Int J Comput Vis (2012) 98:83–102
DOI 10.1007/s11263-011-0497-0

Explaining Activities as Consistent Groups of Events

A Bayesian Framework Using Attribute Multiset Grammars

Dima Damen · David Hogg

Received: 8 August 2010 / Accepted: 13 September 2011 / Published online: 5 October 2011

Abstract We propose a method for disambiguating uncertain detections of events by seeking global explanations for activities. Given a noisy visual input, and exploiting our knowledge of the activity and its constraints, one can provide a consistent set of events explaining all the detections. The paper presents a complete framework that starts with a general way to formalise the set of global explanations for a given activity using attribute multiset grammars (AMG). An AMG combines the event hierarchy with the necessary features for recognition and algebraic constraints defining allowable combinations of events and features. Parsing a set of detections by such a grammar finds a consistent set of events that satisfies the activity's constraints. Each parse tree has a posterior probability in a Bayesian sense. To find the best parse tree, the grammar and a finite set of detections are mapped into a Bayesian network. The set of possible labellings of the Bayesian network corresponds to the set of all parse trees for a given set of detections. We compare greedy, multiple-hypotheses trees, reversible jump MCMC, and integer programming for finding the Maximum a Posteriori (MAP) solution over the space of explanations. The framework is tested for two applications; the activity in a bicycle rack and around a building entrance.

Keywords Activity analysis · Event recognition · Global explanations

D. Damen (✉)
Department of Computer Science, University of Bristol, Bristol, UK
e-mail: damen@cs.bris.ac.uk

D. Hogg
School of Computing, University of Leeds, Leeds, UK
e-mail: d.c.hogg@leeds.ac.uk

1 Introduction

While most existing activity recognition techniques deal with independent events (e.g. running, walking), realistic surveillance tasks typically involve multiple mutually dependent events, often extending over a long temporal duration. These dependencies can be exploited to disambiguate uncertain visual data by seeking a globally consistent explanation. The proposed framework bridges the gap between uncertain visual observations and higher-level activity recognition. Preliminary ideas for this work appeared in conference proceedings (Damen and Hogg 2009a, 2009b).

The paper begins with some definitions to clarify how the joint recognition of a set of events can be seen as a mapping from detections to a consistent global explanation. Section 2 compares this framework to previous approaches. Section 3 explains how attribute multiset grammars can define an event hierarchy along with its features and the activity's natural constraints. Given the grammar, a set of detections is mapped to a Bayesian network that models the probability distribution over the space of all parse trees for those detections. Section 4 explains the derivation of this distribution in terms of event likelihoods. Section 5 then examines the search for the Maximum a Posteriori (MAP) solution using heuristic and exhaustive techniques. Finally, Sect. 6 applies the framework to two activities, and tests on several challenging datasets.

1.1 Definitions

To analyse an activity automatically, evidence is gathered through observing the scene on which to base recognition of the occurring events. A *detector* is an independent evidence collector that targets a given type of entity. Such detectors have been widely used for event recognition, for

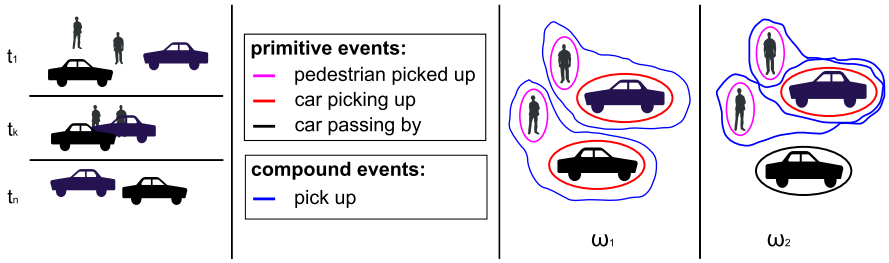

Fig. 1 For the same set of detections, and given primitive and compound events, two different global explanations ω_1 and ω_2 are shown, where each boundary corresponds to an event. In ω_1, each car picks up one pedestrian, while in ω_2 the *blue car* picks up both pedestrians

example in detecting motion (Joo and Chellappa 2006a; Nevatia et al. 2003; Rota and Thonnat 2000), cars (Huang and Russell 1998) and pedestrians (Nguyen et al. 2006). Some detectors are widely applicable and others are specific to a narrow domain. We refer to the output of a detector as a *detection*. A *feature* is a measurable characteristic of a detection.

The terms *activity* and *event* have been used in various, often ambiguous, ways within the computer vision community. To avoid confusion, the terms are defined here and then used consistently throughout the remainder of the paper. An *event* is a context-related interpretation for a detection or a group of detections. An *activity*, on the other hand, is a set of events. One can refer to the *activity* within the car park as the set of all events that occur within the car park. Similarly, the *activity* around the office is the set of events, which could be dependent or independent, yet are related by the space in which they occur. In the simplest case of only one event occurring, the activity and the event would be the same. In the general case, an activity involves multiple events.

We distinguish two kinds of event. A *primitive event* is detected directly and corresponds to one detection exactly. For example, a person walking across a car park could be treated as a primitive event. A *compound event* is a constrained grouping of simpler, compound or primitive, events. An *activity* is thus recursively defined as a composition of events, with primitive events as its elementary components. A *composition* is a hierarchy of events in which each level is made up of a consistent set of simpler events. A *consistent* set of events is one that satisfies the activity's natural constraints.

1.2 Global Explanations (GE)

The detections obtained during an observed period of activity typically belong to several events. A *global explanation* for a set of detections is a consistent set of events that covers all of these detections. The global explanation thus implicitly associates one or more detections with each event. The number of events is not known in advance, and varies between the different explanations for the same set of detections. To clarify, consider the problem of analysing the activity in a car park. Two detectors are available: one for moving cars and another for pedestrians. In both cases, the detections consist of object trajectories along with spatial and temporal features. Primitive events like a car stopping, and a pedestrian passing by, are defined. The compound event 'pick-up' is made up of three primitive events: a car stopping, a person stepping into the car, then the car driving away. Figure 1 shows the mapping from detections to multiple global explanations.

This mapping from detections to a global explanation is constrained. We assume three types of constraint. *Temporal constraints* allow or prevent temporally overlapping events, or enforce an ordering. For example, a person can enter a car only after it stops. *Spatial constraints* limit the separation of objects involved in an event, or the area in which the event occurs. For example, for a car to pick up a pedestrian, the pedestrian should appear within a certain distance from the car. *Sharing constraints* allow or prevent an event from participating in multiple compound events. For example, a car can pick up multiple people, but the same person cannot be picked up by multiple cars.

This paper proposes a framework that starts by formally defining an activity's events and its natural constraints. The set of detections in a video input is then explained in terms of these events and consistent with these constraints. Given prior probabilities, and the events' likelihoods, a Bayesian approach finds the best explanation that maximises the posterior probability. Figure 2 shows the different components of the framework. At the top of the figure, a box indicates the preparatory tasks to be performed once for each considered activity. The natural hierarchy and constraints are employed to create an Attribute Multiset Grammar (AMG). This process is manual, and the AMG is used, along with labelled training sequences, to define priors and likelihood functions that favour some global explanations over others. For a given video sequence, detectors gather a set of detections that are represented by terminal symbols of the grammar, along with the values of selected features (attribute values). A parse of the AMG generates a global explanation for all the detections. The framework proposes an algorithm to transform the AMG, given a finite set of detections, into a Bayesian network structure. Along with the learned probabilities, this Bayesian network models the probability distribution over the space of global explanations for this set of detections. The MAP solution of the Bayesian network is then believed to be the global explanation that best suits the detections.

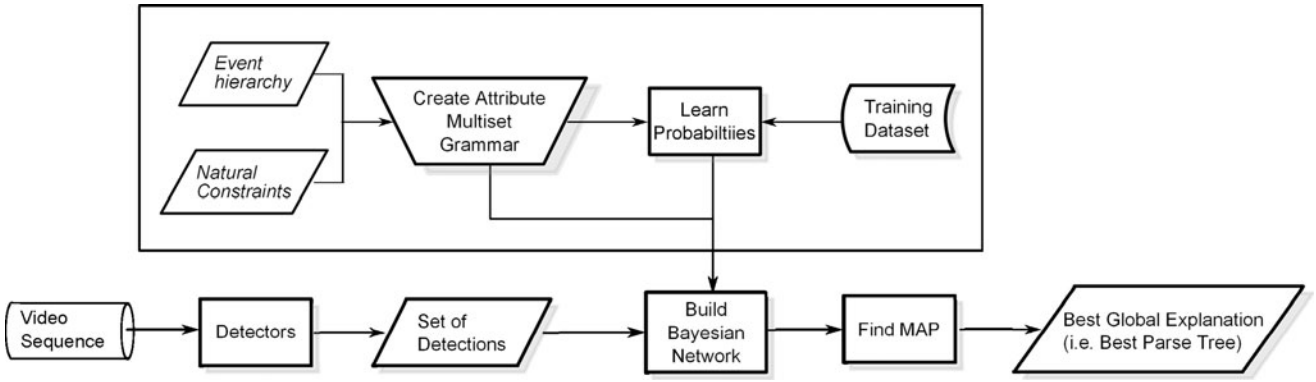

Fig. 2 A flowchart indicating the proposed framework

2 Background Review

Simultaneous analysis of all detections has proven advantageous in many areas of computer vision, such as image denoising (Geman and Geman 1984), segmentation (Rother et al. 2004) and object recognition (Felzenszwalb and Huttenlocher 2000; Wu and Nevatia 2005). As detections are noisy and often incomplete, global analysis outperforms local interpretation. By contrast, global analysis for activity recognition has not been widely explored. This section reviews previous work on the representation of domain knowledge about activities, and the use of such representations in recognition.

2.1 Representing Activities

The decomposition of an activity into a set of events, which can be further decomposed into simpler events, is naturally represented by a hierarchy. Grammars define possible hierarchies, and were used to define activities in video as early as 1998 (Young et al. 1998). A grammar provides a finite set of production rules. Parsing input using these rules results in a semantic interpretation, which can be shown using a *parse tree*. Different types of grammar have different expressive power. For example, ball passes between players in a game of tennis can be modelled using a regular grammar, while a context-free grammar can model football games allowing chains of passes of arbitrary length. For a review of different grammar types, the reader is referred to Aho et al. (1986).

Stochastic Context Free Grammars (SCFG) define a probability distribution over the possible rewrites for each non-terminal symbol within the grammar. This can be used to infer a probability distribution over the sentences of the language. Ivanov and Bobick used SCFG to represent the different ways in which activities can be composed, and demonstrated this for gesture recognition and surveillance within a car park (Ivanov and Bobick 2000). Although not part of the SCFG formalism, they also added a consistency check within the recognition process to enforce temporal constraints necessary for an explanation to be valid. Several non-grammatical linguistic methods have been proposed to incorporate such constraints directly into the formalism (Hongeng et al. 2004; Intille and Bobick 2001; Nevatia et al. 2003; Rota and Thonnat 2000; Siskind 2000; Shi et al. 2004).

In recent work, Tran and Davis (2008) use first-order logic production rules to encode domain knowledge. Four rule types are used: 'definite clauses' which are hierarchical decompositions of activities into events; 'disjunctions' which provide alternative decompositions; 'negative preconditions' which are constraints on applying the rules; and 'exclusion relations' that model constraints between events occurring at the same time. For example, an exclusion relation might specify that a person can drive only one car. Weights are assigned to the clauses to imply rule preferences.

Attribute grammars were originally proposed to extract semantics from the compositional structure of a parse tree (Knuth 1968) through propagating attribute values associated with terminal and non-terminal symbols up and down the tree. They have later been extended with constraints on attribute values that restrict the set of allowable parse trees. Such an approach has been used in vision to identify rectangular objects like floor tiles and windows in static images (Han and Zhu 2005). Strong rectangular candidates from edge detection are used to hypothesise larger structures through the application of grammar rules. This can initiate a search for weaker evidence of rectangles consistent with these structures. The result is a hybrid of top-down and bottom-up processing combined with Markov chain Monte Carlo (MCMC) sampling (Zhu and Mumford 2006). Attribute grammars have recently been used to recognise activities in a car park (Joo and Chellappa 2006b; Lin et al. 2009), although these approaches do not employ the full capabilities of attribute grammars, as they do not use inherited attributes or inherited constraints (explained in Sect. 3).

2.2 Recognising Activities

The activity's representation is then used to recognise events from video input. Single event recognition had used graphical models like hidden Markov models (HMMs) (Ivanov and Bobick 2000; Nguyen et al. 2006) and general Bayesian Networks (Kitani et al. 2005; Intille and Bobick 2001), and partitioned detections into events using Markov Random Fields (MRF) (Lin et al. 2009) or data association techniques (Nguyen et al. 2006; Smith 2007).

In Ivanov and Bobick (2000), recognition is decoupled into two stages: (i) a set of HMMs detects primitive events, and (ii) a modified Earley-Stolcke parser generates the parse with the highest posterior probability given a sequence of uncertain events and the SCFG. A single compound event, involving interacting agents, is recognised in each given video.

Kitani et al. (2005) build a hierarchical Bayesian network from an SCFG. Instead of a parser, 'deleted interpolation' is used to find the explanation with the maximum posterior probability. In deleted interpolation, the probability distribution at each point in time is calculated as a weighted sum of pieces of evidence within a fixed-size window. A solution that better explains recent observations is favoured. Intille and Bobick (2001) also build a Bayesian network and represent each event by a ternary observed node (yes/maybe/no). When applied to the activity of American football, multiple Bayesian networks for different strategies are tested at each point in time to determine which strategy is used by the players. The network with the highest confidence is selected as the recognised strategy. Shi et al. (2004) use particle filtering to sample the space of explanations. This outperforms the parsing from Ivanov and Bobick (2000) in recovering from errors and uncertainties in the data.

Although most prior work on activity recognition has focused on recognising a single event instance from a set of detections, some recent work deals with the more realistic situation in which the detections arise from multiple events within an activity. The approaches in Fan et al. (2009), and Joo and Chellappa (2006a) assign detections to events greedily in a sequential order during recognition. Nguyen et al. (2006) use a combined hierarchical hidden Markov model along with the joint probabilistic data association filter (HHMM-JPDAF) to jointly assign detections and recognise complex events. The approach uses MCMC to sample from the set of possible assignments, then exact inference is used for each HHMM to recognise the event. This requires the number of events to be fixed and known in advance in order to decide on the number of HHMMs. The assignment assumes each detection participates in one and only one event.

Another recent attempt to partition detections into events combines SCFG with a MRF (Lin et al. 2009). The MRF defines the joint probability on nodes in the possible parse trees. The unary term defines an event's likelihood, while pairwise terms define the relationships between nodes. Applied to picking up people in a car park, the pairwise potentials in the MRF are calculated from the spatial proximities of people and cars. A Gibbs sampler is used to find the best set of objects for each event. While this framework can partition the detections, it can not handle the constraints between events in an obvious way, like allowing the car to pick up several people while the person can be picked up by one car at most.

In this paper, we propose a framework that unifies the analysis of primitive and compound events within a single optimisation procedure, similar to Nguyen et al. (2006), while handling a variable number of compound events. The representation allows for instances of simple events to be part of more than one compound event. It also deemphasises the temporal ordering of constituent events by moving to an attribute multi-set grammar in which strings are replaced by bags, and any required ordering is instead represented by constraints over attribute values.

The problem of assigning detections to events has been explored in the more general setting of *data association*. The canonical problem is to find a mapping of detections to a previously unknown number of identities (in this case events), whilst satisfying 'association' constraints. Data association has been employed often in tracking to assign detections or measurements to targets, and to solve the exponential complexity of the search space. Heuristic techniques have included Multiple-Hypotheses Trees (MHT) (Reid 1979; Huang and Russell 1998) and sampling the distribution of associations using importance sampling (Wu and Huang 2004) or MCMC (Oh et al. 2004; Zhao and Nevatia 2004; Smith 2007; Yu et al. 2007). Smith (2007) uses Reversible Jump MCMC (RJMCMC) in a sliding window, and the globally optimal trajectories are computed for each window independently. An exact search technique formulates the problem as a set packing task, and solves it using integer programming (Morefield 1977).

3 Defining Global Explanations of Activities

Attribute Grammars as first introduced by Knuth (1968), also referred to as Feature-Based Grammars (Blevins 2001) and Attribute-Value Grammars (Abney 1997), add attributes to the terminal and nonterminal symbols of a grammar. Attribute rules are associated with the production rules of the grammar and propagate information up towards the root of the parse tree, or down towards the leaves. The motivation was to provide a way to compute semantics in a compositional fashion from a parse tree. Although not in Knuth's original formulation, the attributes can also be used to govern the application of production rules, thereby constraining the language generated by the grammar.

Attribute Multiset Grammars (AMG) were introduced in Gollin (1991) for representing the allowable constituents of visual languages, like defining grammars for flowcharts and state diagrams using terminals such as circles, rectangles and arrows. A multiset (or a bag) is a generalisation of a set where each element (symbol) can appear more than once. As for a set, there is no ordering of the elements. AMGs generalise attribute grammars by removing the sequential ordering of symbols in a sentence, requiring only a multiset of symbols. The same terminal symbol, representing a particular graphical component for example, may appear more than once. We use the formalism from Gollin (1991) in the rest of the paper. This is adapted from Knuth's original terminology (Knuth 1968).

An AMG is defined as a five-tuple $G = (N, T, S, A, P)$ where N is the set of nonterminal symbols denoted with capital letters, T is the set of terminal symbols denoted by lower case letters, S is the start symbol, $A(X)$ is a set of attributes defined for the symbol $X \in N \cup T$, and P is the set of production rules. The notation $X.a$ is used to denote the value of the attribute $a \in A(X)$. Attributes are of two types, $A(X) = A_0(X) \cup A_1(X)$, where $A_0(X)$ is the set of *synthetic* attributes, which have predefined values for all terminals and are calculated for nonterminals based on their children, and $A_1(X)$ is the set of *inherited* attributes, which are calculated based on the attributes of the parents.

Each production rule $p \in P$ is a three-tuple (r, M, C), where r is a *syntactic rule* of the form

$$X_0 \rightarrow X_1, X_2, \ldots, X_{n_p}$$

that rewrites the nonterminal X_0 as a multiset of nonterminal and terminal symbols $X_1, X_2, \ldots, X_{n_p}$. M is a set of *attribute rules*, where each rule $m \in M = M_0 \cup M_1$ assigns a value to one of the attributes of the symbols involved in r. A synthetic attribute rule $m \in M_0$ assigns a value to a synthetic attribute, while $m \in M_1$ assigns a value to an inherited attribute. A set of *attribute constraints* $C = C_0 \cup C_1$ governs the application of the production rule. A parse tree belongs to the grammar's language only if all attribute constraints of the applied production rules are satisfied. An AMG can thus define an activity as follows:

- The start symbol (S) represents the complete activity.
- Nonterminal symbols (N) represent the compound events that can be rewritten into a multiset of simpler events.
- Terminal symbols (T) represent primitive events that are directly detected.
- Synthetic attributes (A_0) are distinguishing features, originating from the detections.
- Inherited attributes (A_1) are explanation-related attributes, like the number of people picked up by one car (Fig. 1). Such attributes are not calculated from the detections, but are part of the explanation, and differ between explanations.
- Synthetic constraints (C_0) define temporal and spatial constraints.
- Inherited constraints (C_1) impose consistency between the constituent events forming an explanation.

The key difference between AMG and conventional string grammar is the absence of a sequential ordering. For string grammars, allowable variations in ordering must be dealt with through the grammar rules—each possible ordering is defined in a separate rule. When such variation is the norm, or when events can occur in parallel, this becomes unwieldy. In AMG by contrast the grammar rules only define the permitted composition of entities (in our case events)—allowable relations between entities (e.g. temporal or spatial relations) are specified via the attribute constraints. This is convenient when there are relatively few such constraints.

Using a multiset instead of a set means that symbols may appear multiple times. An activity can contain multiple instances of the same event. Note that two event instances of the same type are considered identical, which motivated the usage of multiset grammar. In our use of AMGs we also assume *multiple consumption*—each terminal or nonterminal symbol $x \in T \cup N$ can be consumed more than once in the parse tree. This allows the same detection or event to be part of multiple complex events. One can think of this as a cloned copy of the node in the parse tree that shares the same attribute values. Used without care, this could result in an infinite number of parses for a given input. We prevent this through the use of 'counting' attributes and associated constraints which implement natural constraints of our activity domain. For example, while the car can pick up multiple people, the person can be picked up by one car at most.

After a parse tree is built, attribute values are calculated using the attribute rules. For attribute grammars in general, assumptions are normally made about the order in which attributes are computed, assuming such an ordering exists and there is no circularity (Kastens 1980). In our case, we assume a strict ordering of evaluation as follows. First, all synthetic attributes are evaluated bottom-up until the root is reached. Next, inherited attributes are evaluated in a top-down manner until leaf nodes are reached. This implies synthetic attribute rules do not require any inherited attribute values. When multiple attribute rules are associated with the same production rules, they are evaluated in the order in which they appear in the grammar. Because of multiple consumption, a node of the parse tree may have more than one parent. When this occurs, the attribute rules are evaluated in an arbitrary order. We assume the attribute rules are such that the resulting attribute values are invariant to the chosen ordering. Finally, the attribute constraints are evaluated. A parse tree is invalid if any constraint is broken.

To illustrate, consider the AMG G_1 in Table 1. For each input video, detectors are used to retrieve a set of detec-

Table 1 AMG example G_1

Terminals (T):	a, b, c	primitive events
Nonterminals (N):	S, A, B	compound events

Attributes (A):

attribute name	type	domain	defined for
t	A_0	$\mathbb{Z}$	$\{a, b, c, A, B\}$
count (default = 0)	A_1	$\mathbb{Z}$	$\{b, B\}$

Production Rules (P):

rule	Syntactic Rule (r)	Attribute Rules (M)			Attribute Constraints (C)		
p_1	$S \rightarrow A^\star, B^\star, a^\star, c^\star$						
p_2	$A \rightarrow a, B$	$A.t$	$=$	$a.t + B.t$	$a.t$	$<$	$B.t$
		B.*count*	$=$	1	$B.count$	$\neq$	1
p_3	$B \rightarrow b, c$	$B.t$	$=$	$c.t$	$b.t$	$<$	$c.t$
		b.*count*	$=$	$B.count$	$b.count$	$\neq$	1

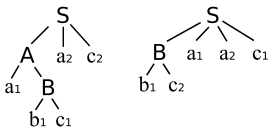

Fig. 3 Two parse trees given a set of detections and AMG G_1

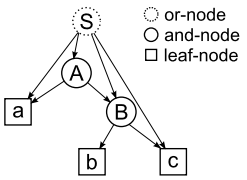

Fig. 4 And-Or graph representation of the grammar G_1

tions D. Each detection is an instance of one of the terminals T in the grammar, together with assigned values for the synthetic attributes defined for that terminal. The set of all derivations of D, given G_1, is the set of all possible explanations for the input video. For the grammar G_1, suppose the detectors generated the following multiset $D = \{a_1\ (t = 1),\ a_2\ (t = 2),\ b_1\ (t = 2),\ c_1\ (t = 3),\ c_2\ (t = 4)\}$—subscripts distinguish different instances of the same terminal. Values for the synthetic attribute t are assigned by the detector for each terminal symbol. Figure 3 shows two possible parse trees. Recall that the left-right order of branches from each non-terminal in the tree is irrelevant.

The basic compositional rules of our AMG formalism are equivalent to those of an And-Or graph (Nilsson 1971). However, the addition of attributes and associated constraints within an AMG provides much greater control over allowable parses. To illustrate, Fig. 4 shows the And-Or graph that is equivalent to the compositional rules in G_1 (excluding attribute rules and constraints).

4 Probability Distribution over Global Explanations

To find the best explanation (i.e. parse tree) for a set of detections and a given AMG, a probability distribution over the space of possible explanations is modelled as a Bayesian network (BN). This section explains the structure of the BN along with the procedure for generating this BN.

The BN contains three kinds of node. The first are Boolean 'event-nodes' representing the presence or absence of possible events in the explanation. There is an event-node for every primitive or compound event derivable from the set of detections. These are hidden nodes in the BN, and a global explanation is a complete labelling of the event-nodes in which the value of a node is true if and only if the corresponding event is present in the explanation. The joint probability of all event-nodes is factorised so compound events are only dependent on their constituent events, according to the given AMG. The second kind are 'observation-nodes' representing continuous or discrete synthetic attribute values obtained from the detectors. These are shaded in the figures to indicate that their values are assumed known. There is an edge connecting each event-node to its associated observation-node. The associated likelihood is a function of the attribute values for the possible event corresponding to the event-node. The third kinds of node are Boolean 'constraint-nodes'—set as true for explanations constrained by the AMG. Each constraint-node is connected to the event-nodes over which the corresponding constraint operates. These are deterministic variables in the BN (denoted by double-circled nodes), as each is functionally dependent on the values of its parents using a Boolean function. A constraint-node evaluates to true if and only if the corresponding constraint specified in the AMG is satisfied. This

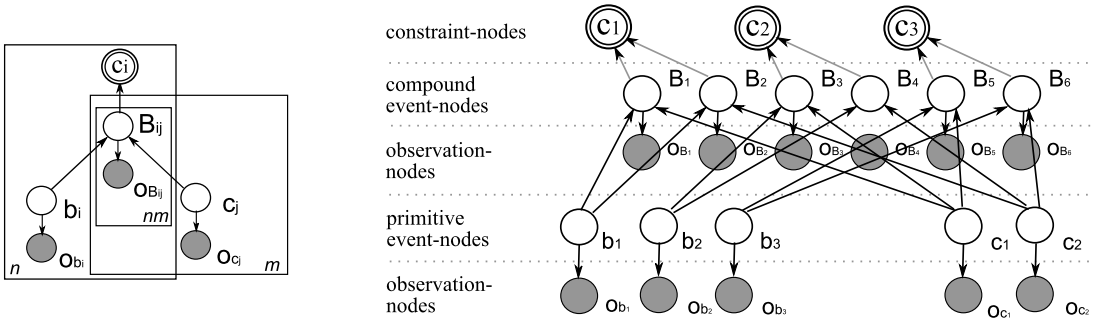

Fig. 5 A plate and unrolled BN for the simple AMG in Table 1, restricted to the single rule $p_3 : B \rightarrow b, c$

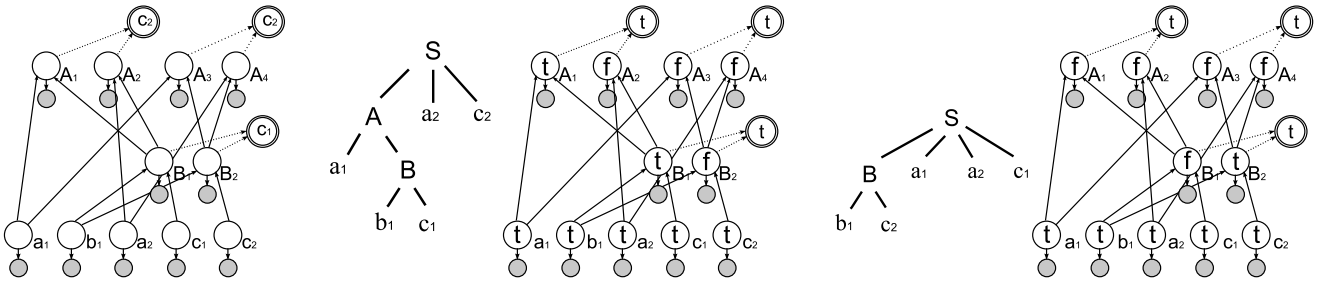

Fig. 6 The Bayesian network for the grammar G_1 along with two labellings that reflect the parse trees in Fig. 3. An event-node is labelled true if the event appears in the parse tree

implies that the joint probability of the BN is zero if any constraint is broken.

To illustrate, Fig. 5 shows the BN generated for the third rule of the simple AMG in Table 1 ($B \rightarrow b, c$), assuming N detections of b and M detections of c, using a plate representation. Also shown is the rolled-out BN for $N = 3$ and $M = 2$, with the different kinds of nodes shown in layers. Note that descendants in a parse tree are parents in the BN.

Algorithm 1 details the steps for building a BN out of a set of detections D and an AMG. First, an event-node is created for each detection $d \in D$. Rules are then considered one-by-one, starting from those rewriting a nonterminal into a multi-set of terminals. For each rule, all combinations of available event-nodes that can be parsed by that rule is considered. The synthetic constraints are checked, and when satisfied, an event-node is created for the nonterminal at the left-hand-side of the production rule. To allow for *direct recursion* in grammars, the if-statement (line 20) checks for new possible multisets of event-nodes in the BN. The current algorithm cannot deal with indirect recursion. Lines 23–30 explain how inter-dependent nodes can be found and linked to deterministic random variables. Algorithm 1 assumes a mapping is known between each inherited constraint and a Boolean function to evaluate that constraint. In all the examples given in this paper, inherited constraints are confined to equality and inequality statements that are mapped to Boolean functions using Boolean operators. For example, in AMG G_1, the inherited constraint $b.count \neq 1$ combined with the inherited rule $b.count = 1$ implies the rule can be parsed only once for each b detection. In the BN, only one parent node of each b can thus be labelled true. The corresponding Boolean function for this constraint, given the parent nodes B_1, B_2, would be $\neg(B_1.count \wedge B_2.count)$. Figure 6 shows the Bayesian network for AMG G_1 and the example detection multiset from Sect. 3 along with two labellings that correspond to the parse trees in Fig. 3.

After defining the topology of the BN, priors and conditional probabilities are specified. To find the best explanation, one needs to infer the MAP labelling $\omega^\star$ of the event-nodes, given the observation-nodes Y:

$$\omega^\star = \arg\max_{\omega} p(\omega|Y) \tag{1}$$

For the BN obtained from one production rule in Fig. 5, and set of detections $\{b_i\}$, $\{c_j\}$, the posterior is written as:

$$p(\omega|Y) = \frac{1}{\mathcal{G}} \prod_i p(o_{b_i}|b_i)p(b_i) \prod_j p(o_{c_j}|c_j)p(c_j) \times \prod_{ij} p(o_{B_{ij}}|B_{ij})p(B_{ij}|b_i, c_j)p(\mathbf{c}|\{B_{ij}\}) \tag{2}$$

The posterior can be re-arranged, and the third factor in (2) can be replaced by a proportional quantity to ensure

input : Grammar $G = (N, T, S, A, P)$, detections set D
output : Bayesian network structure BN

%%% Build Bayesian network structure
initialise an empty Bayesian Network (BN)
foreach *terminal instance* $t \in D$
    add event-node to BN of type t
    if *t has synthetic attributes* **then**
        add a related observation-node to hold the synthetic attribute values
order rules P starting with those containing terminals then bottom-up
foreach *rule* $p \in P$ *(p.r:* $X_0 \rightarrow X_1, X_2, \ldots, X_n$*);* $X_0 \neq S$
    Let I(X_i) be the set of event-nodes in BN of type X_i
    $comb = I(X_1) \times I(X_2) \times \cdots \times I(X_n)$
    while $comb \neq \phi$ **do**
        multiset $b = comb(1)$ % first multiset in comb
        $comb = comb - b$
        if *b satisfies synthetic attribute constraints* $p.C_0$ **then**
            add event-node R to the BN of type X_0
            foreach *synthetic attribute rule* $m \in p.M_0$
                apply m assigning a synthetic attribute value to observation-node of X_0
            all event-nodes in the multiset b parent the created event-node
            if *recursive rule* p **then**
                Let $A(b, X_i)$ be the set of all ancestors of b of type X_i
                $comb_2 = \{I(X_1) - A(b, X_1)\} \times \cdots \times R \times \cdots \times \{I(X_n) - A(b, X_n)\}$
                $comb = comb \cup comb_2$
%%% Find inter-dependent nodes
Let $Nodes_n$ be the set of all event-nodes
while $Nodes_n \neq \phi$ **do**
    find $Nodes_p$ with inherited constraints limiting the same inherited attribute values
    $Nodes_n = Nodes_n - Nodes_p$
    if *size of* $Nodes_p > 1$ **then**
        add constraint-node c to hold the inherited constraints
        all event-nodes in $Nodes_p$ parent the constraint-node c

Algorithm 1: Mapping a set of detections D to the Bayesian network (BN) representing the probability distribution over the possible parses, given an AMG G

tractability (Appendix),

$$p(\omega|Y) = \frac{1}{Q}\prod_i p(b_i|o_{b_i})\prod_j p(c_j|o_{c_j}) \times \prod_{ij:B_{ij}=t}\frac{p(B_{ij}=t|b_i,c_j,o_{B_{ij}})}{p(B_{ij}=f|b_i,c_j,o_{B_{ij}})}\prod_{ij} p(\mathbf{c}|\{B_{ij}\}) \quad (3)$$

Accordingly, evaluating the posterior of a single parse tree takes into consideration only the compound events recognised within the parse tree, and is not concerned with the remaining unrecognised events. This uses the fact that labelling all the event-nodes as false is a fixed quantity. For event-nodes labelled true, the ratios of labelling each node as true to labelling it as false are sufficient to compare the posterior across all labellings of the Bayesian network.

Event-nodes in the BN correspond to possible events in a parse tree derivable from a given set of detections. Although the explanation until now has focused on BNs with Boolean event-nodes, nothing restricts the approach from extending to multi-labelled event-nodes. Such a node can be labelled with one of several possible event types, or a false labelling which implies none of the possible events has occurred. This is suitable for AMGs where there exists more than one consistent setting for inherited attributes associated with structurally identical compositions of primitive events. In this case, the labels for the node in the BN are augmented to denote these different possible settings in addition to 'false'. We use multi-labelled event nodes in the AMG for the *Bicycles* problem in Sect. 6.

5 Searching the Bayesian Network

We explore four methods for finding the MAP explanation for a given BN. Three of these are approximate methods: Greedy search (G), Multiple Hypothesis Tree (MHT) and sampling the distribution using Reversible Jump Markov Chain Monte Carlo (RJMCMC). The fourth method is guaranteed globally optimal and involves the search as an Integer Program (IP). While IP delivers better explanations, an increase in the search space makes IP intractable and the heuristic methods come into their own (Sect. 6).

Greedy search (G) assigns labels to event-nodes working from the bottom layer up and checking constraints at each stage. At each level, the nodes at that level $\{x_i\}$ are sorted by l_{x_i}:

$$l_{x_i} = \frac{p(x_i = t|pa(x_i), o_{x_i})}{p(x_i = f|pa(x_i), o_{x_i})} \quad (4)$$

where $pa(x_i)$ is the (labelled) set of parents of the node x_i. If $l_{x_i} \geq 1$ then x_i is labelled true, unless the explanation becomes inconsistent. The evaluation continues up the hierarchy until all nodes are labelled.

Multiple Hypotheses Tree (MHT) (Reid 1979) propagates a tree of multiple hypotheses (explanations). It assumes an ordering (usually temporal) and starts from the first detection working through to the last. Each level in the search tree is expanded into nodes representing the different hypotheses explaining the detection in hand. Each path,

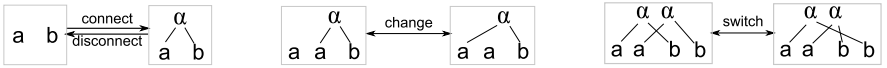

Fig. 7 Four move types to link events, break links, change linked events and switch links

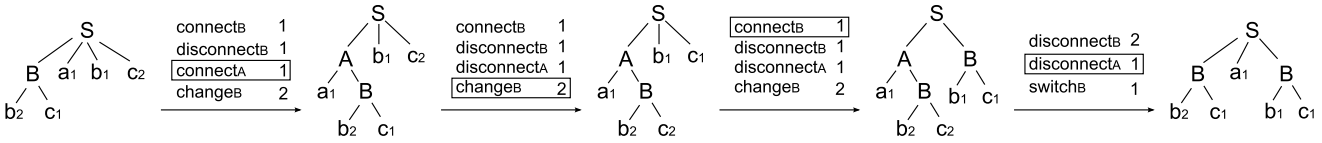

Fig. 8 Four moves are applied in sequence. The label at each *arrow* shows the number of possible moves of each type. The *rectangle* indicates the chosen move type

from root to leaf, in the search tree corresponds to an explanation. Due to the ambiguities in the visual data, the current best path may not be part of the best path to lower levels of the search tree as it propagates into the future. The search tree is pruned at each step to keep the search tractable by retaining only the best k hypotheses. The number of retained branches, k, is selected based on a trade-off between number of calculations and accuracy.

Markov Chain Monte Carlo (MCMC) samples the posterior distribution $\pi(\omega) = p(\omega|Y)$ using a Markov chain. A conditional *proposal distribution* $Q(\omega'|\omega)$ defines the probability of proposing state ω' given the current state is ω. After a state is proposed using Q, the move to that state is made with the probability $\alpha(\omega'|\omega)$ known as the *acceptance probability*. A thorough review of MCMC techniques can be found in Andrieu et al. (2003). The space of possible explanations is a discrete space, thus moves are designed to change a certain explanation ω into a slightly different one, such that the constraints are still satisfied. Green suggested using Reversible Jump MCMC for sampling the joint distribution of both the model dimension and the model parameters (Green 1995). By analogy, given a set of detections, the search is for the number of events and which detections belong to each event. RJMCMC generalises the acceptance probability to include the probability of selecting the move type, and a move-specific probability (Green 2003).

$$\alpha(\omega'|\omega) = \min\left(1, \frac{\pi(\omega')}{\pi(\omega)} \frac{j_{m^R}(\omega')}{j_m(\omega)} \frac{g_{m^R}(u')}{g_m(u)} \left|\frac{\partial(\omega', u')}{\partial(\omega, u)}\right|\right) \quad (5)$$

In (5), assume ξ represents the set of all move types, then $j_m(\omega)$ is the probability of selecting the move type $m \in \xi$ given the current explanation is ω. For each move type m, m^R refers to the reverse move type. Some move types are self-reversible, which means a move of the same type is applied to revert the change. The random variable u is a parameter for applying the move type m and transforming the current explanation ω to the new explanation ω'. The last factor in (5) is the absolute determinant of the Jacobian matrix of this diffeomorphism, which equals the identity matrix for the moves proposed here (Smith 2007 for proofs).

For binary event hierarchies where each production rule in the AMG replaces a symbol by a multiset of two symbols, four move types were designed to traverse the search space (Fig. 7). It should be noted that this is not the minimal set of move types. Adding 'change' and 'switch' move types enables efficient search of the space and faster convergence.

For the grammar G_1 and an initial configuration ω_0, Fig. 8 shows a typical Markov chain. At each step, a list of possible move types with the number of possible moves of each type is shown on the arrow. A subscript indicates the layer at which the move is applied. $connect_B$, for example, recognises a compound event of type B. In presenting the figure, the parse tree is shown rather than the labelled BN. Recall that there is a one-to-one mapping between a labelled BN and a parse tree. When searching the space of explanations using MCMC, the BN need not be actually built. RJMCMC jumps between the different explanations, and avoids unlikely explanations, without requiring the BN structure. Once a move is applied, the attribute values are re-evaluated for affected parts of the tree. Similar to the order in Sect. 3, synthetic attribute rules are first evaluated bottom-up, followed by inherited attribute rules. For reaching the maximum faster, simulated annealing (SA) is added to the MCMC sampling.

Finally, we use integer programming (IP), which is an exhaustive search technique. The list of all partial explanations $[\lambda_i]$ is first accumulated. Assume there are r partial explanations, the explanation ω is then an r-dimensional vector of 0s and 1s. In the case of global explanations for activities, a partial explanation is one event from the possible set of events (primitive or complex) along with all its constituent events (in the case of compound events). For the detection set $D = \{a_1(time = 1), a_2(time = 2), b_1(time = 2), c_1(time = 3), c_2(time = 4)\}$, the list is:

λ_0: a_1 λ_4: B_1, b_1, c_1 λ_8: A_3, a_1, B_2, b_1, c_2
λ_1: a_2 λ_5: B_2, b_1, c_2 λ_9: A_4, a_2, B_2, b_1, c_2
λ_2: c_1 λ_6: A_1, a_1, B_1, b_1, c_1
λ_3: c_2 λ_7: A_2, a_2, B_1, b_1, c_1

The probability of each partial explanation can be calculated independently. Assume v is an r-dimensional real-valued

vector where $v_i = \log(p(\lambda_i))$. The search for the MAP solution using IP would be to find max $v'\omega$. This is because

$$v'\omega = \sum_{i:\omega_i=1} v_i = \sum_{i:\omega_i=1} \log(p(\lambda_i)) \tag{6}$$

Accordingly, $\omega_1 = [0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0]'$ and $\omega_2 = [1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0]'$ correspond to the parse trees in Fig. 3. The posterior of each explanation is $v'\omega_1$ and $v'\omega_2$.

While maximising $v'\omega$, some of the r-dimensional binary vectors are an inconsistent or incomplete set of events. IP includes constraints that ensure the resulting set of events makes up a global explanation. Three constraints are defined for global explanations: all terminals need to be explained (c_1), sharing constraints satisfied (c_2), and occurrence of events in multiple partial explanations preserved (c_3). For c_1 a matrix τ of size $d \times r$, where $d = |D|$ is the number of detections, is constructed so $\tau_{ij} = 1$ if terminal i is explained by the partial explanation j. Similarly for c_2, a matrix θ of size $m \times r$ is constructed, where m is the number of deterministic nodes in the BN, and $\theta_{ij} = 1$ if any inter-dependent node parenting the deterministic node i is explained in the partial explanation j. For c_3, a matrix κ of size $n \times r$, where n is the total number of event-nodes in the BN, is constructed so $\kappa_{ij} = 0$ if node i is not labelled in the partial explanation j, $\kappa_{ij} = 1$ if it is labelled as 'true' and $\kappa_{ij} = 2$ otherwise. The linear optimisation problem is then:

Given matrices $\tau_{d\times r}$, $\theta_{m\times r}$, $\kappa_{n\times r}$ and cost vector v_r, find max $v'\omega$ such that

$$\tau\omega \geq \mathbf{1}, \quad \text{and}$$

$$\theta\omega \leq \mathbf{1}, \quad \text{and}$$

$$\kappa\omega\omega'\kappa' = \mathbf{0}$$

$$\omega \in \mathbb{Z}^r$$

This integer program has one nonlinear constraint that can be converted into a set of linear inequalities (Williams 1999). We use XPRESS-MP to solve the standard linear optimisation (FICO 2007). The search techniques presented in this section are experimentally compared in the next section on two activities.

6 Applications and Results

The proposed framework has been applied in two case studies. The first is in recognising the activity in a bicycle rack, and the second is in associating people and any objects they might be carrying into and out of a building.

6.1 The *Bicycles* Problem

In the *Bicycles* problem, a CCTV camera overlooks a bicycle rack where people lock their bicycles and retrieve them later. We refer to the act of leaving the bicycle in the rack as a **drop**, and the act of retrieving the bicycle as a **pick**. The task is to correctly associate people to the bicycle they have dropped or picked, and to link picks to earlier drops when the corresponding events are both observed. Two types of detections are considered; the first is of people entering and leaving the rack area, and the second is of changes within the racks that indicate the appearance and disappearance of bicycles. These are referred to as 'bicycle-clusters', as each may contain multiple bicycles.

Fig. 9 An activity unit showing 5 individuals (*left*) and 3 bicycle-clusters (*right*)

The *Bicycles* problem is challenging because bicycles are parked very close to each other and are sometimes 'piled' on top of one another. Association ambiguities increase when there are several people in the rack area at the same time. We refer to the intervals during which one or more people are in the rack area as 'activity units', consistent with the terminology in Gong and Xiang (2003) for plane refuelling scenes. Figure 9 illustrates an activity unit by highlighting the detected people and bicycle-clusters. Within an activity unit, each person can be linked to one bicycle-cluster at most, as we assume a person cannot drop or pick more than one bicycle per visit to a rack. On the higher level, each drop can be connected to one pick at most from a later activity unit, and vice versa.

To detect people entering and leaving the rack, an off-the-shelf blob tracker is used (Magee 2002). We define a person detection as starting from the first appearance of a moving blob within the field of view and ending when the blob departs the scene or is fully occluded. The same person returning to the rack is treated as a new detection. To detect bicycles, reference images of the rack area are compared, revealing changed pixels representing objects that have been deposited and removed. The changed image pixels are grouped into connected regions representing bicycle-clusters. Further details on the two detectors can be found in Damen (2009).

An AMG for the *Bicycles* problem, using the notation from Sect. 3, is given in Tables 2 and 3. The AMG contains 5 production rules. Each syntactic rule is associated with attribute rules and constraints. In p_2, possible drops are only linked to picks in later activity units ($Z_1.au < Z_2.au$). In p_5 drop and pick events between people and bicycle-clusters should be detected within the same activity unit

Table 2 AMG for the *Bicycles* problem: terminals, non-terminals, attributes and functions

Terminals (T):	x	person dropping or picking a bicycle
	y	dropped or picked bicycle cluster (i.e. one or more bicycle)
	u	Unobserved drops or picks
Nonterminals (N):	S	Start symbol representing the global explanation
	V	Drop-Pick: relates a drop event to a later pick
	Z	Drop or pick: person drops/picks a bicycle to/from a bicycle-cluster

Attributes (A):

	att. name	type	domain	description
x	id	A_0	$\mathbb{Z}$	a unique id differentiating people detections
	au	A_0	$\mathbb{Z}$	activity unit during which the person was detected
	traj	A_0	$\mathbb{Z}^{4n}$	bounding boxes representing the extent of the person in each frame
	sizeR	A_0	$\mathbb{R}$	ratio of the mean number of pixels representing the foreground before the person enters the rack area to the mean number after departing
	count	A_1	{0,1}	number of events in which the person participates
	action	A_1	{drop (d), pick (p), pass-by (f)}	
y	au	A_0	$\mathbb{Z}$	activity unit at which the cluster was detected
	pos	A_0	$\mathbb{Z}^4$	bounding box of the cluster
	fMap	A_0	Image	map of foreground pixels representing the cluster
	edgeR	A_0	$\mathbb{R}$	ratio of new to removed edges within the cluster
	count = 0	A_1	$\mathbb{Z}^*$	inferred number of bicycles in the bicycle-cluster
	action	A_1	{drop (d), pick (p), noise (f)}	
Z	id	A_0	$\mathbb{Z}$	$= x.id$
	pos	A_0	$\mathbb{Z}^4$	$= y.pos$
	au	A_0	$\mathbb{Z}$	$= x.au$
	traj	A_0	$\mathbb{Z}^{4n}$	$= x.traj$
	edgeR	A_0	$\mathbb{R}$	$= y.edgeR$
	fMap	A_0	Image	$= y.fMap$
	dist	A_0	$\mathbb{R}$	spatial proximity between x and y
	count	A_1	{0, 1}	number of drop-picks in which this event participates
	action	A_1	{drop (d), pick (p), f}	
V	clustO	A_0	$\mathbb{R}$	pixel overlap between the dropped and the picked bicycle-clusters
	pos	A_0	$\mathbb{Z}^4$	bounding box of the intersection area between the dropped and the picked bicycle-clusters
	psDDist	A_0	$\mathbb{R}$	post-segmented distance for the drop event
	psPDist	A_0	$\mathbb{R}$	post-segmented distance for the pick event
	psDEdges	A_0	$\mathbb{R}$	post-segmented edge ratio for the drop event
	psPEdges	A_0	$\mathbb{R}$	post-segmented edge ratio for the pick event
	action	A_1	{drop-pick (dp), drop-only (dx), pick-only (xp), f}	

Attribute Functions

$\psi_{dist}(x.traj, y.pos)$	calculates the spatial proximity between a person and a bicycle-cluster
$\psi_{co}(Z_1.fMap, Z_2.fMap)$	calculates the overlap in foreground map between the dropped and the picked bicycle-clusters
$\psi_{eR}(y.edgeR, y.pos)$	calculates the ratio of new to removed edges within a particular rectangular area

($x.au = y.au$). This rule is prevented from being applied by an inherited constraint ($x.count \neq 1$) when the person has already dropped/picked a bicycle.

We have not attempted to optimise the choice of attributes, and chose a plausible set of attributes to capture the various events and the essential constraints between them. For example, the size of the blob across the trajectory as the person passes through the racks is used to distinguish people dropping from those picking bicycles or simply passing through the racks. The probability for the

Table 3 AMG for the *Bicycles* problem: production rules

Production Rules (P)

	Syntactic Rule (r)			Attribute Rules (M)			Attribute Constraints (C)		
p_1	S	$\rightarrow$	$V^\star, x^\star, y^\star$	$y.action$	$=$	"$noise$"	$y.count$	$<$	1
				$x.action$	$=$	"$pass-by$"	$x.count$	$\neq$	1
p_2	V	$\rightarrow$	Z_1, Z_2	$V.action$	$=$	"$drop-pick$"	$Z_1.au$	$<$	$Z_2.au$
				$Z_1.action$	$=$	"$drop$"	$Z_1.count$	$\neq$	1
				$Z_2.action$	$=$	"$pick$"	$Z_2.count$	$\neq$	1
				$V.clustO$	$=$	ψ_{co} ($Z_1.fMap$, $Z_2.fMap$)			
				$V.pos$	$=$	$Z_1.pos \cap Z_2.pos$			
				$V.psDDist$	$=$	ψ_{dist} ($Z_1.traj$, $V.pos$)			
				$V.psPDist$	$=$	ψ_{dist} ($Z_2.traj$, $V.pos$)			
				$V.psDEdges$	$=$	ψ_{eR} ($Z_1.edgeR$, $V.pos$)			
				$V.psPEdges$	$=$	ψ_{eR} ($Z_2.edgeR$, $V.pos$)			
				$Z_1.count$	$=$	1			
				$Z_2.count$	$=$	1			
p_3	V	$\rightarrow$	Z, u	$V.action$	$=$	"$drop-only$"	$Z.count$	$\neq$	1
				$Z.action$	$=$	"$drop$"			
				$Z.count$	$=$	1			
				$V.pos$	$=$	$Z.pos$			
				$V.psDDist$	$=$	$Z.dist$			
				$V.psPDist$	$=$	1			
				$V.psDEdges$	$=$	$Z.edgeR$			
				$V.psPEdges$	$=$	1			
p_4	V	$\rightarrow$	u, Z	$V.action$	$=$	"$pick-only$"	$Z.count$	$\neq$	1
				$Z.action$	$=$	"$pick$"			
				$Z.count$	$=$	1			
				$V.pos$	$=$	$Z.pos$			
				$V.psDDist$	$=$	1			
				$V.psPDist$	$=$	$Z.dist$			
				$V.psDEdges$	$=$	1			
				$V.psPEdges$	$=$	$Z.edgeR$			
p_5	Z	$\rightarrow$	x, y	$x.action$	$=$	$Z.action$	$x.au$	$=$	$y.au$
				$y.action$	$=$	$Z.action$	$x.count$	$\neq$	1
				$Z.au$	$=$	$x.au$			
				$Z.traj$	$=$	$x.traj$			
				$Z.pos$	$=$	$y.pos$			
				$Z.edgeR$	$=$	$y.edgeR$			
				$Z.fMap$	$=$	$y.fMap$			
				$Z.dist$	$=$	ψ_{dist} ($x.traj$, $y.pos$)			
				$x.count$	$=$	1			
				$y.count$	$=$	$y.count + 1$			

presence or absence of a compound event is a function of the attribute values for that hypothetical event. For example, the occurrence of a drop-pick event V is evaluated using likelihood $p(o_V|V)$. This likelihood is defined as a pair of half-Gaussian distributions of the synthetic attribute $clustO = \psi_{co}(Z_1.fMap, Z_2.fMap)$, measuring the degree of overlap between a dropped bicycle-cluster in Z_1 and a picked bicycle-cluster in Z_2:

$$\psi_{co}(Z_1.fMap, Z_2.fMap)$$

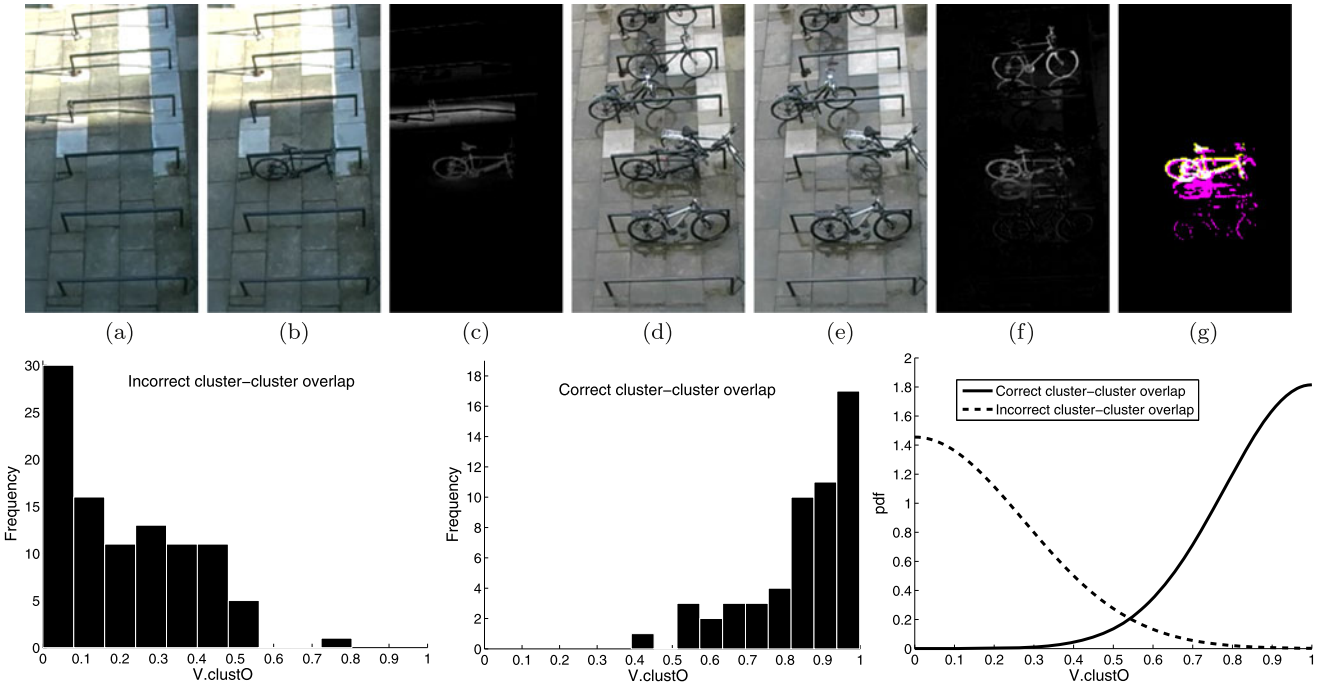

Fig. 10 (Colour online) Consecutive reference image pairs (**a**, **b**) and (**d**, **e**) are compared to reveal changes (**c**, **f**). By comparing the changed blobs (**g**), the clusters overlap $V.clustO$ is evaluated (7). Visually, *yellow* pixels represent the dropped clusters while *pink* pixels represent the picked cluster. Histograms of correct and incorrect values of *clustO* (from manual ground-truth) are shown along with MAP estimate for half-Gaussians

$$= \frac{M(Z_1.fMap \& Z_2.fMap)}{\min(M(Z_1.fMap), M(Z_2.fMap))} \tag{7}$$

Here $M(\cdot)$ returns the number of non-zero pixels in a given binary image, and the operator & is the pixelwise Boolean 'and'. The mean and standard deviation of the half-Gaussian distributions are the MAP estimates for the conditional probability of *clustO* values obtained from hand-labelled examples of true and false associations between drops and picks (Fig. 10).

Algorithm 1 is used to build the Bayesian network given the set of detections. The Boolean node 'u' is labelled true if some bicycles are deposited into the racks before the video sequence starts, and some could still be in the rack at the end of the sequence (open world assumption). Alternatively, if 'u' is labelled false, all drop and pick events are forced to be linked (closed world assumption). Figure 11 shows a parse tree of the AMG along with a labelled Bayesian network. Studying the AMG and the BN reveals exponential complexity in the number of nodes for the *Bicycles* problem.

Two bicycle rack locations have been chosen for testing. The first is within the University of Leeds campus, and the second outside Cambridge train station. The prior conditional probabilities are estimated without observing the testing data, and are kept fixed for all experiments. For each location, one hour is separately recorded and the events are manually labelled to estimate priors and likelihoods. This is

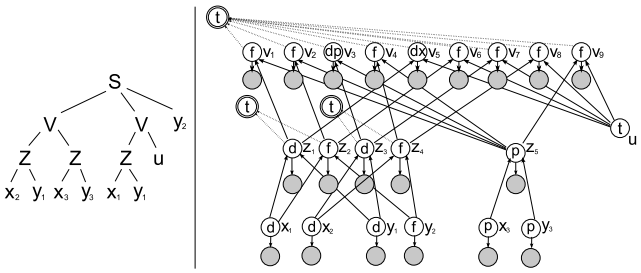

Fig. 11 A sample parse tree and the corresponding labelled BN

because automatic estimation requires a significant amount of data and is a computationally hard optimisation problem due to the dependencies between the production rules that arise from the constraints (Abney 1997). Table 4 contains a summary of statistics for both datasets. The MAP explanation is compared across all sequences for G, MHT, RJMCMC and IP searches (Table 5, Fig. 12).[1] IP finds the MAP explanation for all sequences, yet takes longer and requires more memory. RJMCMC achieved better results than MHT in 4 out of the 7 sequences, and comparable results in the re-

[1]Each RJMCMC chain executes within 3–7 minutes (3 GB) for all sequences of the 3 datasets (Bicycles problem and Entry-Exit problem). MHT executes within 20 minutes for $k = 500$ (4 GB). IP using XPRESS-MP takes 5–30 minutes for these sequences (10 GB). Note that the code was not optimised for performance comparison.

Table 4 Dataset statistics: training (T) and test sequences

	Leeds						Cambridge		
sequence	T	1	2	3	4	5	T	6	7
Duration	1 h	1 h	1 h	11 h	12 h	12 h	1 h	15 h	15 h
$\|\{x\}\|$	30	58	27	128	126	137	35	112	197
$\|\{y\}\|$	28	59	25	72	175	128	8	206	1847
Drops	15	24	11	20	20	14	6	28	39
Picks	13	20	12	19	20	13	4	17	41
Drop-Picks	11	20	11	18	20	13	4	14	22

Table 5 $-\log(p)$ compared across G, MHT, 40 runs ($n_{mc} = 5000$) of RJMCMC and RJMCMC-SA (linear cooling) and IP using XPRESS-MP. The results are not available for MHT ($k = 500$) on sequence 7 due to the implementation running out of memory.

	G	MHT			RJMCMC		RJMCMC-SA		IP
		$k = 50$	$k = 100$	$k = 500$	μ	σ	μ	σ	
1	102.25	58.78	58.78	57.86	57.90	0.11	57.86	0.00	57.86
2	23.54	4.64	4.64	4.64	4.64	0.00	4.64	0.00	4.64
3	609.66	493.18	468.80	468.80	429.30	3.23	423.98	2.36	416.64
4	6272.69	6149.95	6144.98	6144.30	6079.88	3.43	6078.40	3.23	6065.00
5	5034.46	4998.39	4982.86	4975.82	4943.71	3.59	4939.33	1.87	4937.08
6	860.37	812.96	812.96	812.96	814.71	1.69	811.50	2.36	797.29
7	934.36	608.92	607.39	–	451.92	9.29	433.50	7.76	283.51

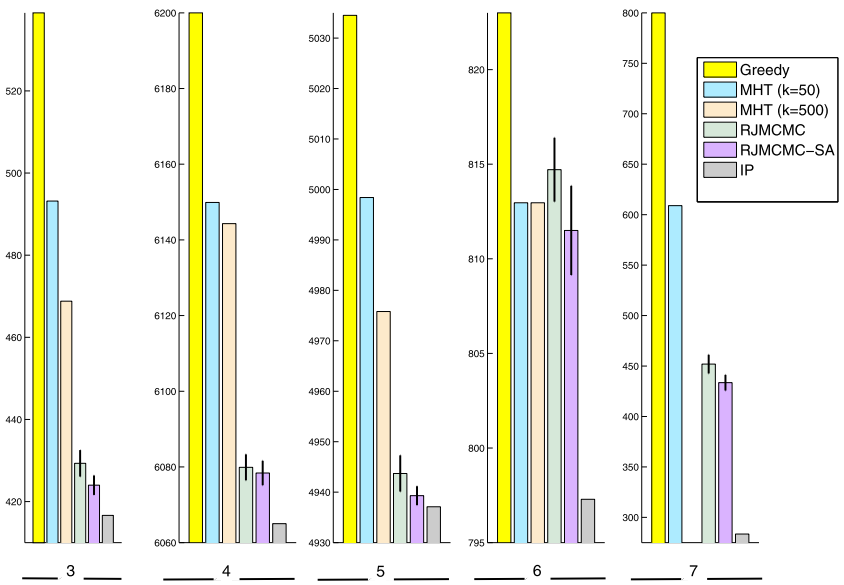

Fig. 12 $-\log(p)$ is compared for sequences (3–7) showing RJMCMC-SA achieves the best heuristic search results. The *vertical line* represents the standard deviation σ. The posterior found using MHT ($k = 50$) is vertically aligned for all sequences

maining sequences. RJMCMC-SA achieved the best results amongst heuristic methods.

When searching the global explanations using RJMCMC, the initial explanation ω_0 specifies that all people are passing by the rack area and all bicycle-clusters are noise. This is a valid explanation, though unlikely to be the MAP solution. At each step of the Markov chain, a move is applied to the current explanation. Figure 13 shows a sequence of moves.

The proposal distribution Q picks a move-type j_m then a specific move g_m. The weighted distribution j_m is estimated from the number of distinct moves of each type that can be applied to the current explanation ω_i. The type-specific distribution is dependent on the ambiguity in the data. For example, the ambiguity in connecting a person x_i to a bicycle y_j is calculated from the number of possible bicycle-clusters $B(x_i)$, and the number of people who came close to the bicycle-cluster $T(y_j)$. The weighting for selecting moves of type $connect_z$ is defined in (8).

$$\delta_{connect_z}(x_i) = \sum_{y_j \in B(x_i)} \frac{1}{|T(y_j)|} \tag{8}$$

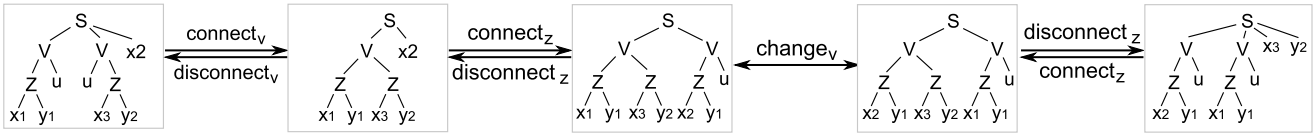

Fig. 13 A sequence of $\{connect_v \rightarrow connect_z \rightarrow change_v \rightarrow disconnect_z\}$ moves was applied. The last move affects both layers as disconnecting a pick cancels the drop-pick

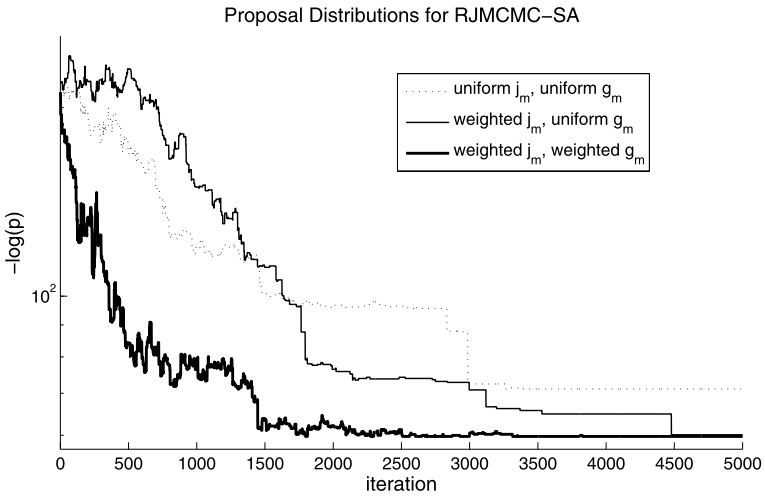

Fig. 14 Convergence under various proposal distribution choices using RJMCMC-SA for chains from the 4th sequence

Table 6 The accuracy results (%) for the MAP solutions. ⋆ denotes that for the same MAP, two or more explanations are found, and only the one with the maximum accuracy is recorded

	Local	G	MHT			RJMCMC		RJMCMC-SA		IP
			$k=50$	$k=100$	$k=500$	μ	σ	μ	σ	
1	74.13	72.41	91.38	91.38	91.38	88.36	1.09	87.46	1.79	91.38
2	85.19	85.19	100.00	100.00	100.00	100.00	0.00	100.00	0.00	100.00
3	64.06	58.59	84.38	84.38	84.38	87.68	0.89	83.36	1.65	87.5⋆
4	74.60	73.81	74.60	75.40	75.40	83.93	1.09	83.15	1.31	83.33⋆
5	86.13	89.05	82.48	84.67	88.32	91.90	0.79	92.65⋆	0.90	94.16
6	65.18	66.07	60.71	60.71	60.71	68.53	1.68	70.98	1.04	73.21
7	46.18	45.69	44.67	45.69	–	47.28	1.18	47.61	0.88	46.70

The type-specific distributions g_m for the remaining move types are explained in Damen (2009). This weighting is similar in spirit to the proposal priors used in DDMCMC (Tu and Zhu 2002), and speeds up convergence. Figure 14 shows an example of convergence for both RJMCMC and RJMCMC-SA chains under various choices of the proposal distribution. The first choice is when both the move type and the individual move are chosen uniformly-at-random (u.a.r). The chains are far from convergence in both cases. Alternatively, if the move type choices are weighted using estimated move counts, while the actual move within that type is selected u.a.r., the algorithm converges but requires a longer Markov chain. Weighted choices in both proposal distributions are capable of converging significantly faster.

The ground truth was manually obtained for each sequence, labelling each person with the event accomplished, then connecting picks to earlier drops. The accuracies for the MAP explanations from Table 5 are shown in Table 6. The last column in the table indicates the accuracy of the best global explanation. The global explanation does not match the ground truth when detections are missing altogether or feature values are incorrect. For example, when a bicycle-cluster is not found by the detector, a person is connected to an incorrect bicycle-cluster, or is thought to be passing by the bicycle rack. In the 7th sequence for example, the scene often changed from shadow to sunlight, and the bicycle-cluster detector often failed to correctly detect the changes in the background. The table also compares local and global analysis. A local solution is a complete but possibly inconsistent set of events, allowing the same drop to link to several pick events and vice versa. The results show higher accuracy for global explanations, as global explanations can resolve ambiguities that cannot be resolved by local analysis. Fig-

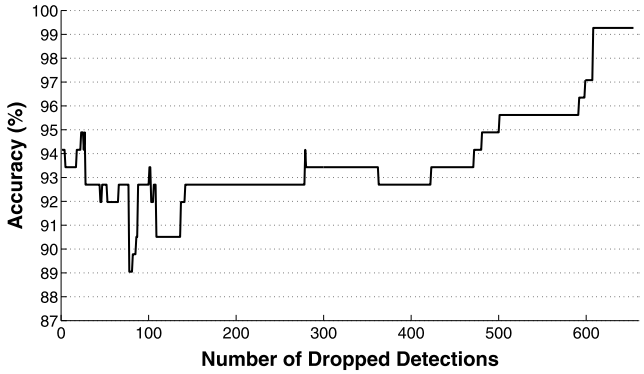

Fig. 15 Accuracy for the 5th sequence as more detections are missed

ure 15 plots the accuracy for the 5th sequence as detections are missed. The detections were dropped at random from the complete set of detections in the sequence, and the corresponding nodes are removed from the BN.

The best solution is then found using RJMCMC-SA, and the accuracy for the considered detections is calculated by comparing to the subset of the ground truth that corresponds to the remaining detections. The figure shows that dropping certain key detections results in a drop in the global explanation's accuracy. When just a few detections remain, the ambiguity is reduced and the accuracy increases. The method will not hallucinate missing detections or ignore false detections, and will simply seek a globally optimal explanation for the given set of detections.

The *Bicycles* problem reflects the complexity and high-level of interleaving that the framework introduced in this paper can deal with. The AMG rules could alternatively be directly mapped to string grammar rules where the ordering of terminals corresponds to the detection's start time with inference using multiple threads (Joo and Chellappa 2006a). This approach greedily assigns detections to open threads, and the performance would thus match the sub-optimal greedy search in Tables 5 and 6. In contrast, our framework is more likely to find the best global explanation. Using HHMM-JPDAF Nguyen et al. (2006), the low-level states of the HHMM would correspond to primitive events, while the states at higher levels would correspond to complex events. For a given number of HHMMs, JPDAF can assign detections to the different HHMMs. The approach from (Nguyen et al. 2006) though requires two modifications to handle the *Bicycles* problem: the data association should allow the same event to participate in multiple complex events, and the number of HHMM should be varied so the best number of events is found. It is not obvious how these modifications can be introduced. Finding the optimal number of HHMMs is a complex optimisation problem in itself.

6.2 The Entry-Exit Problem

This section presents a different problem that requires tracking people, and any objects they might be carrying, as they enter and exit a building. A global explanation links the person entering the building, possibly with some carried objects, to a later departure of a person, with or without carried objects. It also can link the departing person to their return later. The linking depends on comparing the person and the baggage biometrics between both appearances. Natural constraints govern the possible explanations, e.g. a person entering the building can be observed departing only once, and at a later point in time. This problem is similar to the task of tracking people between non-overlapping cameras, yet the person is not restricted to emerge again within a certain amount of time, which increases the number of interleaved events making the explanations intractable in most cases. As before, pedestrian trajectories are detected using the same off-the-shelf tracker (Magee 2002). For each trajectory, protrusions representing candidate carried objects are retrieved using the method in Damen and Hogg (2008).

Similar to the *Bicycles* problem, an AMG is designed and some features are selected (details available in Damen 2009). Simple features were again chosen; people tracked in and out of the building were matched by their projected height and clothing colour. Carried objects were compared by their colour and position relative to the body. Testing was performed on 12 hours of video recorded outside a building entrance. 326 trajectories close to the entrance were detected after manually rejecting groups of people walking together. The baggage detector from Damen and Hogg (2008) resulted in 429 candidate bags. The BN obtained from these detections contains 190849 event-nodes. Table 7 compares the MAP for the BN. The IP solver could not exhaustively search the space of explanations in reasonable time (using 20 GB of memory for about 10 hours) as the constraints in this problem are more complex than those in the *Bicycles* problem. In the *entry-exit* activity, the enter event can be linked to an earlier exit as well as a later one. Conflict checking (Sect. 5) is thus required, which considerably increases the number of constraints to be satisfied by the solver. For a smaller-scale problem, the table shows the MAP solution for the first 25 people (out of 326 in the dataset) and their corresponding candidate bags. RJMCMC-SA is once again the best heuristic search technique. It's the only technique that was able to find the MAP explanation (at some chains).

When compared to ground-truth data for the 326 trajectories, the global explanation achieves a recall of 30%, yet a precision of only 12%. This is because the features used to link events are weakly discriminative. A high number of false links originate from people of similar height and clothing colour. This performance can almost certainly be improved by using other features developed to solve the person

Table 7 $-\log(p)$ compared across search techniques for 25 people detections as well as for all detections. Results were not available for the 326 people using MHT $k > 50$ (implementation running out of memory) or IP (due to the increase in the number of constraints)

	G	MHT			RJMCMC		RJMCMC-SA		IP
		$k = 1$	$k = 20$	$k = 500$	μ	σ	μ	σ	
25 people	85.61	85.49	84.97	84.47	85.55	0.13	84.29	0.03	84.27
326 people	1143.47	1146.58	1137.70	–	1143.09	0.40	1123.02	1.12	–

Fig. 16 Correctly associated detections when global explanations are considered

re-identification problem. Figure 16 shows three sequences that were correctly retrieved only when the global explanation is searched using RJMCMC-SA. The figure shows the framework's ability to correctly discover an 'exit-enter-exit-enter' sequence.

7 Conclusion

This paper proposes a framework for finding a consistent set of events that covers all detections, referred to as a global explanation. Using a Bayesian approach, the Maximum a Posteriori (MAP) explanation is selected as the best explanation. In achieving the task, the activity and its constraints are described using Attribute Multiset Grammars (AMG). Each production rule in the grammar rewrites a nonterminal into an un-sequenced collection of simpler events (i.e. a multiset). The rule is associated with attribute rules and attribute constraints that define the allowable compositions of events in line with the domain's constraints.

For each input video, the detections become the terminal symbols of the AMG with the associated synthetic attribute values. These detection, together with an AMG, determine a Bayesian Network (BN) that models the probability distribution over the set of global explanations. Approximate search techniques are proposed to find the MAP, as a combinatorial search becomes intractable when the complexity and duration of the activity increases. The approach was tested on two case studies. Results show that for the second case study, the number of constraints increases significantly and the exhaustive approach indeed fails to deliver the best global explanation. In these results, RJMCMC along with Simulated Annealing is the best heuristic search technique, and is scalable as the complexity increases.

Like some earlier work (Nguyen et al. 2006; Lin et al. 2009), our framework is intended to provide globally consistent explanations for activities involving interleaved events and uncertainty in detections obtained from video. The use of MCMC in finding the optimal explanation is intended for problems that are intractable using deterministic methods. The ability to group events at different levels of granularity within a compositional hierarchy makes the search for an optimal explanation potentially faster than it would be when grouping only takes place at the lowest level. This can be exploited by MCMC through moves that operate on events at the different compositional levels (e.g. drops, and drop-picks).

The AMG representation is well suited to situations in which there is a natural compositional structure to the events that make up an activity, and any temporal ordering of constituents at each compositional level is only loosely defined. Where there is a rigid temporal ordering of events, the overhead associated with using attributes to represent this ordering will obfuscate the model and it would be better to use an attribute string grammar where ordering is an intrinsic feature. Thus for example, cooking scenarios where the order in which things are done may be highly variable could be well suited to an AMG, whereas assembly steps on a production line would not be as well represented. The framework should scale to events involving more than two objects, and indeed the *Bicycles* problem involves a compound event containing four entities: two instances of the same person at different times and two bicycle clusters. The complexity that this involves is not in the AMG, but in the optimisation of event structures, where we have demonstrated good scalability of the stochastic procedures.

In the current framework, the AMG is manually built for each activity. This includes building the hierarchical structures, deciding on the features (attributes) appropriate to the different event types, and listing the constraints. Ideally, it would be possible to build an AMG automatically from sample videos depicting a target activity, although caution is needed in relying only on statistical learning (Zhu and Mumford 2006), ignoring the overall objectives of the designer. For grammatical representations, there has been some progress on grammar induction from examples (de la

Higuera 2005), and the representation of possible sequences of symbols using suffix trees learnt from examples (Hamid et al. 2007). There has also been prior work on the induction of And-Or graphs representing activities from weakly-labelled data. For example, textual annotations were used in Gupta et al. (2009) to build an initial And-Or graph with actions at the nodes. As new data is parsed, the graph is modified and extended to best accommodate the new data, balanced by the complexity of the resulting model. Spatio-temporal relationships governing interest points have been automatically mined from action videos (Wang et al. 2011). In this work, depending on the underlying data, the relationship's strength is specified as one of three possible levels: strong, weak or stochastic. Discriminative rules are pursued given negative examples from other types of actions. Results show that discriminative hierarchical rules outperform traditional classification on known action datasets. We are not aware of prior work on the induction of attribute grammars from training data. One can envisage learning cardinality constraints on configurations of events (e.g. no more than four people can get into a car), and deriving probability distributions over common attributes, relating to spatial (and temporal) extent and appearance (e.g. colour histograms), that have been propagated up the compositional hierarchy of events. Previous work on feature selection in machine learning may also be relevant here, particularly when applied to activity recognition (e.g. Riberio and Santos-Victor 2005; Smith et al. 2005).

Appendix: Derivation

$$p(\omega|Y) = \frac{1}{\mathcal{G}} \prod_i p(o_{b_i}|b_i)p(b_i) \prod_j p(o_{c_j}|c_j)p(c_j) \times \prod_{ij} p(o_{B_{ij}}|B_{ij})p(B_{ij}|b_i, c_j)p(\mathbf{c}|\{B_{ij}\}) \quad (9)$$

Using Bayes, the first product can be substituted $p(b_i|o_{b_i}) = p(o_{b_i}|b_i)p(b_i)/p(o_{b_i})$. The denominator is a constant that can be part of the normalizing factor $\mathcal{G}$. Similarly for the other terms. The posterior (9) can be re-arranged as

$$p(\omega|Y) = \frac{1}{\mathcal{Z}} \prod_i p(b_i|o_{c_i}) \prod_j p(c_j|o_{c_j}) \times \prod_{ij} p(B_{ij}|b_i, c_j, o_{B_{ij}})p(\mathbf{c}|\{B_{ij}\}) \quad (10)$$

The third factor in (10) becomes intractable to compute as the number of detections increases. Fortunately, this can be avoided by computing a proportional quantity instead ($p(B_{ij}|b_i, c_j, o_{B_{ij}})$ is abbreviated to $p(B_i|\cdot)$ in the derivation).

$$\prod_i p(B_i|\cdot) = \prod_{i:B_i=f} p(B_i = f|\cdot) \prod_{i:B_i=t} p(B_i = t|\cdot) \quad (11)$$

$$= \prod_{i:B_i=f} p(B_i = f|\cdot) \times \prod_{i:B_i=t} p(B_i = t|\cdot) \frac{\prod_{i:B_i=t} p(B_i = f|\cdot)}{\prod_{i:B_i=t} p(B_i = f|\cdot)} \quad (12)$$

$$= \prod_i p(B_i = f|\cdot) \prod_{i:B_i=t} \frac{p(B_i = t|\cdot)}{p(B_i = f|\cdot)} \quad (13)$$

$$\propto \prod_{i:B_i=t} \frac{p(B_i = t|\cdot)}{p(B_i = f|\cdot)} \quad (14)$$

This derivation specifically enables finding a quantity, proportional to the original posterior, that is independent of all false-labelled nodes. The posterior $p(\omega|Y)$ is rewritten to be

$$p(\omega|Y) = \frac{1}{\mathcal{Q}} \prod_i p(b_i|o_{b_i}) \prod_j p(c_j|o_{c_j}) \times \prod_{ij:B_{ij}=t} \frac{p(B_{ij} = t|b_i, c_j, o_{B_{ij}})}{p(B_{ij} = f|b_i, c_j, o_{B_{ij}})} \prod_{ij} p(\mathbf{c}|\{B_{ij}\}) \quad (15)$$

Acknowledgements We would like to thank Les Proll for assisting with formulating and solving the integer programming optimisation problem. This work was supported in part by the EPSRC (Project LAVID, EP/D061334/1).

References

Abney, S. P. (1997). Stochastic attribute-value grammars. *Computational Linguistics*, *23*(4), 597–618.

Aho, A., Sethi, R., & Ulman, J. (1986). *Compilers: principles, techniques and tools*. Reading: Addison-Wesley.

Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning*, *50*, 5–43.

Blevins, J. (2001). Feature-based grammar. In R. Borsley & K. Borjars (Eds.), *Nontransformational Syntax: A Guide to Current Models* New York: Wiley-Blackwell.

Damen, D. (2009). *Activity analysis: finding explanations for sets of events*. PhD thesis, University of Leeds, UK.

Damen, D., & Hogg, D. (2008). Detecting carried objects in short video sequences. In *Proc. European computer vision conference (ECCV)*.

Damen, D., & Hogg, D. (2009a). Recognizing linked events: Searching the space of feasible explanations. In *Proc. computer vision and pattern recognition (CVPR)*.

Damen, D., & Hogg, D. (2009b). Attribute multiset grammars for global explanations of activities. In *Proc. British machine vision conference (BMVC)*.

de la Higuera, C. (2005). A bibliographical study of grammatical inference. *Pattern Recognition*, *38*, 1332–1348.

Fan, Q., Bobbitt, R., Zhai, Y., Yanagawa, A., Pankanti, S., & Hampapur, A. (2009). Recognition of repetitive sequential human activity. In *Proc. computer vision and pattern recognition (CVPR)*.

Felzenszwalb, P., & Huttenlocher, D. (2000). Efficient matching of pictorial structures. In *Proc. computer vision and pattern recognition (CVPR)*.

FICO, D. O. (2007). *XPRESS-MP solver—version 19.00.17*.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*(6), 721–741.

Gollin, E. (1991). *A method for the specification and parsing of visual languages*. PhD thesis, Brown University.

Gong, S., & Xiang, T. (2003). Recognition of group activities using dynamic probabilistic networks. In *Proc. international conference on computer vision (ICCV)*.

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, *82*(4), 711–732.

Green, P. (2003). Trans-dimensional Markov chain Monte Carlo. In P. Green, N. Lid Hjort, & S. Richardson (Eds.), *Highly structured stochastic systems*. Oxford: Oxford University Press.

Gupta, A., Srinivasan, P., Shi, J., & Davis, L. (2009). Learning a visually grounded storyline model from annotated videos. In *Proc. computer vision and pattern recognition (CVPR)*.

Hamid, R., Maddi, S., Bobick, A., & Essa, M. (2007). Structure from statistics—unsupervised activity analysis using suffix trees. In *Proc. int. conf. on computer vision (ICCV)*.

Han, F., & Zhu, S. (2005). Bottom-up/top-down image parsing by attribute graph grammar. In *International conference on computer vision (ICCV)* (Vol. 2, pp. 1778–1785).

Hongeng, S., Nevatia, R., & Bremond, F. (2004). Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, *96*(2), 129–162.

Huang, T., & Russell, S. (1998). Object identification: A Bayesian analysis with application to traffic surveillance. *Artificial Intelligence*, *103*(1-2), 77–93.

Intille, S., & Bobick, A. (2001). Recognizing planned, multiperson action. *Computer Vision and Image Understanding*, *81*(3), 414–445.

Ivanov, Y., & Bobick, A. (2000). Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 852–872.

Joo, S.-W., & Chellappa, R. (2006a). Attribute grammar-based event recognition and anomaly detection. In *Computer vision and pattern recognition workshop (CVPRW)*.

Joo, S.-W., & Chellappa, R. (2006b). Recognition of multi-object events using attribute grammars. In *Proc. int. conf. on image processing (ICIP)* (pp. 2897–2900).

Kastens, U. (1980). Ordered attributed grammars. *Acta Informatica*, *13*, 229–256.

Kitani, K. M., Sato, Y., & Sugimoto, A. (2005). Deleted interpolation using a hierarchical Bayesian grammar network for recognizing human activity. In *Workshop on visual surveillance and performance evaluation of tracking and surveillance (PETS)*.

Knuth, D. (1968). Semantics of context-free languages. *Mathematical Systems Theory*, *2*(2).

Lin, L., Gong, H., Li, L., & Wang, L. (2009). Semantic event representation and recognition using syntactic attribute graph grammar. *Pattern Recognition Letters*, *30*(2), 180–186.

Magee, D. (2002). Tracking multiple vehicles using foreground, background and motion models. In *Proc. workshop on statistical methods in video processing* (pp. 7–12).

Morefield, C. (1977). Application of 0-1 integer programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, *22*(3), 302–312.

Nevatia, R., Zhao, T., & Hongeng, S. (2003). Hierarchical language-based representation of events in video streams. In *Proc. of IEEE workshop on event mining (EVENT)*.

Nguyen, N., Venkatesh, S., & Bui, H. (2006). Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association. In *Proc. British machine vision conference (BMVC)*.

Nilsson, N. (1971). *Problem-solving methods in artificial intelligence*. New York: McGraw-Hill.

Oh, S., Russell, S., & Sastry, S. (2004). Markov chain Monte Carlo data association for general multiple-target tracking problems. In *43rd IEEE Conference on Decision and Control (CDC)* (Vol. 1, pp. 735–742).

Reid, D. (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, *24*(6), 843–854.

Riberio, P., & Santos-Victor, J. (2005). Human activity recognition from video: modeling, feature selection and classification architecture. In *Intl. workshop on human activity recognition and modelling*.

Rota, M., & Thonnat, M. (2000). Video sequence interpretation for visual surveillance. In *IEEE int. workshop on visual surveillance (VS)*, Dublin, Ireland.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut—interactive foreground extraction using iterated graph cuts. In *ACM trans. on graphics (SIGGRAPH)*.

Shi, Y., Huang, Y., Minnen, D., Bobick, A., & Essa, I. (2004). Propagation networks for recognition of partially ordered sequential action. In *Proc. computer vision and pattern recognition (CVPR)*.

Siskind, J. (2000). Visual event classification via force dynamics. In *Association for the advancement of artificial intelligence (AAAI)* (pp. 149–155).

Smith, K. (2007). *Bayesian methods for visual multi-object tracking with applications to human activity recognition*. PhD thesis, Ecole Polytechnique Federale de Lausanne (EPFL).

Smith, P., Lobo, N. Vitoria, & Shah, M. (2005). Temporalboost for event recognition. In *Proc. international conference on computer vision (ICCV)*.

Tran, S., & Davis, L. (2008). Event modeling and recognition using Markov logic networks. In *Proc. European conference on computer vision (ECCV)*.

Tu, Z., & Zhu, S.-C. (2002). Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 657–673.

Wang, L., Wang, Y., & Gao, W. (2011). Mining layered grammar rules for action recognition. *International Journal of Computer Vision*, *93*(2), 162–182.

Williams, H. (1999). *Model Building in Mathematical Programming* (4th edn.). New York: Wiley.

Wu, Y., & Huang, T. (2004). Robust visual tracking by integrating multiple cues based on co-inference learning. *International Journal of Computer Vision*, *58*(1), 55–71.

Wu, B., & Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *Proc. international conference on computer vision (ICCV)*.

Young, R., Kittler, J., & Matas, J. (1998). Hypothesis selection for scene interpretation using grammatical models of scene evolution. In *Int. conf. on pattern recognition*.

Yu, Q., Medioni, G., & Cohen, I. (2007). Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. In *Proc. computer vision and pattern recognition (CVPR)*.

Zhao, T., & Nevatia, R. (2004). Tracking multiple humans in crowded environment. In *Proc. computer vision and pattern recognition (CVPR)*.

Zhu, S.-C., & Mumford, D. (2006). A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, *2*(4), 259–362.