Attribute Multiset Grammars for Global Explanations of Activities

Dima Damen
dima@comp.leeds.ac.uk
David Hogg
dch@comp.leeds.ac.uk

School of Computing
University of Leeds
Leeds, UK

Abstract

Recognizing multiple interleaved activities in a video requires implicitly partitioning the detections for each activity. Furthermore, constraints between activities are important in finding valid explanations for all detections. We use Attribute Multiset Grammars (AMGs) as a formal representation for a domain's knowledge to encode intra- and inter-activity constraints. We show how AMGs can be used to parse all the observations into 'feasible' global explanations. We also present an algorithm for building a Bayesian network (BN) given an AMG and a set of detections. The set of labellings of the BN corresponds to the set of all possible parse trees. Finding the best explanation then amounts to finding the maximum a posteriori labeling of the BN. The technique is successfully applied to two different problems including the challenging problem of associating pedestrians and carried objects entering and departing a building.

1 Introduction

Automatic surveillance requires recognizing activities that involve one or more interacting agents. While most activity recognition techniques focus on recognizing a single activity, realistic surveillance involves multiple interleaved activities, often extending over a long duration. In these situations, the activities are often mutually constrained. For example, a person entering a building can be observed departing only once at a later time. In visual interpretation, these constraints can be exploited to disambiguate uncertain visual data through seeking a globally consistent explanation [4, 8]. However, a general way to formalise the set of globally consistent explanations for a given domain is not yet available. In the current paper, we show how this can be achieved using a grammar formalism.

During the 80s and early 90s, picture layout grammars were used to parse two-dimensional visual languages like sketches, flowcharts and state diagrams [9, 10, 16]. Grammars have also been used for activity recognition. The work of Ivanov and Bobick [12] highlighted the importance of formal methods to encode expert knowledge for recognizing activities in video. They used Stochastic Context Free Grammars (SCFG) to represent the ways in which complex activities can be constructed. The Earley-Stolcke parser generates the best parse of the detected sequence of primitive events given the grammar. They evaluated their approach on gesture recognition and surveillance within a car park. However the method expects a single activity, involving one or more interacting agents, in each given sequence. SCFG has also been used to recognize visual activities in a blackjack game [17]. Given one dealer (assumed known) and multiple players, a parser infers the sequence of the game and identifies the winner. The parser corrects possible detection errors by an exhaustive search which is tractable given the small number of detections in one game.

Although context-free grammars have been used successfully for recognizing sequential activity patterns [12, 17], they do not easily extend to interleaved activities where the patterns are overlapping and not easily segmented. A related problem exists for graphical components in two-dimensional sketches - there is no natural sequential ordering. Attribute Multiset Grammars (AMG) have been proposed for such problems [10]. Each rule in an AMG rewrites a nonterminal symbol as a multiset instead of a string. It combines this with attributes to extract meaning from parse trees and to constrain the application of rules, for example to impose geometric constraints between entities. We will return to explain AMGs in more detail in Section 2.1.

Attribute graph grammars are closely related and have been used to identify man-made rectangular objects like tables, floor tiles and windows in static images [11]. Strong hypotheses of rectangles from edge detection are used to hypothesize larger structures through the application of grammar rules. This can initiate a search for weaker evidence of rectangles consistent with these larger structures. To parse the given image, recursive top-down/bottom-up parsing is used. At each iteration, Data Driven Markov Chain Monte Carlo (DDMCMC) samples from the possible hypotheses in the grammar, and the evidence in the image is tested. The paper shows the power of attribute grammars, as the parsing conveys information up and down the parse trees. Attribute graph grammars have also been used to detect anomalous events in a car park [13], i.e. those inputs that could not be parsed according to the grammar. In a video with multiple interleaved activities, the set of objects are assumed to be partitioned into the different threads of activity.

A recent attempt to overcome an assumed partitioning of primitive events into activity threads uses attribute graph grammars to analyze activities in a car park [15]. For a pick-up event, for example, several of the detected people and cars could have performed the activity. The attribute graph grammar together with a set of detected objects determines a set of possible interpretations of a scenario. A probability distribution over this set of interpretations is expressed as a Markov Random Field (MRF), with a list of candidate objects at each vertex. The pairwise potentials in the MRF are derived from the proximities of people and cars. While this framework can partition the primitive events into activities, it does not take into consideration the constraints between activities. In some situations, this could lead to globally inconsistent (infeasible) explanations. In the car park domain, a car can drop-off several people, yet a person can be dropped off by only one car. Such constraints have been expressed by first order logic rules in [19]. Markov logic networks are then used in the inference process. This approach though does not distinguish between rules that define activities and those that constrain feasible explanations. An entirely different approach defines events first and then adds constraints, and is solved as a constraint satisfaction problem [18]. In this work, we wish to combine a constraint satisfaction approach with the expressive power of grammars.

In [6], we searched for a constrained explanation for the Bicycles problem by maintaining a multiple-hypotheses tree (MHT). The activities and the constraints were textually described. A solution to the bicycles problem correctly links people and bicycles, and 'drops' followed by 'picks' in a bicycle rack. In our previous work [8], events in a chosen domain are assumed to form a hierarchy, with primitive events at its base and compound events (composed of simpler primitive and compound events) at higher levels. The hierarchies are expressed using diagrammatic trees, along with implied constraints on the 'feasible' hierarchies. The work assumes the compositional hierarchy is not recursive, and represents the probability distribution over possible explanations using a Bayesian network (BN). We proposed an approach for searching such a BN for the Maximum a Posteriori (MAP) solution using Reversible Jump Markov Chain Monte Carlo (RJMCMC), which outperformed MHT.

In this paper, we augment the method in [8] with a formal grammar to characterise the set of possible global explanations. This is sufficiently general to be applicable in different domains. We use Attribute Multiset Grammars, where activities and constraints can be formally defined, to represent global explanations. Parsing a set of detections by such a grammar would **explain all detected events and group them into threads of activities, providing a 'feasible' explanation that satisfies the domain's intra- and inter-activity constraints**. To find the best parse tree given a set of detections, we present an algorithm that transforms the grammar into a Bayesian network (BN). The set of possible labellings of the Bayesian network represents all parse trees for the given set of detections. The generality of this formalism is demonstrated by re-formulating the Bicycles problem [8], and giving the specification for a challenging new problem in which the task is to link people and carried bags into and out of a building. Section 3 tests the approach on one day of recorded video along with results that demonstrate the framework's ability to provide global explanations.

2 The method

This section describes how attribute multiset grammars can encode sets of complex visual activities and how all the events detected in a video sequence are parsed into a global explanation. Constraints on the attributes govern the parsing process and confine the parse trees to feasible explanations. The section then presents an algorithm to map a set of detections, given the domain's AMG grammar, into a Bayesian network structure. The prior and conditional probabilities contained in this BN are associated with the different grammar rules, and obtained from expert knowledge with some parameter training. The desired explanation is then the Maximum A Posteriori (MAP) solution of the BN.

2.1 Attribute Multiset Grammars

Attribute Grammars were first introduced by Knuth by adding attributes to the terminal and nonterminal symbols of a grammar [14]. They are also referred to as Feature-Based Grammars (FBG) [2] and Attribute-Value Grammars [1]. These attributes can be used in three ways. The first is to propagate information towards the root of the parse tree; ancestors can derive their attributes from those of their descendants. The second is to propagate attributes down towards the leaves; descendants inherit characteristics of their ancestors. The third is to use attributes to govern the application of production rules, thereby constraining the language generated by the grammar.

While a conventional (context-free) grammar rewrites a symbol into a sequence of symbols, in multiset grammars production rules rewrite a symbol into a multiset [1]. Attribute Multiset Grammars (AMG) were introduced in [10] for representing the constituents and layout of a picture. We define an AMG 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 (S $\in$ N), A(x) is a set of attributes defined for the symbol

[1] A multiset (or a bag) is a generalization of a set where the order is irrelevant although each symbol can still appear more than once

$x \in N \cup T$, and P is the set of production rules. We use the notation $x.a$ to mean 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 descendants, and $A_1(x)$ is the set of *inherited* attributes which are calculated based on the attributes of the ancestors.

Each production rule $p \in P$ is a 3-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. M is a set of attribute rules, where each rule $m \in M$ assigns a value to one of the attributes of the symbols involved in r. C defines a set of attribute constraints that govern the application of the production rule; the production rule can only be applied if all the attribute constraints are satisfied. To illustrate, consider the AMG G_a = ({S,A,B}, {α, β, γ}, S, A_0(A) = A_0(B) = $A_0(\alpha)$ = $A_0(\beta)$ = $A_0(\gamma)$ = {time} and A_1(B) = A_1 (β) = {count}, P). Figure 1 shows the production rules.

rule	Syntactic Rule (r)			Attribute Rules (M)			Attribute Constraints (C)		
p_1	S	$\rightarrow$	$A^\star$, $B^\star$, $\alpha^\star, \gamma^\star$						
p_2	A	$\rightarrow$	α, B	A.time	=	α.time+B.time	α.time	<	B.time
				B.count	=	1	B.count	$\neq$	1
p_3	B	$\rightarrow$	β, γ	B.time	=	γ.time	β.time	<	γ.time
				β.count	=	1	β.count	$\neq$	1

Figure 1: Production rules for a simple Attribute Multiset Grammar (G_a)

Given an input video, detectors are used to retrieve a multiset of detections D. Each detection is an instance of one of the terminals T together with assigned values for the synthetic attributes defined for that terminal. The set of all derivations of D, given the AMG, is the set of all possible explanations for the input video. For the grammar G_a, suppose the detectors generated the following multiset D = $\{\alpha_1(time = 1), \alpha_2(time = 2), \beta_1(time = 2), \gamma_1(time = 3), \gamma_2(time = 4)\}$ - subscripts distinguish different instances of the same terminal. Figure 2 shows two possible derivations (parse trees). Following the approach proposed in [8], we build a Bayesian network (BN), with conditional links between events and their associated observations, between compound events and their constituent events, and between nodes and a deterministic random variable when enforcing consistency in the parse tree. The desired explanation is the MAP for this network. Figure 3 shows this Bayesian network for the specified detection multiset D along with two labellings that reflect the parse trees in Figure 2. A hidden random variable (RV) represents each possible nonroot nonterminal in a parse tree, and is labeled true if the nonterminal appears in the parse tree, and false otherwise. Algorithm 1 details the steps for building a BN out of a set of detections and an AMG grammar. The algorithm distinguishes between the synthetic and inherited attribute constraints. Inher-

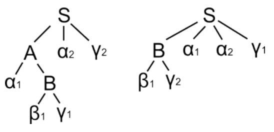

Figure 2: Two parse trees given a multiset of detections and AMG G_a

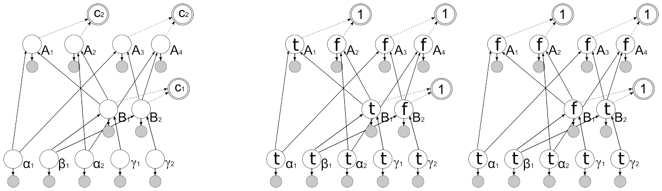

Figure 3: The Bayesian network for the detections D and AMG G_a. The two labellings reflect the parse trees in Figure 2. A node is labeled true if it appears in the parse tree

```
input  : Grammar G = (N, T, S, A, P), detections multiset D
output : Bayesian Network Structure
initialize an empty Bayesian Network (BN)
orders rules $P$ starting with those containing terminals then bottom-up
foreach terminal instance $t \in D$
    add hidden RV to BN of type $t$
    if $t$ has synthetic attributes then
        add a related observed RV to hold the attribute values
foreach rule $p \in P$ ($p.r : X_0 \rightarrow X_1, X_2, ..., X_n$)
    if $X_0 \neq S$ then
        Let I($X_i$) be the set of nodes in BN of type $X_i$
        foreach tuple $b \in I(X_1) \times I(X_2) \times ... \times I(X_n)$
            if $b$ satisfies attribute constraints $p.C$ then
                add hidden RV to the BN of type $X_0$
                foreach attribute rule $m \in p.M$
                    if $m$ updates a synthetic attribute then
                        run $m$ assigning a synthetic attribute value to $X_0$
                add a related observed RV to hold attribute values
                all nodes in the tuple $b$ parent the created hidden RV
Let Nodes$_n$ be the set of all hidden RVs associated with nonterminal symbols $N$
while $Nodes_n \neq \phi$ do
    find one set 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 deterministic RV $c$ to hold the inherited constraints
        all nodes in $Nodes_p$ parent the deterministic RV $c$
```

Algorithm 1: Mapping a multiset of detections D to the Bayesian network structure that represents the probability distribution over the set of possible parses, given an AMG grammar G

ited constraints define constraints between the different activities. For example, to recognize the event of picking a person up by a car, the person can be picked up once, while the car can still pick up other people. The number of pick-ups of each individual is thus constrained to a maximum of 1. Such inherited constraints are enforced by adding deterministic random variables linking the inter-dependent activities.

In the next two subsections, we present an AMG for two different problems. A brief description of the prior and conditional probability distributions is given in Section 3.

2.2 An AMG for the *Bicycles* problem

The *Bicycles* problem [8] aims at recognizing bicycle drop and pick events, as well as linking the drop of a bicycle to its subsequent pick. For this problem, an AMG G_b is defined as follows;

- Two detectors are required, one to retrieve people tracked within the racks area (x) and another to detect bicycle clusters (y).
- N = {S, V, Z}; T = {x, y, u} where u represents unseen events, Z links a person to a bicycle, and V links drop and pick events.

- A_0 (x) = {au, traj}, A_0 (y) = {au, pos}, A_0 (Z) = {au, pos, match}, A_0 (V) = {match}, A_1(x) = A_1(y) = A_1(Z) = {action, count}, A_1 (V) = {action}
  A(S) = A(u) = ϕ
  where *au* (integer) represents the activity unit during which the symbol is detected [8], *traj* is a set of bounding boxes representing the trajectory and extent of a tracked object, *pos* is a bounding box around a bicycle cluster, *match* (real) assesses the likelihood of linking two detections into a compound event, *count* (integer) represents the number of activities in which a detection participates, and *action* (string) describes the activity. Two functions are defined on these attributes:
  ψ_Z (traj, pos): finds the maximum overlap between a trajectory and a bicycle cluster.
  ψ_V (pos, pos): finds the pixel-to-pixel match between two bicycle clusters.
- We chose an attribute called 'action' and defined it for all grammar symbols. For each symbol, the values assigned to 'action' by the production rules form the set of non-false labels for nodes of that type. For example, nodes of type *Z* in the BN can take any of the labels {drop, pick, false}, while nodes of type *V* have two possible labels {drop-pick, false}.

The production rules in Figure 4 encode all the domain's knowledge and constraints. Algorithm 1 transforms this AMG for a set of detections into the same BN as that obtained in [8].

Syntactic Rule (r)				Attribute Rules (M)			Attribute Constraints (C)		
p_1	S	→	$V^\star, x^\star, y^\star$	y.action	=	"noise"	y.count	<	1
				x.action	=	"pass-by"	x.count	≠	1
p_2	V	→	Z_1, Z_2	V.action	=	"drop-pick"	Z_1.au	<	Z_2.au
				Z_1.action	=	"drop"	Z_1.count	≠	1
				Z_2.action	=	"pick"	Z_2.count	≠	1
				V.match	=	ψ_V (Z_1.pos, Z_2.pos)			
				Z_1.count	=	Z_2.count = 1			
p_3	V	→	Z, u	V.action	=	"drop-only"	Z.count	≠	1
				Z.action	=	"drop"			
				Z.count	=	1			
p_4	V	→	u, Z	V.action	=	"pick-only"	Z.count	≠	1
				Z.action	=	"pick"			
				Z.count	=	1			
p_5	Z	→	x, y	x.action	=	Z.action	x.au	=	y.au
				y.action	=	Z.action	x.count	≠	1
				Z.au	=	x.au			
				Z.pos	=	y.pos			
				Z.match	=	ψ_Z (x.traj, y.pos)			
				x.count	=	1			
				y.count	=	y.count+1			

Figure 4: Production rules for the bicycle attribute multiset grammar G_b. Subscripts are used to distinguish between occurrences of like nonterminals in the same production rule.

2.3 An AMG for the *Enter-Exit* Problem

We now consider a new problem and define the activities using an AMG. A different AMG is specified for each new problem, yet the same technique is used to convert the AMG to a BN, and for searching the BN for the MAP solution. The task is to associate individuals entering and exiting a building along with bags they may be carrying. For example, we wish to recognize an individual entering a building with a bag and departing without it. Two detectors are needed to detect people (t) and carried objects (b). An AMG G_c is defined as follows

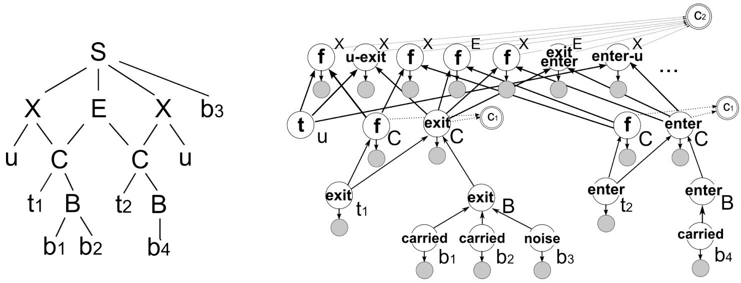

Figure 5: A sample parse tree and labelled BN for the *Enter-Exit* problem

- Two detectors are required, one to detect people (t) and one for carried objects (b).
- N = {S, X, E, C, B}; T = {t, b, u} where S is the start symbol, X finds exit-enter links, E finds enter-exit links, C links people to carried objects, B represents one or more carried objects for each person, and u represents unobserved events.
- A_0 (t) = {trajID, time}, A_0 (b) = {trajID}, A_0 (B) = {NoBags, trajID}, A_0 (C) = {time, NoBags}, A_0 (E) = A_0 (X) = {bagDiff, match}
  A_1(t) = A_1(b) = A_1 (B) = {action, count}, A_1 (C) = {action, eCount, xCount},
  A_1 (E) = A_1 (X) = {action}, A(S) = A(u) = ϕ
  where *trajID* (integer) is an id for each trajectory, *time* (integer) indicates when the event happened, *NoBags* (integer) represents the number of carried bags, *bagDiff* (integer) represents the number of non-matching bags, *match* (real) assesses the likelihood of matching two trajectories, {*count, eCount, xCount*} (integer) represent the number of different activities in which the detection participates, and *action* (string) describes the activity.
  One function is defined ψ_M (traj, traj) for the match between two trajectories.

Figure 6 shows the production rules of the AMG for this problem. Figure 5 shows a parse tree for a multiset of detections along with a labeled BN, built using Algorithm 1.

3 Experiments and Results

Experimental results for the *Bicycles* problem were presented in our earlier work [8] as the BN used there is the same as the one generated from the AMG. We focus here on results for the *enter-exit* problem, demonstrating the power of the AMG formulation and algorithm. We recorded a full day (12 hours) outside a building entrance. 326 instances of someone passing through the entrance area were detected after manually rejecting groups of people walking together. The baggage detector from [7] was run on the dataset resulting in 429 candidate bags. Figure 7 shows the viewpoint and a few detected bags. To distinguish false detections from actual carried objects, the frequency (ratio of frames during which the bag was detected) as well as the colour similarity between the bag and the adjacent clothing are used. From hand-classified bags for the first two-hours of video, we estimated a Gaussian likelihood function for the frequency given the class of the detection (noise or carried-object). Similarly, we estimated a Gaussian likelihood function for the colour similarity. For the given camera view, we estimate the conditional probability distribution for the trajectory's mean direction given someone is entering, exiting or passing by. The conditional probability density is estimated using supervision and represented using von Mises distribution.

	Syntactic Rule (r)			Attribute Rules (M)			Attribute Constraints (C)		
p_1	S	→	$X^\star$, $E^\star$, $t^\star$, $b^\star$	b.action	=	“noise”	b.count	$\neq$	1
				t.action	=	“pass-by”	t.count	$\neq$	1
p_2	X	→	C_1, C_2	C_1.action	=	“exit”	C_1.action	$\neq$	“enter”
				C_2.action	=	“enter”	C_2.action	$\neq$	“exit”
				X.action	=	“exit-enter”	C_1.time	$<$	C_2.time
				X.match	=	ψ_M (C_1, C_2)	C_1.xCount	$\neq$	1
				X.bagDiff	=	$\lvert C_1$.NoBags - C_2.NoBags$\rvert$	C_2.xCount	$\neq$	1
				C_1.xCount	=	C_2.xCount = 1			
p_3	X	→	C, u	C.action	=	“exit”	C.action	$\neq$	“enter”
				X.action	=	“exit-u”	C.xCount	$\neq$	1
				C.xCount	=	1			
p_4	X	→	u, C	C.action	=	“enter”	C.action	$\neq$	“exit”
				X.action	=	“u-enter”	C.xCount	$\neq$	1
				C.xCount	=	1			
p_5	E	→	C_1, C_2	C_1.action	=	“enter”	C_1.action	$\neq$	“exit”
				C_2.action	=	“exit”	C_2.action	$\neq$	“enter”
				E.action	=	“enter-exit”	C_1.time	$<$	C_2.time
				E.match	=	ψ_M (C_1, C_2)	C_1.eCount	$\neq$	1
				E.bagDiff	=	$\lvert C_1$.NoBags - C_2.NoBags$\rvert$	C_2.eCount	$\neq$	1
				C_1.eCount	=	C_2.eCount = 1			
p_6	E	→	C, u	C.action	=	“enter”	C.action	$\neq$	“exit”
				E.action	=	“enter-u”	C.eCount	$\neq$	1
				C.eCount	=	1			
p_7	E	→	u, C	C.action	=	“exit”	C.action	$\neq$	“enter”
				E.action	=	“u-exit”	C.eCount	$\neq$	1
				C.eCount	=	1			
p_8	C	→	t, B	t.action	=	C.action	t.trajID	=	B.trajID
				B.action	=	C.action	t.count	$\neq$	1
				C.NoBags	=	B.NoBags	B.count	$\neq$	1
				C.time	=	t.time			
				t.count	=	B.count = 1			
p_9	C	→	t	t.action	=	C.action	t.count	$\neq$	1
				C.NoBags	=	0			
				C.time	=	t.time			
				t.count	=	1			
p_{10}	B	→	$b^\star$	b.action	=	“carried”	b_i.trajID	=	b_j.trajID
				b.count	=	1	b.count	$\neq$	1
				B.NoBags	=	$\lvert b*\rvert$			
				B.trajID	=	b.trajID			

Figure 6: Production rules for the enter-exit attribute multiset grammar G_c

Matching two trajectories (referred to as ψ_M in the grammar G_c) used the height and clothing colour along with matching any carried objects. The projected height at each frame was estimated, up to a constant factor, using a cross-ratio after manually retrieving the vanishing point and the horizon vanishing line [5]. Given the projected heights for two trajectories, the Welch t-test estimated the goodness of match between the two distributions. Supervised training is used to map the scores into likelihoods for correct and incorrect pairs. To match the clothing, a per-bin median histogram was calculated [3]. Histogram intersection scored the clothing colour match, and supervision converted those matches into Gaussian likelihoods. Similarly, carried objects were matched based on colour and the relative height of the detected bag to that of the individual. Priors and conditional probabilities for the BN were selected based on a separate one hour recording.

For the 326 person detections and associated detected bags, a BN is constructed from G_c using Algorithm 1. The number of hidden RVs in the generated BN is 190849 ($\lvert$I(B)$\rvert$

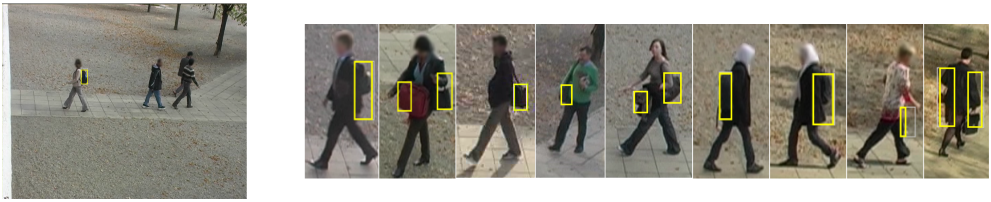

Figure 7: The viewpoint and several baggage detections

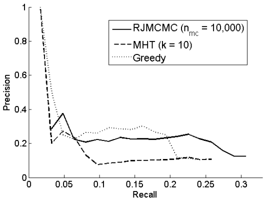

Figure 8: Precision-Recall curve for three search methods.

	Local	Global		
		Greedy	MHT	RJMCMC
Paired	13	14	16	19
Unpaired	49	48	46	43
Incorrect Pairs	173	133	135	142

Figure 9: The number of correctly paired activities, given expertise knowledge priors, comparing the unconstrained local explanation with global feasible explanations.

= 116, |I(C)| = 435, |I(X)| = |I(E)| = 95149). A MAP estimate was obtained using greedy search, Multiple-Hypotheses Tree (MHT) and RJMCMC [8]. In greedy search, given the set of detections, the activity with the highest posterior is selected iteratively until the posterior of the global explanation can no longer be improved. In MHT search, the best k global explanations are kept as the detections are explained in sequence. Using RJMCMC, the space of explanations is sampled to find the MAP. A ground truth was manually obtained in which 62 pairings were found, with each pair connecting a person entering the building to the same person leaving later, or a person leaving the building and subsequently returning to it. Figure 8 presents a Precision-Recall curve that compares the three search techniques. The curve is drawn by changing the conditional prior for connecting pairs or leaving them disconnected. Figure 9 shows the number of correctly paired activities. Notice that the best search technique (RJMCMC) only found 19 of the 62 ground truthed pairs. This is because height and colour are only weakly discriminant, as they vary under segmentation errors and illumination changes. Figure 10 shows three sequences that were correctly retrieved only when a constrained global explanation is found using RJMCMC. The second example failed to be correctly paired originally because the carried object as the person returns to the building was classified as a false detection. As the search progressed, a higher posterior was found by changing the labeling of the bag and linking the ‘exit’ to the subsequent ‘enter’. The figure also shows a correctly linked ‘exit-enter-exit-enter’ sequence.

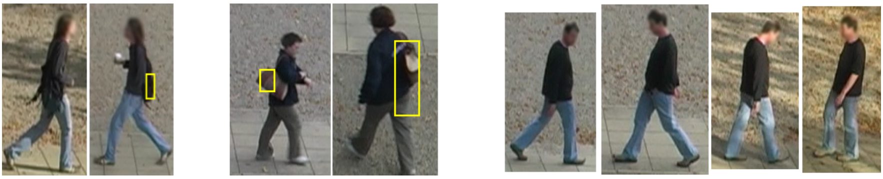

Figure 10: Correctly paired sequences when global constrained explanations are considered.

4 Conclusion

This paper highlights the power of Attribute Multiset Grammars to formalise a domain's activities in practical computer vision applications. AMG defines intra-activity temporal and spatial constraints, and inter-activity constraints that relate different activities. A parse tree for a set of detections provides a global feasible explanation. The paper presents an AMG for the previously presented *Bicycles* problem as well as for a task of associating pedestrians and carried objects entering and exiting a building. The experimental results on the second problem demonstrate the effectiveness of using the AMG formalism in finding better explanations. The extent to which this formalism is truly general and can be applied to other problems is an interesting question for future work.

References

[1] Steven P. Abney. Stochastic attribute-value grammars. *Computational Linguistics*, 23 (4):597–618, 1997.

[2] James Blevins. Feature-based grammar. In R.D. Borsley and K. Borjars, editors, *Non-transformational Syntax: A Guide to Current Models*. Blackwell, TO APPEAR.

[3] R. Bowden and P. KaewTraKulPong. Towards automated wide area visual surveillance: tracking objects between spatially-separated, uncalibrated views. *Vision, Image and Signal Processing*, 152(2):213–223, 2005.

[4] Michael Chan, Anthony Hoogs, Rahul Bhotika, Amitha Perera, John Schmiederer, and Gianfranco Doretto. Joint recognition of complex events and track matching. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1615–1622, 2006.

[5] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 1, pages 434–441 vol.1, 1999.

[6] Dima Damen and David Hogg. Associating people dropping off and picking up objects. In *Proc. British Machine Vision Conference (BMVC)*, volume 1, pages 72–81, 2007.

[7] Dima Damen and David Hogg. Detecting carried objects in short video sequences. In *European Computer Vision Conference (ECCV)*, volume 3, pages 154–167, 2008.

[8] Dima Damen and David Hogg. Recognizing linked events: Searching the space of feasible explanations. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 927–934, 2009.

[9] Larry Davis and Thomas Henderson. Hierarchical constraint processes for shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, PAMI-3(3): 265–277, 1981.

[10] Eric Gollin. *A Method for the Specification and Parsing of Visual Languages*. PhD thesis, Brown University, 1991.

[11] Feng Han and Song-Chun Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1778–1785, 2005.

[12] Yuri Ivanov and Aaron Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.

[13] Seong-Wook Joo and Rama Chellappa. Attribute grammar-based event recognition and anomaly detection. In *Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 107–114, 2006.

[14] Donald Knuth. Semantics of context-free languages. *Mathematical Systems Theory*, 2 (2), 1968.

[15] Liang Lin, Haifeng Gong, Li Li, and Liang Wang. Semantic event representation and recognition using syntactic attribute graph grammar. *Pattern Recognition Letters*, 30 (2):180–186, 2009.

[16] Kim Marriott. Constraint multiset grammars. In *IEEE Symposium on Visual Languages*, pages 118–125, 1994.

[17] Darnell Moore and Irfan Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *National conference on Artificial intelligence*, pages 770 – 776. AAAI, 2002.

[18] Ram Nevatia, Tao Zhao, and Somboon Hongeng. Hierarchical language-based representation of events in video streams. In *Proc. of IEEE Workshop on Event Mining (EVENT)*, 2003.

[19] Son Tran and Larry Davis. Event modeling and recognition using markov logic networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2008.