model. One of the reasons is that treating time as an additional dimension of a 3D space and applying the same techniques for learning a 3D structure is not valid in general. Time is a unique dimension and can imply complex causal relations among actions. Such causality generates an exponential branching factor in the search space. Therefore, learning a high-level scene model often involves enforcing a learning bias (e.g., in terms of constraints on the structures or the representational languages) in order to control the search space. We divide the literature in this area into three categories: finite-state machine induction, temporal pattern mining and logic-based induction.

3.2.1 Finite-state machine induction

Arguably, one of the most common object behaviors is a consecutive sequence of actions. A classical learning approach is to use a probabilistic finite-state machine (e.g., HMM) to find temporal dependencies between primitive occurrences. An HMM of action sequences consists of a set of hidden event states, each of which encodes a characteristic distribution of motion features. The dynamics of action sequences are encoded by the probabilistic transitions among the event states. Given the number of states, these motion feature distributions and the state transition probabilities must be learned from the training video data. After learning, the HMMs can be used for segmenting sequences of primitive occurrences in a video. One of the weaknesses of a HMM is that the number of states are often not known in advance, requiring an extensive trial and error. Without an educated choice of the number of states, the HMM states do not necessarily correspond to occurrence concepts in a natural language. In recent years, some researchers have proposed methods for learning highly-

structured HMMs.

In [2] , Brand develops a method for discovering normative behavior in an office environment, which is represented by a "concise" hidden Markov model. The learning algorithm introduces and exploits an entropic prior for fast, simultaneous estimation of model structure and parameters. These entropically trained models are more concise and highly structured than the conventionally trained HMMs.

In [10], Galata et al. use variable-length Markov models (VLMMs) to encode interactive vehicle behavior in a traffic scene. A VLMM is a random process in which the memory length varies, in contrast to a first-order Markov model. The advantage of having a variable memory length is the ability to locally optimize the length of memory, capturing both higher-order and lower-order temporal dependencies adaptively. Also, in contrast with the hidden states of HMMs, the states of a VLMM correspond to the conceptual prototypes discussed in Section 3.1.

3.2.2 Temporal pattern mining

Techniques for mining temporal patterns have been studied mostly in the context of database mining. For example, events in a database of sales transactions may consist of customer transactions, each of which is tagged with a customer-id, the transaction time and the items bought. An example of temporal pattern may be that "computers and mice" are purchased first, followed by "printers", and then "memory sticks". Unlike in HMMs, events in such a pattern need not be consecutive and customers may purchase other things in between. Elements of a sequential pattern may also consist of other sub-patterns.

The research in sequence-mining contains many general-to-specific algorithms for finding sequences. In most earlier work, researchers have studied the problem of mining temporal patterns using languages that place constraints on partially or totally ordered sets of time points, e.g., sequential patterns [1] and episodes [15]. More recently there has been work on mining temporal patterns using interval-based pattern languages [9, 3, 8]. Even though the languages and learning frameworks vary among these approaches, they all use standard general-to-specific search techniques, where the learning results obtained at step k are used to constrain the search space of the models to be specified further at step $k+1$. One advantage of these methods is that the patterns are searched based on frequency and can handle a noisy temporal data set.

3.2.3 Logic-based induction

As described in section 2.1, temporal events can be represented using a logic-based language. Given a set of both positive and negative training examples, a common general-purpose relational learning technique such as inductive logic programming (ILP) [12] can be applied. In [4], relations among audio-visual concepts in a game playing *paper-scissor-stone* are learnt using PROGOL [11]. Relations of event concepts in PROGOL are represented by horn clauses and lack the handling of concrete domains (both time and space) necessary for a general visual event representation.

In [5], Fern et al. propose a simple logic called AMA (simplified from Allen's temporal logic) for representing temporal events. Spatial relations of objects are represented implicitly using simple concepts of force-dynamic (e.g., the touching of object bounding boxes). Based on the AMA language, they provide a mechanism for learning temporal, relational, force-dynamic event definitions from a positive-only input.

4 Discussion

In this review, we have illustrated several computational frameworks for dynamic scene interpretation. A common approach in Computer Vision is to model visual events as a statistical process (e.g., HMM). In a large conceptual knowledge base, such statistical approaches are doomed to fail and a logic-based representation provides an attractive alternative. In a logic-based approach, an event is modeled in a hierarchical fashion, where primitive occurrences are abstracted from pixel-based image representations. These primitive occurrences provide a basis, on which a scene model is constructed using logic-based languages.

We have reviewed some learning techniques that automatize the model construction processes for primitive occurrences and high-level scene models. Visual event learning is an inherently ill-posed problem, due to the complex causal relations induced by time. Even though recent advances have been made in all fronts, many of the existing learning techniques can only cope with simple event patterns.

A logic-based induction seems to provide a generic solution to the learning of hierarchical event models necessary for a large conceptual knowledge base. Logic-based inductive learning approaches are differentiated by the languages used for representing the events. While the logic language needs to be expressive enough to represent realistic scene models,

it is known that the determination of concept subsumption in the expressive First-Order Logics is semi-decidable. Most learning approaches often simplify the representational languages that limit the bounds of the subsumption and generalization problems. We believe that the trade-off between the expressiveness of the language and the complexity of inductive process is likely to play a key role in the future research.

References

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh on International Conference on Data Engineering*, pages 3–14, 1995.

[2] M. Brand. Pattern discovery via entropy minimization. In *Proceedings of the Artificial Intelligence and Statistics*, 1998.

[3] P. Cohen. Fluent learning: Elucidating the structure of episodes. In *Proceedings of the Fourth Symposium on Intelligent Data Analysis*, 2001.

[4] D. R. Magee, C. J. Needham, P. Santos, A. G. Cohn and D. C. Hogg. Autonomous learning for a cognitive agent using continous models and inductive logic programming from audio-visual input. In *Proceedings of the AAAI Workshop on Anchoring Symbols to Sensor Data*, pages 17–24, 2004.

[5] A. Fern, R. Givan and J. M. Siskind. Specific-to-general learning for temporal events with application to learning event definitions from video. *Journal of Artificial Intelligence Research*, 17:379–449, 2002.

[6] R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984.

[7] A. Guenter. Konwerk - ein modulares konfigurierungwerkzeug. In *F. Maurer, M. M. Richter (Hrsg.) Expertensysteme'95*, pages 1–18, infix Verlag, St. Augustin, Germany, 1995.

[8] F. Hoppner. Discovery of temporal patterns: Learning rules about the qualitative behaviour of time series. In *Proceedings of the Fifth European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2001.

[9] P. Kam and A. Fu. Discovering temporal patterns for interval-based events. In *Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery*, 2000.

[10] A. Galata, A. Cohn, D. Magee and D. Hogg. Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models. In *Proceedings of the European Conference on Artificial Intelligence*, July 2002.

[11] S. Muggleton. Inverting entailment and progol. *Machine Intelligence*, 14:133–188, 1995.

[12] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19-20:629–679, 1994.

[13] B. Neumann and R. Moeller. On scene interpretation with description logics. Technical Report FBI-HH-B-257/04, Dept. of Computer Science, Hamburg University, Hamburg, Germany, May 2004.

[14] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.

[15] H. Mannila, H. Toivonen and A. I. Verkamo. Discovery of frequent episodes in sequences. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 1995.

[16] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, pages 379–385, Champaign, IL, 1992.