Noname manuscript No.
(will be inserted by the editor)

Explaining Activities as Consistent Groups of Events
A Bayesian Framework using Attribute Multiset Grammars

Dima Damen · David Hogg

7 August 2010

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). AMG combines the event hierarchy with the necessary features for recognition and all natural constraints. Parsing a set of detections by such a grammar finds a consistent set of events that satisfies the activity's natural 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 (BN). 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

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, extending over a long temporal duration. These dependencies can be exploited to disambiguate uncertain visual data by seeking a global 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 [7, 8].

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.

Dima Damen
Department of Computer Science , University of Bristol, E-mail: damen@cs.bris.ac.uk

David Hogg
School of Computing, University of Leeds, E-mail: d.c.hogg@leeds.ac.uk

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. The search for the Maximum a Posteriori (MAP) solution is performed using heuristic and exhaustive techniques in Section 5. Finally, Section 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 example in detecting motion [9, 26, 34, 39], cars [23] and pedestrians [35]. 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, that 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 hierarchical consistent grouping, where 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 implictly 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,

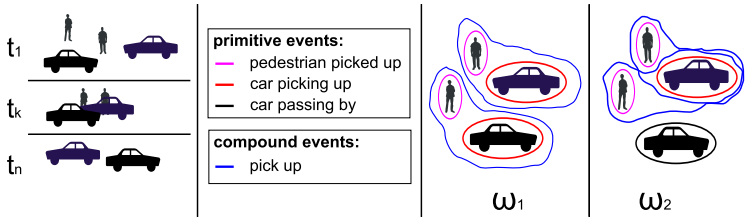

Fig. 1 For the same set of detections, and given primitive and compound events, two different global explanations ω_1 and ω_2 are shown, where each boundary corresponds to an event. In ω_1, each car picks up one pedestrian, while in ω_2 the blue car picks up both pedestrians.

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 the activity's events and its natural constraints. This framework finds the best global explanation for all detections in a video input. 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 tasks to be performed once for each considered activity. The hierarchy and the natural constraints are employed to create an Attribute Multiset Grammar (AMG). This process is manual, and the AMG is used, along with labeled 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, which represents terminal symbols, along with assigning values to the selected features. 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.

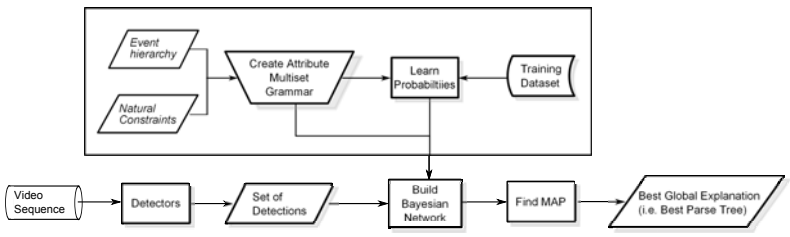

Fig. 2 A flowchart indicating the proposed framework.

2 Background Review

Simultaneous analysis of all detections has proven advantageous in many areas of computer vision, such as image denoising [13], segmentation [40] and object recognition [11,47]. 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 [49]. 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 [2].

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 [25]. 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 [22, 24, 34, 39, 41, 42].

In recent work, Tran and Davis [45] 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, and have weights to imply rule preferences; 'disjunctions' which provide alternative decompositions; 'negative preconditions' which are constraints on applying the rules; and 'exclusion relations' which 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 orignally proposed to extract semantics from the compositional structure of a parse tree [30] 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 [21]. 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 [52]. Attribute grammars have been recently used to recognise activities in a car park [27, 31], although these approaches do not employ the full capabilities of attribute grammars, as they do not use inherited attributes or inherited constraints (explained in Section 3).

2.2 Recognising Activities

The activity's representation is then used to recognise events from video input. Single event recognition uses graphical models like hidden Markov models (HMMs) [25, 35] and Bayesian Networks [24, 29], and partitioning detections into events uses Markov Random Fields (MRF) [31] or data association techniques [35, 43].

In [25], 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. Shi *et al.* use discrete Condensation to sample the space of explanations [41]. This outperforms the parsing from [25] in recovering from errors and uncertainties in the data.

Kitani *et al.* build a hierarchical Bayesian network from an SCFG [29]. 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 window of size l. A solution that better explains recent observations is favoured. Intille and Bobick also build a Bayesian network and represent each event by a ternary observed node (yes/maybe/no) [24]. 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.

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 [10, 26] assign detections to events greedily in a sequential order during recognition. Nguyen *et al.* [35] 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 expects 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 [31]. 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.

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) [23, 37] and sampling the distribution of associations using importance sam-

pling [48] or MCMC [36, 43, 50, 51]. Smith [43] 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 [33].

3 Defining Global Explanations of Activities

Attribute Grammars as first introduced by Knuth [30], also referred to as Feature-Based Grammars [4] and Attribute-Value Grammars [1], 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 [14] 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 the order is irrelevant although each symbol can still appear more than once. AMGs generalise attribute grammars by removing the sequential ordering of symbols in a sentence, requiring only a multiset of symbols. Thus, the same terminal symbol, representing a particular graphical component for example, may appear more than once. We use the formalism from [14] in the rest of the paper. This is adaptated from Knuth's original terminology [30].

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 their parents.

Each production rule $p \in P$ is a three-tuple (r, M, C) where r is a *syntactic rule* of the form $X_0 \rightarrow X_1, X_2, ..., X_{n_p}$ that rewrites the nonterminal X_0 as a multiset of nonterminal and terminal symbols $X_1, X_2, ..., X_{n_p}$. 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 distinguising features, originating from the detections.
- Inherited attributes (A_1) are explanation-related attributes, like the number of people picked up by one car (Figure 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 AGs in general, assumptions are normally made about the order in which attributes are computed, assuming such an ordering exists and there is no circularity [28]. 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-bottom 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 detections 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.

For readers familiar with And-Or Graphs, it is worth noting that this is an equivalent representation to context-free grammars [19]. Figure 4 shows the sample AMG

Terminals (T):	a, b, c	primitive events
Nonterimanls (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

Table 1 AMG example G_1

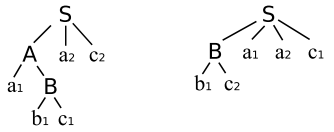

Fig. 3 Two parse trees given a set of detections and AMG G_1.

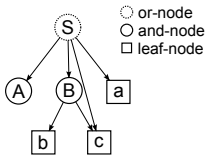

Fig. 4 And-Or graph representation of the grammar G_1.

from Table 1 represented as an And-Or graph using the notation from [52]. Notice that And-Or graphs are usually drawn for string-grammars, where the order of children (left-to-right) represents the order of symbols in the production rule. For multiset grammars, this order is not preserved. We have chosen to represent the grammar in a more traditional way - using tables of syntactic rules - as it clarifies the correspondence between syntactic rules, attribute rules and attribute 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 a 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 constraint operates. These are deterministic variables in the BN (denoted

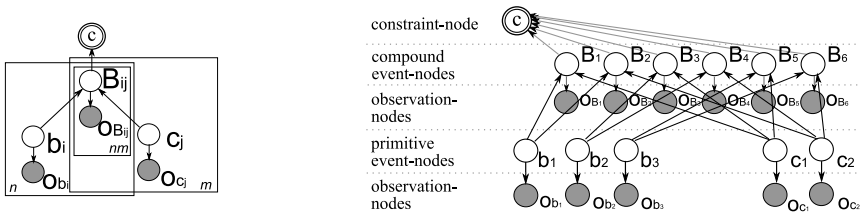

Fig. 5 A plate and unrolled BN for the simple AMG in Table 1, restricted to the single rule $p_3 : B \rightarrow b, c$.

by double-circled nodes), as each is functionally depend on the values of its parents using a Boolean function. The constraint-node evaluates to true if and only if the corresponding constraint specified in the AMG is satisfied. This implies that the joint probability of the BN is zero if any constraint is broken.

To illustrate, Figure 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.

```
   input    : Grammar G = (N, T, S, A, P), detections set D
   output   : Bayesian network structure BN
 1 %%% Build Bayesian network structure
 2 initialise an empty Bayesian Network (BN)
 3 foreach terminal instance t ∈ D
 4     add event-node to BN of type t
 5     if t has synthetic attributes then
 6         add a related observation-node to hold the synthetic attribute values
 7 order rules P starting with those containing terminals then bottom-up
 8 foreach rule p ∈ P (p.r : X_0 → X_1, X_2, ..., X_n); X_0 ≠ S
 9     Let I(X_i) be the set of event-nodes in BN of type X_i
10     comb = I(X_1) × I(X_2) × ... × I(X_n)
11     while comb ≠ φ do
12         multiset b = comb(1) - first multiset in comb
13         comb = comb - b
14         if b satisfies synthetic attribute constraints p.C_0 then
15             add event-node R to the BN of type X_0
16             foreach synthetic attribute rule m ∈ p.M_0
17                 apply m assigning a synthetic attribute value to observation-node of X_0
18             all event-nodes in the multiset b parent the created event-node
19             if recursive rule p then
20                 Let A(b,X_i) be the set of all ancestors of b of type X_i
21                 comb_2 = {I(X_1) − A(b, X_1)} × .. × R × .. × {I(X_n) − A(b, X_n)}
22                 comb = comb ∪ comb_2
23 %%% Find inter-dependent nodes
24 Let Nodes_n be the set of all event-nodes
25 while Nodes_n ≠ φ do
26     find Nodes_p with inherited constraints limiting the same inherited attribute values
27     Nodes_n = Nodes_n - Nodes_p
28     if size of Nodes_p > 1 then
29         add constraint-node c to hold the inherited constraints
30         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.

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. 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 non-terminal at the left-hand-side of the production rule. To accommodate for *direct recursion* in grammars, the if-statement (line 20) checks for new possible multisets of event-nodes in the BN. The algorithm cannot deal with indirect recursion. This is not seen as a limitation to defining activities, because direct recursion is sufficient to define repetitive patterns in the grammar. 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 specified detection multiset along with two labellings that reflect the parse trees in Figure 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 from one production rule in Figure 5, and set of detections $\{b_i\}, \{c_j\}$, the posterior is written as

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

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

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

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 various labellings of the Bayesian network.

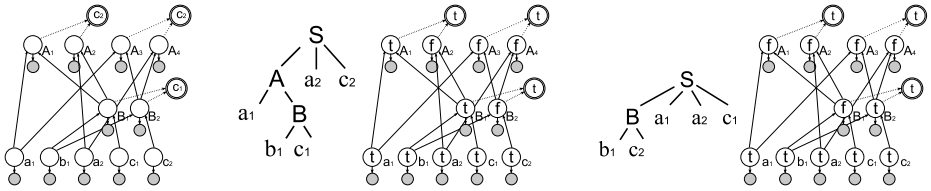

Fig. 6 The Bayesian network for the grammar G_1 along with two labellings that reflect the parse trees in Figure 3. An event-node is labeled true if the event appears in the parse tree.

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 (MLEN). A MLEN can be labelled with one of 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'. MLENs are used in the AMG for the *Bicycles* problem in Section 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). One method is guaranteed globally optimal which is formulating 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 (Section 6). This section explains how each of these search techniques searches the BN built in Section 4.

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},

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

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 labeled true, unless the explanation becomes inconsistent. The evaluation continues up the hierarchy until all nodes are labeled.

Multiple Hypotheses Tree (MHT) [37], 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, 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 $\pi(\omega)$ 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 [3]. The space of possible explanations is a discrete space, thus moves are designed to change a certain explanation ω into a slightly different one, preserving the constraints. Green suggested using Reversible Jump MCMC for sampling the joint distribution of both the model dimension and the model parameters [16]. By analogy, given a set of detections, the search is for the number of events and which detections belong to each event. RJMCMC

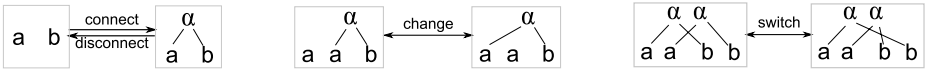

Fig. 7 Four move types to link events, break links, change linked events and switch links.

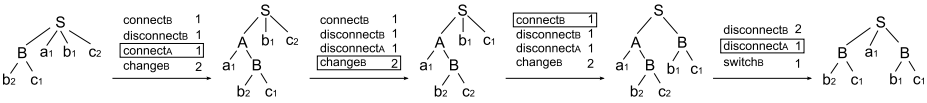

Fig. 8 Four moves are applied in sequence. The label at each arrow shows the number of possible moves of each type. The rectangle indicates the chosen move type.

generalises the acceptance probability to include the probability of selecting the move type, and a move-specific probability [17].

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

In Equation 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 Equation 5 is the absolute determinant of the Jacobian matrix of this diffeomorphism, which equals the identity matrix for the moves proposed here (Refer to Smith [43] 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 (Figure 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, Figure 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 labeled BN. Recall that there is a one-to-one mapping between a labeled 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 Section 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 F 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:

$\lambda_0 : a_1$	$\lambda_4 : B_1, b_1, c_1$	$\lambda_8 : A_3, a_1, B_2, b_1, c_2$
$\lambda_1 : a_2$	$\lambda_5 : B_2, b_1, c_2$	$\lambda_9 : A_4, a_2, B_2, b_1, c_2$
$\lambda_2 : c_1$	$\lambda_6 : A_1, a_1, B_1, b_1, c_1$	
$\lambda_3 : c_2$	$\lambda_7 : A_2, a_2, B_1, b_1, c_1$	

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 Figure 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
$\tau\omega \geq \mathbf{1}$, and
$\theta\omega \leq \mathbf{1}$, 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 [46]. We use XPRESS-MP to solve the standard linear optimisation [12]. 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

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

drops when the corresponding events are both observed. Due to the highly-interleaved activities, previous approaches like string grammars or HMMs result in overly complex representation. For three interleaved drop and pick events the number of rules (or states) required equals 10, and the increase is exponential. For these types of interleaved events, multiset grammars introduce a significant simplification in representation over available approaches. 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.

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 [15] for plane refueling 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 [32]. 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 [5].

The AMG for the *Bicycles* problem, using the notation from Section 3, is given in Tables 2 and 3. Simple features have been used to recognise the events at the different levels of the hierarchy. In this work, we have not attempted to find the best feature(s) for recognising the events, as we focus on the global explanation. 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 presence or absence of a compound event is a function of the attribute values for that hypothetical event. For example, the likelihood $p(o_V|V)$ 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-

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
Nonterimanls (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

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

cluster in Z_1 and a picked bicycle-cluster in Z_2:

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

Here M(·) 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 (Figure 10).

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_2). In p_5 drop and pick events between people and bicycle-clusters should be detected within the same activity unit ($x.au = y.au$ in p_5). An inherited constraint expects that each trajectory passing through the lab can drop/pick only one bicycle ($x.count \neq 1$).

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			

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

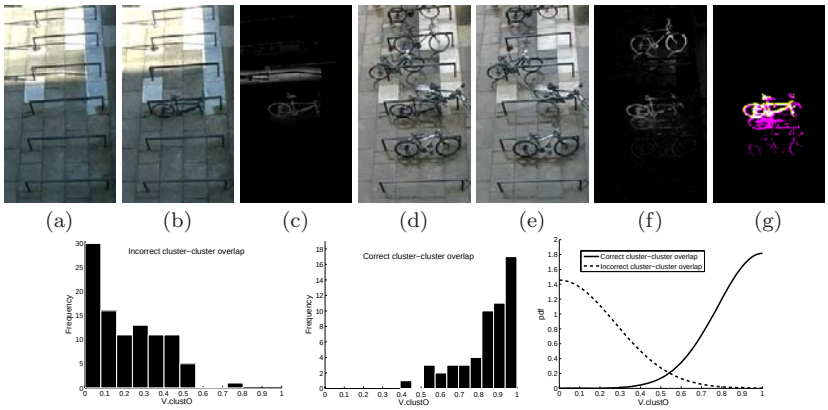

Fig. 10 Consecutive reference image pairs (a,b) and (d,e) are compared to reveal changes (c,f). By comparing the changed blobs (g), the clusters overlap *V.clustO* is evaluated (Equation 7). Visually, yellow pixels represent the dropped clusters while pink pixels represent the picked cluster. Correct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

Algorithm 1 is used to build the Bayesian network given the set of detections. The Boolean node 'u' is labeled true if an open world assumption [1] is considered.

[1] An open world implies that 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.

Alternatively, if 'u' is labeled false, all drop and pick events are forced to be linked and the world is assumed closed. Figure 11 shows a parse tree of the AMG along with a labeled Bayesian network. Studying the AMG and the BN reveals exponential complexity in the number of nodes for the *Bicycles* problem.

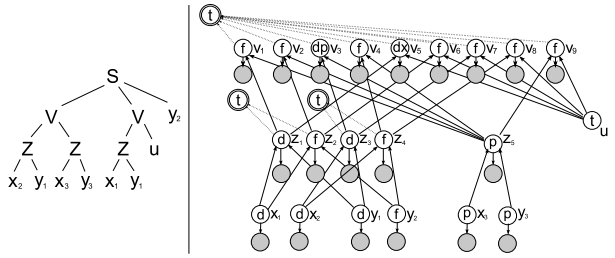

Fig. 11 A sample parse tree and the corresponding labelled BN.

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 12 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 come close to the bicycle-cluster $T(y_j)$. The weighting for selecting moves of type $connect_z$ is defined in Equation 8.

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

The type-specific distributions g_m for the remaining move types are explained in [5].

The prior conditional probabilities are manually estimated without observing the testing data, and are kept fixed for all experiments. This is because estimating them from training data 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 [1].

Two bicycle rack locations have been chosen for testing. The first is within the University of Leeds, and the second outside Cambridge train station. Table 4 contains a summary of statistics for both datasets. The MAP explanation is compared across all

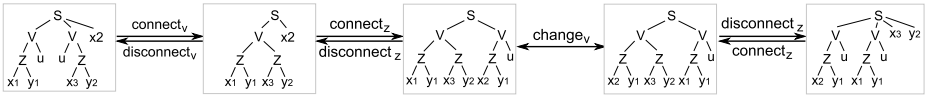

Fig. 12 A sequence of $\{\text{connect}_v \rightarrow \text{connect}_z \rightarrow \text{change}_v \rightarrow \text{disconnect}_z\}$ moves was applied. The last move affects both layers as disconnecting a pick cancels the drop-pick.

sequences for G, MHT, RJMCMC and IP searches (Table 5, Figure 13) [2]. 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 remaining sequences. RJMCMC-SA achieved the best results amongst heuristic methods.

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

Table 4 Dataset statistics

	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

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.

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.

Figure 15 shows how the optimisation changes as more detections are being considered. We process the detections in their temporal order, so at each point in time the output is a valid explanation given the detections from the start of the video sequence up to that point in time. The figure shows that some detections introduce higher ambiguity in the global explanation, while the others resolve ambiguities by increasing the MAP (decrease in -log(p)). It should though be noted that the figure does not take the normalising factor in the posterior into consideration.

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,

[2] Each RJMCMC chain executes within 3-7 minutes (3GB). MHT executes within 20 minutes for $k = 500$ (4GB). IP using XPRESS-MP takes 5-30 minutes for these sequences (10GB). Note that the code was not optimised for performance comparison.

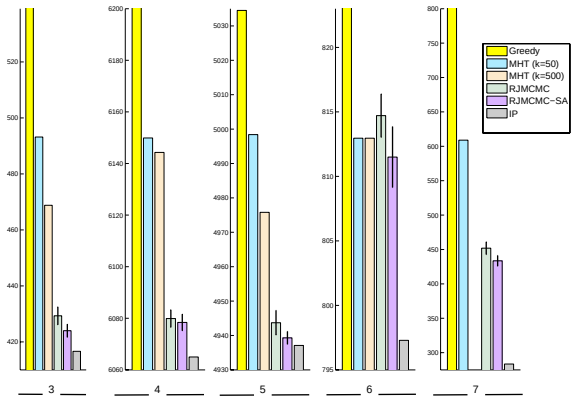

Fig. 13 $-\log(p)$ is compared for sequences (3-7) showing RJMCMC-SA achieves the best heuristic search results. The vertical line represents the standard deviation σ. The posterior found using MHT (k=50) is vertically aligned for all sequences.

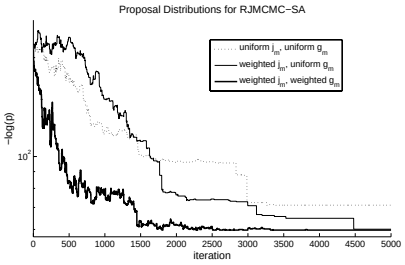

Fig. 14 Convergence under various proposal distribution choices using RJMCMC-SA for chains from the 4^{th} sequence.

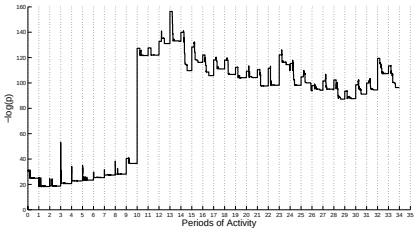

Fig. 15 MAP (-log(p)) for the 1^{st} sequence as more detections are being handled. Vertical dotted lines separate the optimisation at the end of each activity unit.

a person can be connected to an incorrect cluster, or thought to be passing by the bicycle rack. In the 7^{th} sequence for example, the scene often changed from shadow to sunlight, and the bicycle-clusters 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

	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

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.

explanations, as global explanations can resolve ambiguities that cannot be resolved by local analysis.

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 [32]. For each trajectory, protrusions representing candidate carried objects are retrieved using the method in [6].

Similar to the *Bicycles* problem, an AMG is designed and some features are selected (details available in [5]). Simple features were again chosen; people tracked in and out of the building were matched by their projected height and clothing colour. 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 [6] 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 solvers could not exhaustively search the space of explanations in reasonable time [3] 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 (Section 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).

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

Table 7 $-\log(p)$ compared across G, MHT, 40 runs of 10 parallel chains RJMCMC and RJMCMC-SA and IP using XPRESS-MP. MAP is shown for the 25 people detections and the corresponding candidate bags as well as all detections. For the larger-scale problem, result were not available for MHT search with larger k and IP due to the implementation running out of memory.

When compared to ground-truth data, 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

[3] using 20GB of memory for about 10 hours

Fig. 16 Correctly associated detections when global explanations are considered.

height and clothing colour. 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). AMGs allow specifying attribute rules, as well as constraints that confine the grammar's parses to consistent ones.

For each input video, detectors retrieve the set of detections, which represents terminal symbols along with the synthetic attribute values. An algorithm then automatically builds a Bayesian Network (BN) to model the probability distribution over the set of global explanations for these detections. The set of possible labellings of the BN corresponds to the set of all global explanations. Search techniques are proposed to find the MAP, as the combinatorial search becomes intractable when the complexity and duration of the activity increase. The approach was tested on two case studies. Results show that RJMCMC along with Simulated Annealing is the best heuristic search technique, that is scalable when the complexity increases.

7.1 Types of Activities

Any activity can indeed be represented by an AMG and recognised using the framework proposed in this paper. Nothing restricts the approach from extending to multi-agent activities, and this is left for future work. The framework is expected to outperform other representations and recognition approaches in cases of

- Highly interleaved events: In this case the usage of 'multisets' simplifies the representation. The BN models all possible interleaved events, and the MAP enables recognising the most probable explanation. The two cases introduced in this paper are examples of higly interleaved events.
- Temporal flexibility: When events can occur in any order, and only a few temporal constraints need to be enforced, AMG provides a concise representation. For example, cooking activities are typically flexible, and would benefit from this framework.
- Expected ambiguity in detections: Global explanations prove valuable in cases where constraints can disambiguiate low-level measurements.

Alternatively, when the events are highly-strucutred, least-interleaved and occur in order like activities on a factory production line, then string grammars provide equally simple representation, and parsing approaches can be applied.

7.2 Learning AMGs

In the current framework, the AMG is manually built for each activity. This includes building the hierarchical structures, deciding on the features that could distinguish the different event types, and listing the constraints. Though Zhu and Mumford emphasise that learning a compositional structure depends on the objective of the composition, and cannot be merely based on statistical data [52], recent advancements in learning hierarchies from unlabelled, or weakly-labelled data are worth highlighting.

The leading work of [20] uses mining techniques to extract spatio-temporal relationships from unlabelled data. Causal relationships are concluded from multiple occurrences, and might be hallucinated. Alternatively, weakly-labelled data can be used to build or adapt grammars. Textual annotations were used in [18] to build an initial And-Or graph. Given the annotation, actions and their temporal relationships are mapped to nodes and temporal constraints. As weakly-labelled data is parsed, the graph is iteratively modified and extended to best explain the underlying observations while maintaining the representation's simplicity. These approaches could be utilised to extract hierarchies for different domains.

After the hierarchy is built, attributes and attribute rules can be learnt from features that best distinguish events at every layer of the hierarchy. Feature selection is of primary interest to the machine learning community. Examples in activity recognition include selecting from a pool of available features [38], and boosting [44]. The combined learning of hierarchies, features and constraints from unlabelled, or weakly-labelled data requires further research.

A Derivation

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

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

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

The third factor in Equation 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).

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

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

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

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

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

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

References

1. S. P. Abney. Stochastic attribute-value grammars. *Computational Linguistics*, 23(4):597–618, 1997.
2. A. Aho, R. Sethi, and J. Ulman. *Compilers: principles, techniques and tools*. Addison-Wesley, 1986.
3. C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.
4. J. Blevins. Feature-based grammar. In R. Borsley and K. Borjars, editors, *Nontransformational Syntax: A Guide to Current Models*. Blackwell, TO APPEAR.
5. D. Damen. *Activity Analysis: Finding Explanations for Sets of Events*. PhD thesis, University of Leeds, UK, 2009.
6. D. Damen and D. Hogg. Detecting carried objects in short video sequences. In *European Computer Vision Conference (ECCV)*, volume 3, pages 154–167, 2008.
7. D. Damen and D. Hogg. Attribute multiset grammars for global explanations of activities. In *Proc. British Machine Vision Conference (BMVC)*, 2009.
8. D. Damen and D. Hogg. Recognizing linked events: Searching the space of feasible explanations. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 927–934, 2009.
9. I. Ersoy, F. Bunyak, and S. R. Subramanya. A framework for trajectory based visual event retrieval. In *Proc. Int.Conf. on Information Technology: Coding and Computing (ITCC)*, volume 2, pages 23–27, 2004.
10. Q. Fan, R. Bobbitt, Y. Zhai, A. Yanagawa, S. Pankanti, and A. Hampapur. Recognition of repetitive sequential human activity. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.
11. P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2000.
12. D. O. FICO. XPRESS-MP solver - version 19.00.17, 2007.
13. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721 – 741, 1984.
14. E. Gollin. *A Method for the Specification and Parsing of Visual Languages*. PhD thesis, Brown University, 1991.
15. S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *Proc. International Conference on Computer Vision (ICCV)*, 2003.
16. P. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.
17. P. Green. Trans-dimensional Markov chain Monte Carlo. In P. Green, N. Lid Hjort, and S. Richardson, editors, *Highly structured stochastic systems*. Oxford University Press, Oxford, 2003.
18. A. Gupta, P. Srinivasan, J. Shi, and L. Davis. Learning a visually grounded storyline model from annotated videos. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.
19. P. A. Hall. Equivalence between and/or graphs and context-free grammars. *Communications of the ACM*, 16(7):444–445, 1973.
20. R. Hamid, S. Maddi, A. Bobick, and M. Essa. Structure from statistics - unsupervised activity analysis using suffix trees. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2007.
21. F. Han and S. Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1778–1785, 2005.
22. S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, 2004.
23. T. Huang and S. Russell. Object identification: A Bayesian analysis with application to traffic surveillance. *Artificial Intelligence*, 103(1-2):77–93, 1998.
24. S. Intille and A. Bobick. Recognizing planned, multiperson action. *Computer Vision and Image Understanding*, 81(3):414–445, 2001.
25. Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.
26. S.-W. Joo and R. Chellappa. Attribute grammar-based event recognition and anomaly detection. In *Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 107–114, 2006.

27. S.-W. Joo and R. Chellappa. Recognition of multi-object events using attribute grammars. In *Proc. Int. Conf. on Image Processing (ICIP)*, pages 2897–2900, 2006.
28. U. Kastens. Ordered attributed grammars. *Acta Informatica*, 13:229–256, 1980.
29. K. M. Kitani, Y. Sato, and A. Sugimoto. Deleted interpolation using a hierarchical Bayesian grammar network for recognizing human activity. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS)*, pages 239–246, 2005.
30. D. Knuth. Semantics of context-free languages. *Mathematical Systems Theory*, 2(2), 1968.
31. L. Lin, H. Gong, L. Li, and L. Wang. Semantic event representation and recognition using syntactic attribute graph grammar. *Pattern Recognition Letters*, 30(2):180–186, 2009.
32. D. Magee. Tracking multiple vehicles using foreground, background and motion models. In *Proc. Workshop on Statistical Methods in Video Processing*, pages 7–12, 2002.
33. C. Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Trans. on Automated Control*, 22(3):302–312, 1977.
34. R. Nevatia, T. Zhao, and S. Hongeng. Hierarchical language-based representation of events in video streams. In *Proc. of IEEE Workshop on Event Mining (EVENT)*, 2003.
35. N. Nguyen, S. Venkatesh, and H. Bui. Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association. In *Proc. British Machine Vision Conference (BMVC)*, volume 3, pages 1239–1248, 2006.
36. S. Oh, S. Russell, and S. Sastry. Markov chain Monte Carlo data association for general multiple-target tracking problems. In *Decision and Control, (CDC)*, volume 1, pages 735–742, 2004.
37. D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24(6):843–854, 1979.
38. P. Riberio and J. Santos-Victor. Human activity recognition from video: modeling, feature selection and classification architecture. In *Intl. Workshop on Human Activity Recognition and Modelling*, 2005.
39. M. Rota and M. Thonnat. Video sequence interpretation for visual surveillance. In *IEEE Int. Workshop on Visual Surveillance (VS)*, Dublin, Ireland, 2000.
40. C. Rother, V. Kolmogorov, and A. Blake. Grabcut -interactive foreground extraction using iterated graph cuts. In *ACM Transa. on Graphics (SIGGRAPH)*, 2004.
41. Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation networks for recognition of partially ordered sequential action. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 862–869, 2004.
42. J. Siskind. Visual event classification via force dynamics. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 149–155, 2000.
43. K. Smith. *Bayesian Methods for Visual Multi-object Tracking with Applications to Human Activity Recognition*. PhD thesis, Ecole Polytechnique Federale de Lausanne (EPFL), 2007.
44. P. Smith, N. Vitoria Lobo, and M. Shah. Temporalboost for event recognition. In *IEEE Int. Conf on Computer Vision (ICCV)*, 2005.
45. S. Tran and L. Davis. Event modeling and recognition using Markov logic networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2008.
46. H. Williams. *Model Building in Mathematical Programming*. Wiley, 4th edition, 1999.
47. B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *Proc. Int. Conf. on Computer Vision(ICCV)*, volume 1, pages 90–97, 2005.
48. Y. Wu and T. Huang. Robust visual tracking by integrating multiple cues based on co-inference learning. *Int. Journal of Computer Vision*, 58(1):55–71, 2004.
49. R. Young, J. Kittler, and J. Matas. Hypothesis selection for scene interpretation using grammatical models of scene evolution. In *Int. Conf. on Pattern Recognition*, Australia, 1998.
50. Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2007.
51. T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2004.
52. S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362, 2006.