Activity Analysis: Finding Explanations for Sets of Events

by

Dima Jamal Al Damen

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy.

UNIVERSITY OF LEEDS

The University of Leeds
School of Computing

September 2009

The candidate confirms that the work submitted is her own and that the appropriate credit has been given where reference has been made to the work of others.

Vision is the art of seeing the invisible...

Thoughts on Various subjects
Jonathan swift (1667-1745)

Abstract

Automatic activity recognition is the computational process of analysing visual input and reasoning about detections to understand the performed events. In all but the simplest scenarios, an activity involves multiple interleaved events, some related and others independent. The activity in a car park or at a playground would typically include many events. This research assumes the possible events and any constraints between the events can be defined for the given scene. Analysing the activity should thus recognise a complete and consistent set of events; this is referred to as a global explanation of the activity. By seeking a global explanation that satisfies the activity's constraints, infeasible interpretations can be avoided, and ambiguous observations may be resolved.

An activity's events and any natural constraints are defined using a grammar formalism. Attribute Multiset Grammars (AMG) are chosen because they allow defining hierarchies, as well as attribute rules and constraints. When used for recognition, detectors are employed to gather a set of detections. Parsing the set of detections by the AMG provides a global explanation. To find the best parse tree given a set of detections, a Bayesian network models the probability distribution over the space of possible parse trees. Heuristic and exhaustive search techniques are proposed to find the maximum a posteriori global explanation.

The framework is tested for two activities: the activity in a bicycle rack, and around a building entrance. The first case study involves people locking bicycles onto a bicycle rack and picking them up later. The best global explanation for all detections gathered during the day resolves local ambiguities from occlusion or clutter. Intensive testing on 5 full days proved global analysis achieves higher recognition rates. The second case study tracks people and any objects they are carrying as they enter and exit a building entrance. A complete sequence of the person entering and exiting multiple times is recovered by the global explanation.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

Damen, D. and Hogg, D., “Attribute Multiset Grammars for Global Explanations of Activities”, *British Machine Vision Conference (BMVC)*, London - UK, BMVA, 2009.

Damen, D. and Hogg, D., “Recognizing Linked Events: Searching the Space of Feasible Explanations”, *Computer Vision and Pattern Recognition (CVPR)*, Miami - Florida, 2009.

Damen, D. and Hogg, D., “Detecting Carried Objects in Short Video Sequences”, *European Conference on Computer Vision (ECCV)*, Marseille - France, 2008.

Damen, D. and Hogg, D., “Associating People Dropping off and Picking up Objects”, *British Machine Vision Conference (BMVC)*, Warwick - UK, BMVA, 2007.

Damen, D. and Hogg, D., “Bicycle Theft Detection”, *International Crime Science Conference. (CS2)*, London-UK, 2007.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

2D Two Dimensional

3D Three Dimensional

AMG Attribute Multiset Grammar

BN Bayesian Network

ICP Iterative Closest Point algorithm

IP Integer Programming

FN False Negative

FP False Positive

MAP Maximum a Posteriori

MCMC Markov Chain Monte Carlo

MCMCDA Markov Chain Monte Carlo Data Association

MH Metropolis-Hastings algorithm

MHT Multiple Hypothesis Tree

MRF Markov Random Field

pdf Probability Density Function

PR Precision-Recall curve

RJMCMC Reversible Jump Markov Chain Monte Carlo

ROC Receiver Operating Characteristic curve

RV Random Variable

SA Simulated Annealing

SCFG Stochastic Context Free Grammar

TP True Positive

Chapter 1

Introduction

The word “activity” is defined in Merriam-Webster and Oxford English dictionaries as the “state of being active” [1, 119]. It, by definition, involves the motion or translation of objects in the environment. Visual sensors are essentially more suitable for distinguishing motion than other sensors. Analysing the activity, using visual information, is thus finding an explanation for the detections that conform to the understanding of possible scenarios.

Automatic activity recognition, which is the main subject of this thesis, is part of the discipline of artificial intelligence, and is the process of analysing visual input and reasoning about detections, using a computerised algorithm, to understand the performed events. This thesis proposes overcoming the unreliability of visual detection methods by seeking global explanations for activity recognition. 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 proposed framework bridges the gap between noisy visual observations and higher-level activity recognition. The introduction explains the need for global explanations, and the range of domains where recognition is assisted by seeking a global explanation. The rest of this chapter introduces the novelties of this research along with an overview of the chapters of the thesis.

1.1 Global explanations for activity recognition

Activity recognition has been studied intensively in computer vision. Simple actions like walking, running, waving or boxing have been recognised within clear or cluttered scenes [82, 92]. Sequences of events performed by the same individual, or events involving interactions between multiple people have also been studied. Current research has achieved significant progress towards recognising complex events in difficult scenes.

One of the major limitations in most state-of-the-art activity recognition techniques is their focus on recognising a single event given a set of detections. Some of the approaches assume that only one event can occur at each point in time. Alternatively, other approaches can recognise multiple events by assuming the detections belonging to each event can be separated from the remaining detections. Figure 1.1 shows a typical set of surveillance scenes, where the ability to separate the detections into disjoint sets cannot be realistically assumed.

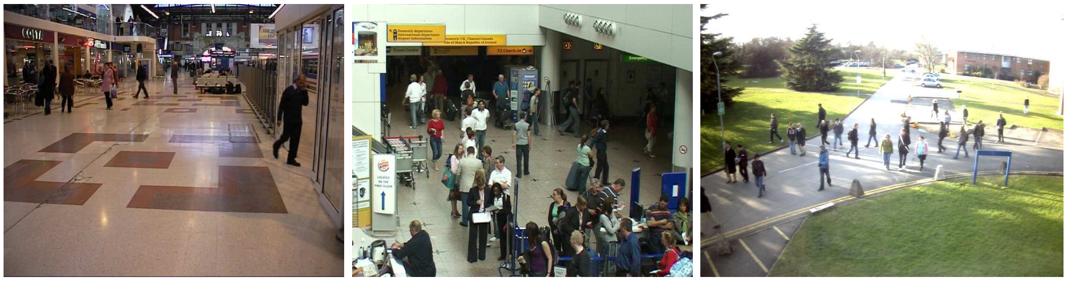


Figure 1.1: Three examples of surveillance scenes from the PETS datasets (2006, 2007 and 2009).

The terms ‘activity’ and ‘event’ have been used in various, often ambiguous, ways by the computer vision community. To avoid confusion, the terms are defined here and then used consistently throughout the remainder of the thesis. 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 related 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. Yet, in the general case the activity involves multiple related events.

To automatically analyse the activity, some evidence is gathered from observing the scene to assist recognising the occurring events. A *detector* is an independent evidence collector that targets certain evidence types, like motion detectors, car detectors or pedestrian detectors. The same detector can be used to recognise various events of different activities. It is unaware of the context in which it operates. A *detection* is a discovered entity that is acquired by a detector. For example, the trajectory of a moving object is a detection obtained using a motion detector. Given an activity, reliable detectors are chosen to retrieve a set of detections that would assist recognising the activity's events. A *feature* is a measurable characteristic of the detection. For example, the velocity is a feature of the trajectory. The detector would measure the value of this feature for each detection.

The set of detections obtained during an observed period of activity typically belongs to several events within the activity. The recognition thus requires partitioning the detections along with recognising the events. A *global explanation* for the set of detections is a *consistent set of events that covers all the detections*. The number of events is not known in advance, and varies between the different explanations for the same set of detections.

To understand the value of global explanations, let us consider the activity at a train platform. As trains approach and depart, some trajectories of people end close to the train, other trajectories appear, and some continue to move at the platform. A global explanation would recognise all the boarding and alighting events along with recognising those waiting for the next train. Assume a person is observed waiting at a train platform. As the train approaches, the person could not be detected. This implies the person boarded the train, or is still waiting another train and is currently occluded. After the train departs, the person is again detected at the platform. A global explanation would correctly attribute the person's absence to occlusion despite the initial ambiguity.

In addition to resolving uncertainties, recognising events independently can result in an inconsistent set of events. For example, a person cannot board the train while it is moving. A train can be boarded by many people at once, yet a person cannot board multiple trains. A person can though alight from one train then board another. Human cognition naturally allows explanations that satisfy such constraints. A global explanation satisfies the natural constraints by finding a consistent set of events. Figure 1.2 shows three diagrammatic sets of events for a period of activity at a train platform. These events involve people boarding and alighting trains. The three diagrams show one inconsistent set of events and two consistent sets that represent global explanations.

Though the term *global explanation* implies the complete set of events, the term *explanation* on its own is used in the thesis at times to refer to the global explanation. The sought *explanation* is the best complete and consistent set of events, covering all the detections during a period of activity. The *best explanation* is found using a Bayesian approach.

In this thesis, I assume the expected activity, given a scene, can be defined, and the recognition focuses merely on the activity's events. For example, the activity at a train platform can be defined as sets of trains approaching and departing, along with people

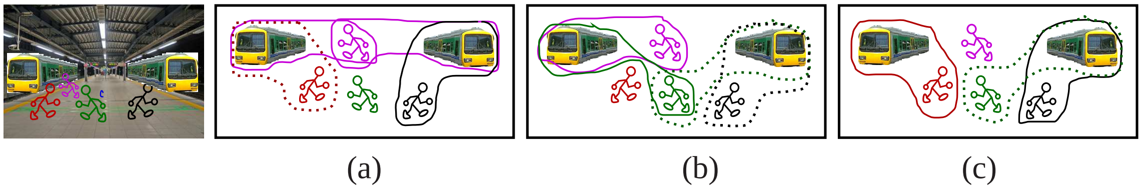


Figure 1.2: For the activity at a train platform, two trains and four people were detected, three sets of events are shown. A border is used to associate each person with a train. Dotted borders indicate alighting while solid borders indicate boarding a train. The first diagram (a) is an inconsistent set of events as a person is thought to have boarded two trains. The second (b) and third (c) diagrams are consistent sets of events.

boarding, alighting and waiting. Detections are explained, in terms of this defined activity.

1.2 Motivation, goals and novelty

Seeking global explanations and the framework proposed in this thesis were motivated by the *Bicycles* problem discussed in Chapter 5. When observing a rack area, multiple people are seen simultaneously dropping and picking bicycles. The ambiguity in each event increases with occlusion, and the uncertainty in recognising the event performed by each person can be resolved by finding a global explanation. While tackling this problem, I noticed the significant improvement in recognising uncertain input when seeking global explanations. The framework used for solving the *Bicycles* problem was generalised and applied to a different problem for tracking people and their carried objects in and out of a building.

The goal is to propose a framework that starts by formally defining the activity's events and the natural constraints. This framework should enable finding 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 1.3 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 activity and the natural constraints are employed to create an Attribute Multiset Grammar (AMG). This process is manual, and the notations and formulations of the AMGs are explained in Chapter 3. AMG is used, along with a labeled set of training sequences, to define probabilities 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 visual features. A parse of the AMG generates a global explanation for all the detections. The framework proposes an

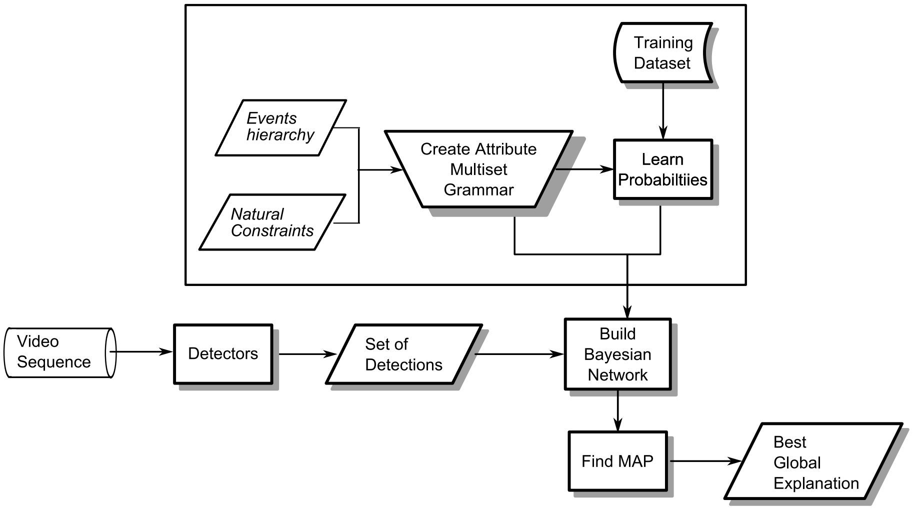


Figure 1.3: A flowchart indicating the proposed framework.

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.

The primary contributions of this research are:

- A framework for defining global explanations to recognise the complete and consistent set of events that occurred during an observed period of activity. The best explanation is found in a Bayesian approach, given a set of detections, based on the defined activity and its constraints.
- Case studies of two activities in which the framework can be used to provide global explanations.
- An experimental demonstration which shows that global solutions resolve visual ambiguities that cannot be locally resolved.
- A comparison of different techniques for searching the space of explanations.

Secondary contributions are:

- A novel detector for carried objects in short video sequences.
- A system for analysing activities in a bicycle rack. The system is tested using data recorded over 5 days at two different sites.

- A system for associating people and carried objects entering and exiting a building entrance. The system was tested on 12 hours of data.

1.3 Thesis overview

The rest of the thesis is organised as follows. Chapter 2 reviews the previous attempts in the literature to recognise complex activities using rule-based, logic-based and graphical models. The relevance of these techniques for finding global explanations is discussed.

Chapter 3 presents a grammar formalism that encodes the domain's knowledge and constraints, in order to express the global explanations. Attribute Multiset Grammars (AMG) are used to explain activities as hierarchies of events, where the leaves are primitive events that are directly detected from input video. Attributes of the grammar correspond to features of these events, and can be propagated up and down the hierarchy. The probability distribution over all global explanations, given a set of detections, is modeled by a Bayesian Network (BN). For simplicity, the chapter only presents an abstract AMG that does not correspond to a real-life problem.

The exhaustive search for the Maximum A Posteriori (MAP) labelling of the Bayesian network is intractable in all but the simplest problems. Chapter 4 presents a number of heuristic search techniques that have previously been used in the literature for searching such a BN. The chapter explains how these approaches can be applied to searching the BN representing global explanations for activity recognition.

The framework is applied to two problems. The first, and extensively analysed case study is the *Bicycles* problem briefly explained previously. An AMG is detailed in Chapter 5 for the activity, given tracked people and the appearance/disappearance of bicycles within the rack area. The chapter explains the different selected features, and how they are retrieved from the input video. The approach is tested on a real dataset of 67 hours recorded at two sites. This case study compares the search techniques, and experimentally evaluates the ability of heuristic searches to find the best global explanation given 7 video sequences of varying length and complexity.

The second case study, the *Enter-Exit* problem, is studied in Chapter 6. Similar to the first case, an AMG, Bayesian network, experiments and initial results are presented. The second case study differs from the first in its ability to recognise sequences representing the individual entering and exiting a building multiple times during the course of the day. It tracks both people and their carried objects using a single camera mounted next to a building entrance. Tested on a single day of video, preliminary results demonstrate the validity of the framework for a different activity.

The *Enter-Exit* problem requires detecting carried objects from video sequences. One of the contributions of this thesis is a novel detector for carried objects that is based on detecting protrusions from the silhouette of the person. Chapter 7 gives details of this detector along with examples and extensive testing.

Finally, Chapter 8 offers insights into future directions and the framework's limitations. It summarises the findings and contributions, and concludes the thesis.

Chapter 2

Background Review

The approach presented in this thesis attempts to find global explanations given a set of detections. Section 2.1 highlights the recent successful trend toward global analysis to resolve local ambiguities in various computer vision problems. When all the detections are evaluated simultaneously, or constraints within the explanation are considered, a ‘better’ explanation can be found.

Though little previous work deals with global explanations of activities, Section 2.2 reviews previous frameworks for complex activity recognition. An activity recognition framework enables defining activities, then recognising the activity from the input video, based on the definition. The ability of each reviewed framework to recognise the complete and consistent set of events is discussed.

When recognising interleaved events, partitioning the detections is required. This is very similar to the data association task used for tracking. Data association techniques were first introduced to establish trajectories from radar measurements, and used later for visual tracking. In the radar surveillance problem, the space of possible associations is huge. Searching this space is a combinatorial optimisation problem. Many search techniques like multiple-hypotheses trees, integer programming, and reversible jump Markov chain Monte Carlo, were compared in this domain. Section 2.3 reviews the radar surveillance problem and the seminal papers in this area. It also shows how these techniques were used for visual tracking; for connecting tracklets within the field of view of a single camera, or between non-overlapping cameras.

2.1 Global analysis in computer vision

The simultaneous analysis of all the detections has proven advantageous in many areas of computer vision, like image denoising, segmentation, shape analysis and object recognition. As detections are noisy, and often incomplete, global analysis has been introduced in these domains, and shown to outperform local interpretations. This section highlights some of the previous work that adopts global analysis, which often involves defining hard or soft constraints between local detections. Though this cannot be an exhaustive review of global analysis in vision, it motivates the significance of global explanations for activity analysis.

Several image interpretation problems can be expressed as pixel-wise labelling of the image. Labeling a pixel in isolation from its surrounding is often noisy, while global analysis combines all the information to provide a reliable explanation. Global explanations maximise the joint probability distribution of all pixel labels in the image. Using the Markovian assumption, each pixel is dependent on its neighbouring pixels, and the joint probability distribution is factorised as a Markov Random Field (MRF). This remains to date one of the most influential models in image analysis, particularly since the discovery of efficient optimisation methods, such as the Gibbs sampler [50]. Used initially for image denoising and restoration, the technique was employed later for binary image segmentation [121] and multi-class image labeling [6, 38, 144]. As Figure 2.1 shows, the local interpretation for each pixel (referred to as unary likelihood) is assisted by pairwise terms to result in a reliable segmentation. Despite the combinatorial complexity in inference using MRFs, and the delicate choice of the energy minimisation function [90], efficient exact and approximate solutions were proposed and extensively used for the optimisation [20, 21, 89].

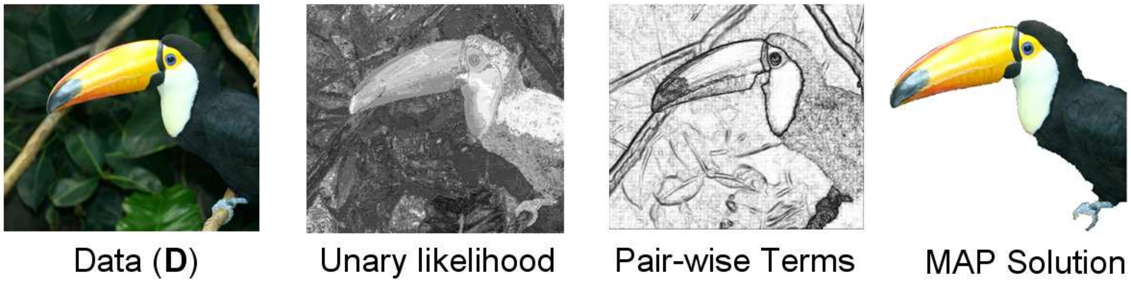


Figure 2.1: Energy minimisation for object segmentation using a MRF. Figure from [138]

Shape from Shading is an under-determined problem when each pixel is considered independently. Given the intensity at each pixel, one wishes to determine the surface gradient. Solutions incorporated global constraints like smoothness and integrability, and introduced global energy functions for minimisation [153]. The minimum solution, depending on the chosen constraints, can produce a consistent shape given a single image.

Global analysis was also introduced, around the same time as MRFs, for shape analysis. Defining algebraic and geometric constraints between volumetric object parts was used by Brooks in the novel framework ACRONYM for recognising objects in images from 3D models [23]. Each class of objects is defined as a coarse-to-fine hierarchy, where the root is a general class model with the minimum constraints. Specialised classes are recursively defined, adding geometric and algebraic constraints between the model parts, until the leaves represent specific object instances. Electric motors and planes are modeled as examples, and line segments in the images are interpreted using the models. After the model parts are detected, a combinatorial search is carried out to collect object hypotheses which represent the location, the scale and the viewpoint. The assignment of detections to model parts are globally satisfiable according to the constraints defined in the model. Such hierarchical models were also defined as a grammar by Davis and Henderson [35].

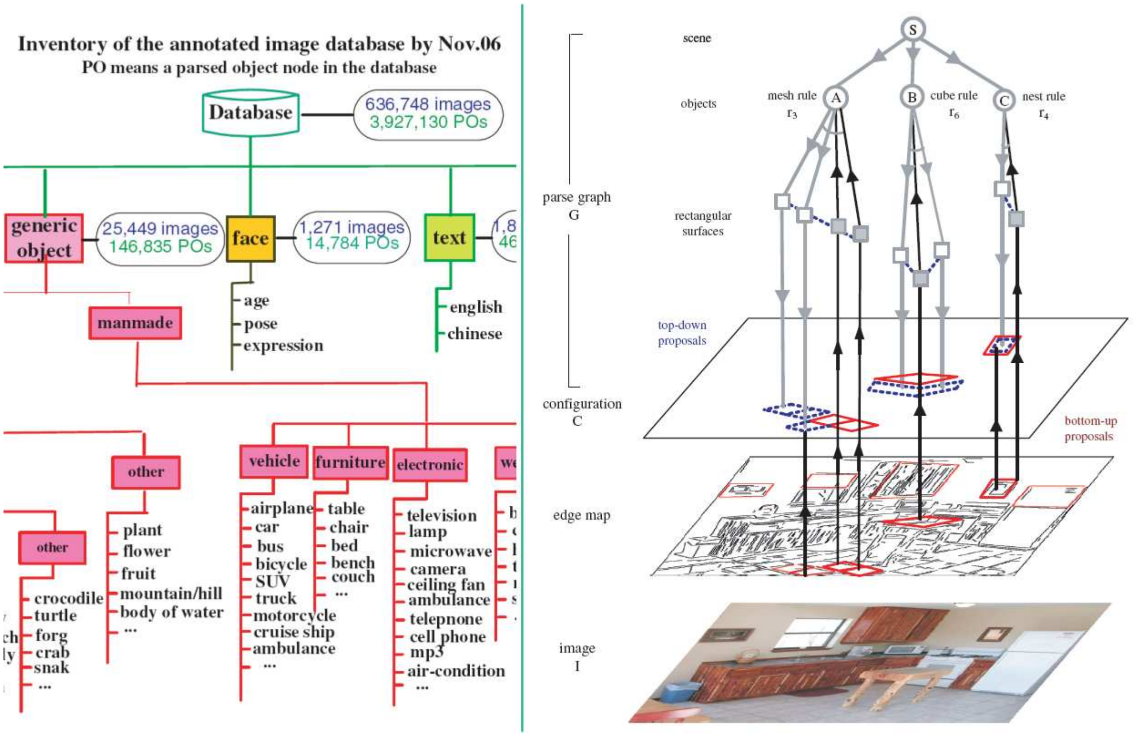


Figure 2.2: Part of the human-annotated AND-OR graph for interpreting images (left), and a corresponding recognition for rectangles (right). Figures from [157]

Closely related, attribute graph grammars have been recently used to identify man-made rectangular objects like tables, floor tiles and windows in static images [62]. Strong rectangle 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 larger structures. This top-down/bottom-up approach was further justified and explained by Zhu and Mumford in their survey [157]. The paper argues that the ultimate goal of image interpretation is to generate a comprehensive stochastic grammar that can interpret all images, as represented by the And-Or graph in Figure 2.2. An And-Or graph is an equivalent representation to context-free grammars [60]. Given a grammar, learning the parameters and the spatial relationships between image parts can be achieved from training images. The grammar is though provided by an expert, as structural decomposition is steered by the objective of this decomposition. To parse a given image, recursive top-down/bottom-up parsing is used, and Markov chain Monte Carlo (MCMC) samples the possible top-down hypotheses. The approach was applied to recognising human clothing and object categories like bicycles. Figure 2.3 shows examples of the applications from [157]. A related work for recognising facades using grammars was introduced in [118]. The derivation tree that best suits the given image is found using reversible jump Markov chain Monte Carlo.

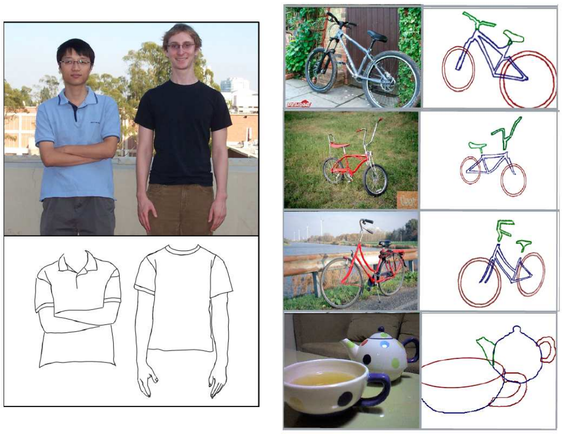


Figure 2.3: Examples of global analysis using stochastic grammars in images. Figures from [157]

Recognising an object using the joint recognition of several interrelated parts can also be considered a global approach, albeit for a single object. Such models are often referred to as ‘pictorial structures’. A pictorial structure is a deformable configuration of parts, that can be perceived as a graph with links between dependent parts. It combines a hierarchy of parts with spatial relationships between neighbouring parts. Since an efficient inference approach approximated the graph by a tree [43], pictorial structures were used frequently for object detection. A global energy function matches each part to image features along with maintaining the spatial relationships between parts. Figure 2.4 shows an example of how global analysis using pictorial structures can assist finding ambiguous body parts. In the figure, edge detection is used to retrieve the evidence from the image. The pictorial structure represents the person as a tree of ten parts: the head, the torso, and four limbs divided into upper and lower parts. Searching for each part in the image is local analysis that can miss some parts or hallucinate others. A global explanation is though capable of resolving such uncertainties and providing a consistent explanation.

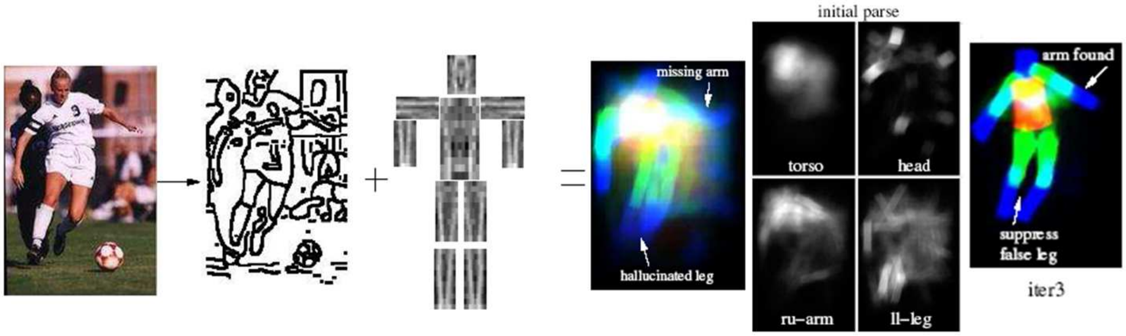


Figure 2.4: Body-pose estimation using pictorial structures. Figures from [115]

Wu and Nevatia detect multiple, possibly occluded, people given all edgelet features in a single image [145]. The paper shows that the joint likelihood of all the edgelets produces better detections, as the occlusion inter-dependency does not penalise the hidden body parts. The paper uses an iterative search algorithm to find the best explanation for all the detections simultaneously.

Global analysis has also been recently employed to jointly recognise an object and its surrounding context. By learning the spatial relationships between the object and its context, Heitz and Koller improved the detection of objects in aerial images [68]. While object detection can often lead to unrealistic explanations, considering the surrounding supports weaker evidence or rejects inconsistent explanations. In Figure 2.5, false car detections were rejected by studying the context, as cars cannot exist on top of roof buildings. Similarly, the output of a bicycle detector can be improved by recognising the surrounding context.

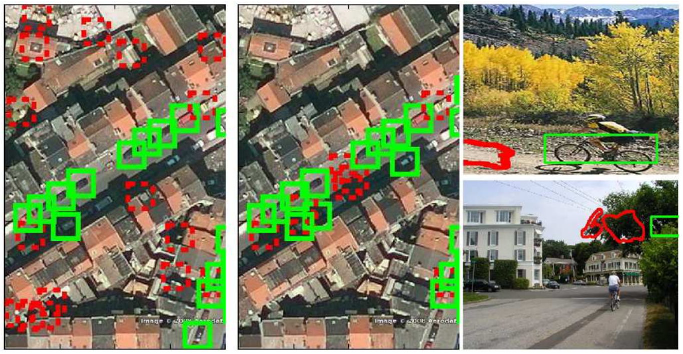


Figure 2.5: Detecting objects is improved by studying the spatial relationships between an object and its surrounding. Figure from [68]

2.2 Activity recognition frameworks

By contrast, global analysis for activity recognition has not been widely used. As explained in the introduction, this thesis proposes a framework for finding global explanations for all detections during a period of activity. This section reviews the influential research in the area of recognising complex activities. Prior to the review, different types of activities can be defined. An activity is said to contain *interleaved* events when, at a point in time, more than one event can occur. This disallows partitioning the period of activity temporally so only one event occurs within each partition. Figure 2.6 distinguishes interleaved from single-event activities. Moreover, activities can contain ordered or unordered events. Two events are *ordered* if they end in the same order they started. The need for defining *ordered* events arises from certain solutions for recognising multiple interleaved events that assume the detection is assigned to the event that started first, like queues or production lines. In these cases, the detection belongs to the earliest event expecting a detection of this type. Figure 2.6 also distinguishes ordered from unordered activities. Activities involving interleaved unordered events is the most general case.

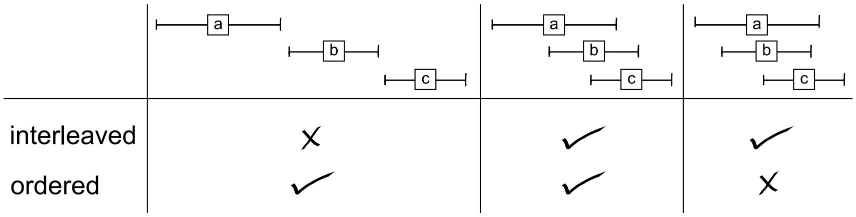


Figure 2.6: In a single-event activity, the timeline can be partitioned so one event occurs within each partition. Interleaved-event activity, on the other hand, expects more than one event at each point in time. In ordered activities, like queues, the event that ends first is the one that started first. In unordered activities, events can end in any order regardless of their starting order.

Generally, two kinds of events are distinguished. I will refer to these as primitive and compound events. A *primitive event* is an event that is detected directly and corresponds to one detection exactly. A primitive event thus labels the detection depending on the activity. For example, a trajectory detection could correspond to the primitive event of a person walking across the platform when analysing the activity at the train platform. A *compound event* is a grouping of other simpler, compound or primitive, events. In the literature, the phrases *compound/primitive events* are substituted with event/subevent [69, 108], compound/simple events [25], compound/atomic events, or the words are simply used interchangeably [139]. An *activity* is thus recursively defined as a composition of events, until primitive events are only available.

This thesis covers recognising activities with interleaved unordered compound events.

A framework designed to recognise such activities depends on the choice of detections. Previous work often used motion detectors to retrieve trajectories [40, 69, 77, 96, 108, 109, 120, 127]. Some of these researches assume all moving objects are of the same type like people [109] or cars [71]. Others used object detectors to classify trajectories like people detectors [120] or hand detectors [127]. Some detectors were domain-specific like detecting fridges and hobs [109] or even a glucose monitor detector using template matching [127]. Differently, low-level recognisers were modelled by hidden Markov models that retrieve temporally overlapping durations as detections along with a likelihood of the primitive events [74]. The framework would then contain two parts. The first is the definition part, where the activity is formally expressed, and its events are specified. The second is the recognition part for finding a consistent set of events, given the definition, for a finite set of detections. Though the framework requires both parts, this section explains each one separately to clarify the different approaches in the literature for each task.

2.2.1 Frameworks for defining activities

The work of Ivanov and Bobick [74] highlighted the importance of formal methods to encode expert knowledge for recognising activities in video. This is because the recognition expects a "rich knowledge base" to make out the possible explanations [18]. While learning the structure of the activity from noisy image sequences is hard, this structure is explicit and known in advance.

The decomposition of the activity into a set of events, which can be further decomposed into simpler events, is naturally represented by a hierarchy. Two different hierarchical representations are shown in Figure 2.7. In the literature, some define the activity by drawing those hierarchies [42, 69].

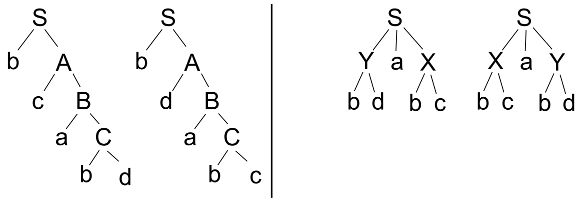


Figure 2.7: Hierarchical representations of the activity.

Grammars naturally define a hierarchy, and were used to define activities in video as early as in 1998 [148]. Different types of grammars can give rise to different hierarchical structures. The hierarchies on the left of Figure 2.7 can be represented by a regular grammar, while the ones on the right can be represented by a context-free grammar (which is more general) [5]. Regular grammars are used to define the class of languages accepted by finite state automata, while context free grammars define the class of languages that are accepted by push-down automata. Figure 2.8 shows regular and context-free grammars corresponding to the hierarchies in Figure 2.7.

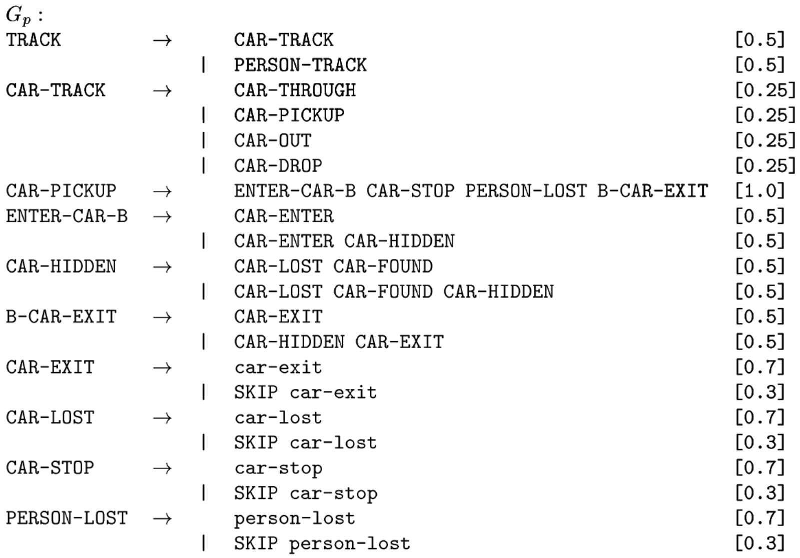
Regular Grammar	Context-Free Grammar
$\mathrm{S} \rightarrow \mathrm{bA}$	$\mathrm{S} \rightarrow \mathrm{XaY} \mid \mathrm{YaX}$
$\mathrm{A} \rightarrow \mathrm{cB} \mid \mathrm{dB}$	$\mathrm{X} \rightarrow \mathrm{bc}$
$\mathrm{B} \rightarrow \mathrm{aC}$	$\mathrm{Y} \rightarrow \mathrm{bd}$
$\mathrm{C} \rightarrow \mathrm{bc} \mid \mathrm{bd}$	

Figure 2.8: Regular grammar (left) can represent the hierarchies in Figure 2.7 (left), while Context-Free grammar represents the hierarchies in Figure 2.7 (right).

When used for recognising activities, regular grammars are suitable for modelling a series of parallel models [104], but as the number of variations increases it becomes harder to represent them using a concise finite state machine increases. For example, ball passes between players in a game of tennis can easily be modelled using a regular grammar, but in a football game a context-free grammar provides a more compact representation by allowing chains of passes of arbitrary length. A context-free grammar rule $A \rightarrow BbC$ rewrites a compound event into a sequence of primitive and compound events. Stochastic Context Free Grammars (SCFG) can be defined where a probability is associated with each rule indicating its preference over alternative rules. Ivanov and Bobick used SCFG to represent the different ways in which complex activities can be constructed, and assign probabilities to each [74]. They evaluated their approach on gesture recognition and surveillance within a car park. An example SCFG presented in their paper for the car pickup task is shown in Figure 2.9. They realised that SCFGs are not sufficient to define the valid explanations, and therefore added an additional consistency check enforcing *temporal constraints* that allow or prevent overlapping events. This is because the rule $A \rightarrow abB$ does not specify whether the events can overlap or not. They added this check to the recognition process, rather than the formal definition of the activity.

SCFGs have been intensively used since then to recognise different activities, like events in a blackjack game [104] and surveillance applications [44, 86]. The work of Zhang *et al.* augments the grammatical rule with a matrix of temporal relations R [152]. Each element r_{ij} in the matrix R defines the temporal relationship [7] between symbol i and symbol j in the rewritten string.

Non-temporal constraints, such as limits on the separation of objects involved in an event, can also be formally defined. Ivanov *et al.* textually describe the spatial constraints between the events in SCFG [75]. To provide such constraints as part of the activity's



```
G_p :
TRACK        →     CARE-TRACK                                     [0.5]
                |  PERSON-TRACK                                   [0.5]
CAR-TRACK    →     CAR-THROUGH                                    [0.25]
                |  CAR-PICKUP                                     [0.25]
                |  CAR-OUT                                        [0.25]
                |  CAR-DROP                                       [0.25]
CAR-PICKUP   →     ENTER-CAR-B CAR-STOP PERSON-LOST B-CAR-EXIT    [1.0]
ENTER-CAR-B  →     CAR-ENTER                                      [0.5]
                |  CAR-ENTER CAR-HIDDEN                           [0.5]
CAR-HIDDEN   →     CAR-LOST CAR-FOUND                             [0.5]
                |  CAR-LOST CAR-FOUND CAR-HIDDEN                  [0.5]
B-CAR-EXIT   →     CAR-EXIT                                       [0.5]
                |  CAR-HIDDEN CAR-EXIT                            [0.5]
CAR-EXIT     →     car-exit                                       [0.7]
                |  SKIP car-exit                                  [0.3]
CAR-LOST     →     car-lost                                       [0.7]
                |  SKIP car-lost                                  [0.3]
CAR-STOP     →     car-stop                                       [0.7]
                |  SKIP car-stop                                  [0.3]
PERSON-LOST  →     person-lost                                    [0.7]
                |  SKIP person-lost                               [0.3]
```

Figure 2.9: A car-pickup SCFG as presented in [74].

definition, different linguistic formulations have been proposed [69,108,120,128]. Nevatia *et al.* proposed the 'Event Recognition Language' (ERL) [108]. ERL is an ontology that includes a complex set of spatio-temporal relationships. It divides events into three types: primitive events that can be directly detected; single-thread events made up of one sequence of events; and multi-thread events where temporal, spatial and 'logical' relationships are allowed. The paper argues that activities can be defined more easily using this ontology than using stochastic grammars. The ontology does not only define events, but also allows defining the scene, regions of interest, occluders, etc. A predefined set of temporal, spatial and logical relationships is presented.

In Rota and Thonnat [120], an activity is defined as a four-tuple:

1. A set of positive events that should occur for the activity to be recognised, along with a set of negative events that should not occur.

2. Temporal constraints between positive events in the activity.

3. Non-temporal constraints, such as spatial relationships between the events or object sizes.

4. Any action that needs to be taken if the event was recognised. This is defined in the context of surveillance applications to raise a warning when needed.

The approach is applied to define certain activities in a metro station. An example of a defined activity is shown in Figure 2.10.

Name = “forbidden access to area”,
Events = $(t_1,\ enters(p_1 : Person,\ a_1 : Area))$,
not$(t_2,\ leaves(p_1 : Person,\ a_1 : Area))$,
Constraints = $t_1 \leq t_2,\ t_2 \leq t_1 + 1.0$,
Conditions = function(a_1, “forbidden_access”),
Success = alarm(p_1, “has entered area”,a_1)

Figure 2.10: A tuple defined for the activity of detecting a person in a forbidden area. Figure from [120]

A simpler approach by Chan *et al.* [25] defines positive and negative events; though is only suitable for two levels of hierarchy, where an activity is defined as a set of complex events that are directly decomposed into primitive events. A table is used to represent the domain’s knowledge, where rows represent primitive events and columns are the compound events representing consecutive states of the activity. A cell in that table is labeled 0 if the primitive event is not allowed, 1 if the primitive event is required, and is left empty if the compound event is indifferent to the detection of this primitive event.

The work of Siskind is based on the assumption that the world is made up of lines, and thus lists general spatial relationships like ‘supported’ and ‘attached’ [128]. An event is then recognised as a logical expression made up of spatio-temporal relationships to govern the interacting objects. The work though expects each object to be detected and tracked correctly. This approach was later used by Ersoy *et al.* to query a database of primitive spatio-temporal relationships for interesting events [40].

Intille and Bobick defined multi-agent activities as sets of compound and fundamental (i.e. primitive) goals (i.e. events) with temporal and logical constraints governing the relationships [72]. The activity is viewed as a ‘partial set’ of goals, where temporal relationships are identified between some of the goals. Logic constraints, like ‘or’ and ‘xor’ relations, are added to the definition when needed. The technique models interactions of players in American football. A collection of ‘plays’ are defined by an expert, and the definition is then mapped to a Bayesian network that links the partially-ordered events defining causality and allowing for parallel relations. The same approach was used by Shi *et al.* to define activities [127] (Figure 2.11). In addition to training the probabilities and the observation likelihoods, a Gaussian models the time elapsed for each event.

The recent work of Tran and Davis [139] uses first-order logic production rules to encode the domain’s knowledge. Four rule types are used: definite clauses which are hierarchical decompositions of activities into events; disjunctions which provide alternative explanations; negative preconditions which are constraints on applying the rules; and exclusion relations which model relationships between events. The work provides an in-

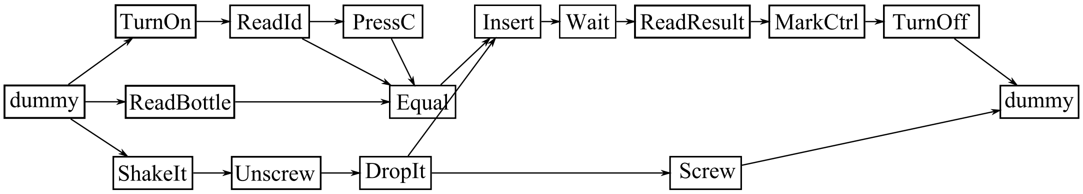
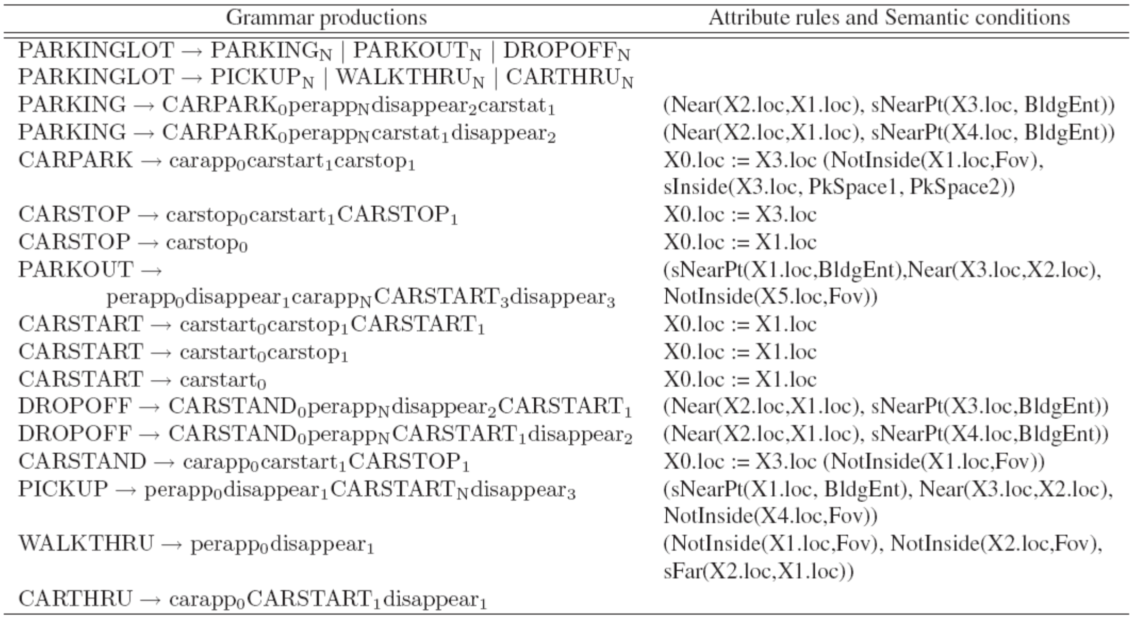


Figure 2.11: The activity of glucose calibration is represented by a Bayesian network. Figure from [127]

sight into constraints between events occurring at the same time. For example, a person belongs to only one group of walking pedestrians at a time, or a person drives only one car. These constraints are modeled using exclusion relations in this work. Some of the rules presented in their approach for activities in a car park are not intuitive to think of, like: ‘if a person opens the trunk of the car, he/she will (likely) enter that car’, or ‘two persons shaking hand with each other will (likely) not enter the same car’. They extend beyond the hierarchy of events. A simple hierarchy cannot relate the parking event to hand shaking. Weights are assigned to the clauses to differentiate hard from soft constraints, and imply rule preferences. Tran and Davis introduce logic rules because stochastic grammars are incapable of defining constraints.

Attribute grammars are one way to define constraints within a grammar formulation [87]. These have recently been used to recognise activities in a car park by different authors [77, 78, 96]. This follows previous success in using attribute grammars to constrain the spatial relationships in visual languages [55,102] and the detection of objects in images [62,157]. Attribute grammars allow defining attributes to accompany terminal and nonterminal symbols, and defining constraints that govern the allowable values of those attributes (more in Section 3.2). Using attribute grammars, attribute rules and constraints are incorporated into the grammar. The previous approaches in [77,78,96] do not employ the full abilities of attribute grammars to define rules and constraints. Attributes are only sparsely defined, while our approach incorporates attribute rules that evaluate the likelihoods for all events at higher levels in the hierarchy, and constraints between dependent events. Figure 2.12 shows a sample attribute grammar for the car park from [77]. The approach rewrites a nonterminal as a string of symbols. The grammars in this thesis rewrite a nonterminal as a multiset, and only introduce temporal relationships as constraints on valid interpretations. This avoids multiple rules that only differ in the ordering of symbols such as the rules rewriting the PARKING event in the figure. Moreover, the grammar in Figure 2.12 does not define how the events can be shared when multiple interleaved events are to be recognised. A car can pick up multiple people, while a person cannot be picked up by multiple cars at the same time. These approaches also differ from the work



Grammar productions	Attribute rules and Semantic conditions
$\mathrm{PARKINGLOT} \rightarrow \mathrm{PARKING_N} \mid \mathrm{PARKOUT_N} \mid \mathrm{DROPOFF_N}$	
$\mathrm{PARKINGLOT} \rightarrow \mathrm{PICKUP_N} \mid \mathrm{WALKTHRU_N} \mid \mathrm{CARTHRU_N}$	
$\mathrm{PARKING} \rightarrow \mathrm{CARPARK_0 perapp_N disappear_2 carstat_1}$	(Near(X2.loc,X1.loc), sNearPt(X3.loc, BldgEnt))
$\mathrm{PARKING} \rightarrow \mathrm{CARPARK_0 perapp_N carstat_1 disappear_2}$	(Near(X2.loc,X1.loc), sNearPt(X4.loc, BldgEnt))
$\mathrm{CARPARK} \rightarrow \mathrm{carapp_0 carstart_1 carstop_1}$	X0.loc := X3.loc (NotInside(X1.loc,Fov), sInside(X3.loc, PkSpace1, PkSpace2))
$\mathrm{CARSTOP} \rightarrow \mathrm{carstop_0 carstart_1 CARSTOP_1}$	X0.loc := X3.loc
$\mathrm{CARSTOP} \rightarrow \mathrm{carstop_0}$	X0.loc := X1.loc
$\mathrm{PARKOUT} \rightarrow \mathrm{perapp_0 disappear_1 carapp_N CARSTART_3 disappear_3}$	(sNearPt(X1.loc,BldgEnt),Near(X3.loc,X2.loc), NotInside(X5.loc,Fov))
$\mathrm{CARSTART} \rightarrow \mathrm{carstart_0 carstop_1 CARSTART_1}$	X0.loc := X1.loc
$\mathrm{CARSTART} \rightarrow \mathrm{carstart_0 carstop_1}$	X0.loc := X1.loc
$\mathrm{CARSTART} \rightarrow \mathrm{carstart_0}$	X0.loc := X1.loc
$\mathrm{DROPOFF} \rightarrow \mathrm{CARSTAND_0 perapp_N disappear_2 CARSTART_1}$	(Near(X2.loc,X1.loc), sNearPt(X3.loc,BldgEnt))
$\mathrm{DROPOFF} \rightarrow \mathrm{CARSTAND_0 perapp_N CARSTART_1 disappear_2}$	(Near(X2.loc,X1.loc), sNearPt(X4.loc,BldgEnt))
$\mathrm{CARSTAND} \rightarrow \mathrm{carapp_0 carstart_1 CARSTOP_1}$	X0.loc := X3.loc (NotInside(X1.loc,Fov))
$\mathrm{PICKUP} \rightarrow \mathrm{perapp_0 disappear_1 CARSTART_N disappear_3}$	(sNearPt(X1.loc, BldgEnt), Near(X3.loc,X2.loc), NotInside(X4.loc,Fov))
$\mathrm{WALKTHRU} \rightarrow \mathrm{perapp_0 disappear_1}$	(NotInside(X1.loc,Fov), NotInside(X2.loc,Fov), sFar(X2.loc,X1.loc))
$\mathrm{CARTHRU} \rightarrow \mathrm{carapp_0 CARSTART_1 disappear_1}$	

Figure 2.12: Attribute grammar for car parking scenario. Figure from [77]

in this thesis in the recognition methods as will be explained in Section 2.2.2.

Other approaches define the activity using graphical models. A Hierarchical Hidden Markov Model (HHMM) was used in [109] for modelling activities in a domestic environment. These are more suitably learnt rather than defined by a human expert. Gong and Xiang learn the temporal and causal dependencies between events using Dynamic Multi-linked HMMs [56]. As opposed to the other frameworks in this section, this work used unsupervised learning for activity definition. The approach learns causal and temporal relationships from videos of loading and unloading planes. The number of possible dependencies in the BN is limited using the Bayesian information criterion (BIC). The emerging structure of the BN would then be used to define the activity, along with the entries in a state transitions matrix.

Most of the previous work for activity recognition distinguishes between temporal and non-temporal constraints [56, 74, 108, 120, 128]. In fact, time can just be treated as another attribute in the framework - temporal and non-temporal constraints need not be made distinct. For example, given two events a and b, where t is an attribute that signifies time and c is an attribute for position, then constraints like $a.t < b.t + 10$ and $|a.c - b.c| < 25$ can be treated in the same way. Moreover, a general list of spatial and temporal relationships does not need to be gathered in advance, given the difficulty in compiling such a list. The framework proposed in this thesis treats all types of constraints in the same way, and allows defining any relationships between the events. By unifying the method of defining temporal and non-temporal constraints, the sequencing constraints can be dropped from the grammar. This thesis uses multiset grammars where a nonterminal is rewritten as a multiset of other symbols. Previous work based on string grammars had to provide solutions to resolve cases when the ordering is not strict or events can occur in parallel. This is because temporal constraints are enforced in string grammar in all cases, even when no temporal ordering of the events is required.

Apart from [96, 109, 139], all the frameworks presented above recognise one event given a video sequence. Typically, one video sequence involves multiple interleaved events. Defining activities with interleaved events should include defining the constraints between the events. In [139], first order logic captures these constraints. Lin *et al.* [96] and Nguyen *et al.* [109] assume each detection participates in one and only one event. This may be an incorrect assumption for some activities, e.g. for the activity of cars picking-up individuals, the pick-up event involves a car stopping, the person approaching then disappearing close to the car, followed by the car's departure. As the car can pick up several people, the detected car can be shared by multiple picking-up events. A person can though be picked up by one car. The formal definition should consider these constraints between the recognised events to provide a consistent set of events.

After formally defining the activity, this definition can be used to recognise activities. The next subsection reviews techniques used for activity recognition.

2.2.2 Activity recognition methods

Recognising a previously-defined event is the task of finding one or more instances of that event in a given video input, or indicating that such an instance is not present. The recognition technique is thus dependent on the way the event has been defined.

Assuming a SCFG is used to define the activity, a probabilistic parser can be used for the recognition. One efficient parser, referred to as the Early-Stolcke parser, can parse probabilistic production rules and find the parse with the highest probability [135]. The parser uses top-down dynamic programming performed in cycles of three tasks: predicting, scanning and completion until the input sequence is fully scanned. At the prediction stage, the set of all possible productions is accumulated. The scanning then reads the input and calculates the probability of the produced string. Finally, the completion step is performed when all the symbols in the production rule are successfully scanned. Parsing a string of primitive events can then be performed by this parser given a SCFG.

Such recognition has two underlying assumptions. The first ignores the uncertainty in detecting primitive events. Often detections are ambiguous and the primitive events can only be probabilistically defined. The second assumption is expecting only one compound event within a given input video. Previous work has attempted to drop one or both of these assumptions.

The uncertainty of the input can be resolved independently from the recognition task, where Maximum a Posteriori (MAP) assigns a primitive event to each detection. Alternatively, incorporating the uncertainty in the recognition task can resolve local ambiguities. In [74], the recognition is decoupled into two stages. First, hidden Markov models (HMMs) are used to detect primitive events. The likelihood of each primitive event is retained and used in the parsing process. A modified Earley-Stolcke parser generates the parse with the highest posterior probability given a sequence of uncertain events and the SCFG. During scanning, the posterior is calculated as the multiplication of the rule's prior probability and the events' likelihood terms. Three types of errors in the input have to be dealt with. Insertion errors arise when one of the detected events is actually a noisy observation or does not belong to the activity. Substitution errors occur when a detection is misclassified, and the actual primitive event is not detected as the most likely one. Deletion errors occur when a primitive event fails to be detected altogether. When the parser fails to parse the given input, it attempts to correct some for these errors, before running the parser again. The method also checks for temporal constraints. During the completion step of parsing, the parser rejects parses that do not satisfy the constraints. Ivanov and Bobick recognise a single compound event, involving one or more interacting agents, in each given video.

While Ivanov and Bobick only correct for errors when the input fails to be parsed, Moore and Essa [104] expand the approach and modify the input to accommodate for possible insertion/deletion/substitution errors even when the current input can be parsed correctly. The parse with the highest probability is found by maintaining multiple hypotheses at a time. At each step, the possible three errors are considered, and different parses are generated. The work discusses pruning the hypotheses to avoid growing complexity, yet in their work exhaustive search was tractable given the small number of detections.

Kitani *et al.* build a hierarchical Bayesian network from the SCFG [86]. Probabilities are embedded in the hierarchical Bayesian network. Instead of a parser, deleted interpolation is used to find the explanation with the maximum posterior. In 'deleted interpolation', the probability distribution at each point in time is calculated as a weighted sum of explaining partial evidences over a window of size l. A solution that strongly explains recent observations is favoured. Unlike [74, 104], they do not incorporate the uncertainties in recognising the primitive events into the approach. The probabilities are only confined to priors of the grammar rules. Though the paper argues that activities are 'constrained and temporally overlapped', no explanation was provided on how the constraints were satisfied.

Shi *et al.* use discrete condensation [127] for finding the best explanation using their P-Net representation. They modify the condensation algorithm [73] to sample a discrete search space, and refer to this as discrete condensation. They compare discrete condensation with the parsing from [74] and present results that demonstrate discrete condensation has a higher capability of recovering from errors and uncertainties in the data.

Hongeng *et al.* build a Bayesian network so primitive events are independent, and compound events are conditionally dependent on the simpler events [69]. The posterior of the Bayesian network is evaluated using belief propagation in one direction, from the bottom layer to the top layer. The joint probability of primitive and compound events is thus simplified to that in Figure 2.13. The approach then compares $p(H|e_1, e_2, e_3)$ with $p(\neg H|e_1, e_2, e_3)$. The same independence assumptions for the joint probability are used in [98]. Hongeng's novel framework recognises one compound event given each sequence. It exhaustively searches the possible combinations of primitive events to find the one that maximises the posterior. The method presented in this thesis adopts the same independence assumptions as these in Figure 2.13. This will be further explained in Chapter 3.

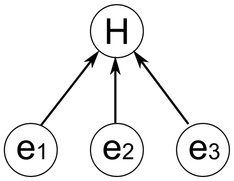


Figure 2.13: In [69], primitive events are assumed independent and compound events depend on their primitive events. Graphical model from [69]

Similarly in [108], the event with the least uncertainty is recognised by finding the combination of primitive events that satisfies the temporal constraints with the highest likelihood. The paper suggests pruning methods to limit the complexity of the approach, but it focuses on formulating the problem rather than solving the recognition task.

Intille and Bobick automatically build a Bayesian network and link each event to an observed node [72]. All the observed nodes are binary or ternary. An observed node is labeled as (yes/maybe/no), which does not probabilistically incorporate the underlying uncertainty. When applied to the activity of American football, multiple Bayesian networks 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, which suits the context of a football game, where one strategy is present at a time.

Despite the majority of activity recognition frameworks focusing on recognising a single instance of the compound event given a separated set of detections, some recent work deals with the more realistic situation where a complete set of detections, belonging to different events within the activity, is available. Chan *et al.* argue that joining tracklets into complete trajectories can benefit from recognising the events performed by each tracklet [25]. Applied to plane refueling activities, a motion tracker yields broken tracklets representing the movements of different actors (e.g. person, hose, plane). A combined approach is sought where tracking and activity recognition are decided-upon jointly. The work builds a dynamic Bayesian network, then uses brute force to search through the set of possible explanations. Though this framework is very suitable for jointly recognising primitive and compound events, it expects one compound event at a time, which suits plane refueling scenes. It cannot be used to recognise interleaved events.

Recognising interleaved -yet ordered- activities, like a cashier scanning items one at a time, is achieved in [42] using a special Viterbi algorithm. Ordered activities expect events to end in the same order they started, which suits the events at a point of sale. The approach is though unsuitable for unordered activities.

Tran and Davis use Markov logic networks, built using first-order logic rules from the activity definition [139]. Observed events are grounded and a recursive procedure adds new ground atoms using the logic rules to the Markov logic network. Inference is then performed using Gibbs sampling with simulated annealing.

A recent attempt to recognise interleaved unordered events is that of Joo and Chellappa [77, 78]. Similar to Ivanov and Bobick's work, HMMs are used to recognise primitive events, and parsing recognises the compound event satisfying the constraints and considering the uncertainty of the primitive events. To recognise interleaved events, multiple threads are maintained and detections are greedily assigned to threads. The resulting explanation is not necessarily one that maximises the joint posterior of the activity, as detections are assigned independently in a sequential order.

Nguyen *et al.* proposed a framework to assign detections and recognise interleaved events [109]. The authors acknowledge that a reliable assignment of detections to events is often unavailable. The proposed approach splits the tasks into two. First, detections are partitioned into events. Then, multiple hierarchical hidden Markov models (HHMM) are used to recognise the events. This though assumes the number of events is fixed and known in advance, in order to decide on the number of HHMMs. Assigning detections uses the Joint Probabilistic Data Association Filter (JPDAF). This maximises the joint probability of assigning all detections to events at each point in time. A combined HHMM-JPDAF is presented using a dynamic Bayesian network (Figure 2.14). The approach uses MCMC to sample from the set of possible assignments, then exact inference is used for each HHMM. Though the problem solved by Nguyen *et al.* is the closest to the problem posed in this thesis, the number of events cannot be reliably known in advance. In [109], the assignment was not formally defined, and is simply a 1-1 assignment in the discussed cases.

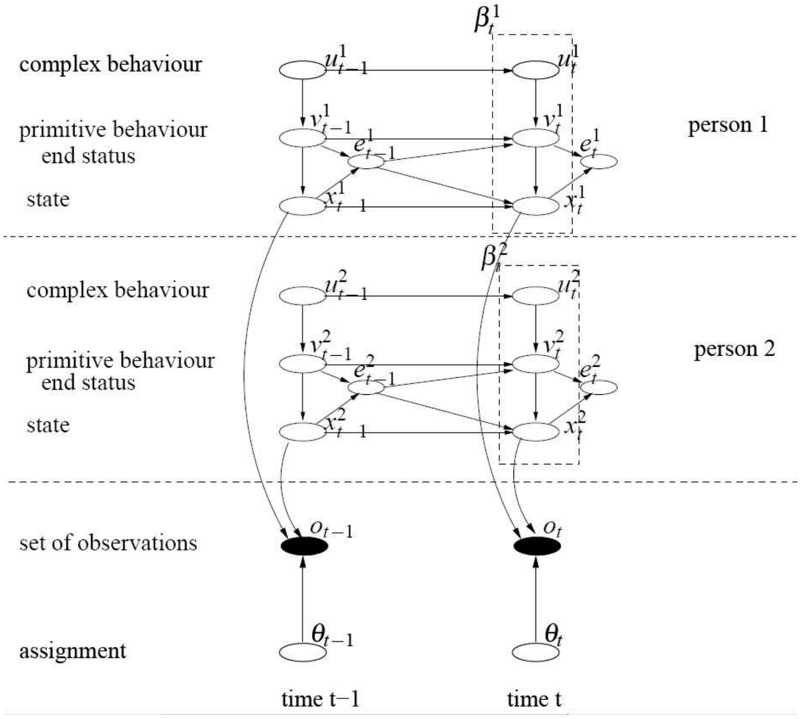


Figure 2.14: A DBN representing the HHMM-JPDAF in the case of two compound events. Each one is represented by a Hierarchical HMM. The assignment of detections to events is performed separately at each time step. Diagram from [109]

A recent attempt to overcome an assumed partitioning of detections into events combines SCFG with a Markov Random Field (MRF). The MRF is defined as a joint probability on nodes in the possible parse trees. The unary term defines the primitive 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 does not take into consideration the constraints between events. As previously explained, this could lead to an inconsistent set of events. For example, a car can drop-off several people, yet a person can be dropped off by only one car. The MRF should be aware of such constraints when sampling from the list of candidate objects.

This section highlights the need for a framework that defines and recognises activities taking into consideration not only the temporal and spatial constraints within the events, but also the constraints between the events. As attribute grammars have been used successfully for defining spatial and temporal constraints, they will be adopted in the suggested framework. For recognition, a Bayesian approach, similar to [69] has been extended to jointly recognise the complete set of events within a period of activity.

2.3 Data association for tracking

Section 2.2 explained how recognising interleaved events has been previously tackled in the literature. When interleaved events are expected but their number is unknown, and constraints between the events should be satisfied, the recognition task involves a data association process. Data association maps detections to a previously unknown number of identities, in this case - events. The mapping should satisfy the association constraints. As explained in [95], data association has two components, a similarity measure which favours some associations over others, and an association optimisation method which finds the best association satisfying defined constraints. Data association has been employed often in tracking to assign detections or measurements to objects. This section reviews proposed solutions for three relevant problems from the tracking literature: multi-target tracking within radar surveillance, intra-camera visual tracking and inter-camera visual tracking. In all these problems global consistent associations have been used to resolve uncertainties and improve tracking performance.

2.3.1 Multitarget radar tracking

The problem of data association for detections from radar and similar sensors is explained using the following example. Assume a radar periodically scans for aircraft in a specified area. Detections represent aircraft as well as false alarms. Figure 2.15 shows the detections at times $t-1$, t and $t+1$. The detections are recorded asynchronously, as such sensors require a specified time to scan the observed area before starting a new scan. The data association problem tries to group those detections into trajectories, identifying any false alarms. It assumes targets move independently according to a Markovian process [112]. A target can appear at any point in time, persist for a random duration, then disappear. The task would be to partition the detections into trajectories representing targets. Each detection at time t represents one target at most. If the detection is not part of any trajectory, it is thought to be a false alarm. At least two detections are expected for a trajectory to be established. Alternative variations of the radar problem expect at least n

detections before a trajectory is considered.

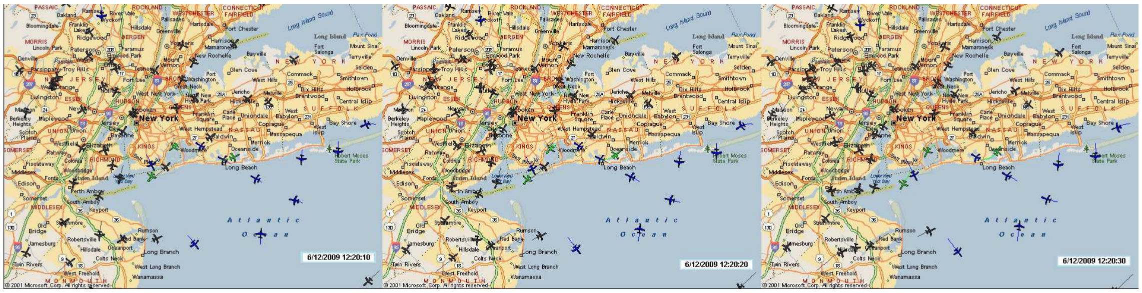


Figure 2.15: Three images from Airport Monitor$^{\text{TM}}$ *2.0 (Copyright of PASSUR-AEROSPACE www.passur.com) covering JFK Airport area within a range of 40 miles on the 12*$^{\text{th}}$ *of June 2009 at 12:10, 12:20 and 12:30.*

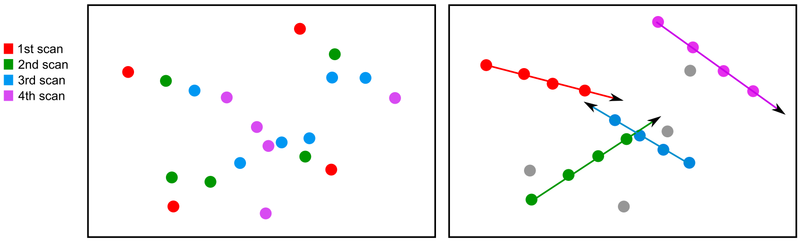


Figure 2.16: An abstract 4-scan example of multi-target tracking.

As the detections are not visually distinguished from each other, this task is referred to as the ‘motion correspondence’ task. Given these indistinguishable detections, distances and velocities must be used to resolve ambiguities in the partitioning process. Though the search space of all possible partitions is huge, the difficulty in the motion correspondence task is not measured by the number of detections, but by the ambiguity in the partitioning process. Even if the number of detections is vast, but each target is moving far enough from other targets, the task would be considered trivial, and simple Kalman filtering [9] would be sufficient. The uncertainty arises from dense detections, and a high rate of false detections [112]. When the ambiguities increase, researchers in the radar domain proposed techniques that rely on deferred logic [36], where the decision could be amended by future scans. In deferred logic, detections within a sliding window are analysed and the best global explanation is considered. Figure 2.16 presents a 4-scan example along with the corresponding correct trajectories, and detected false alarms.

The paper ‘A Review of Statistical Data Association Techniques for Motion Correspondence’ by Cox in IJCV(1993) lists the various techniques for data association used to solve the radar problem until then [28]:

- Nearest Neighbour: matches each detection at time t to its nearest neighbour at time $t-1$.

- Track Splitting Filter: Instead of taking the decision for each consecutive pair of scans, this technique splits the trajectory into the best two possible explanations. Branching is performed independently for each track. This method does not ensure disjoint tracks. The solution can associate a single detection to two separate tracks.

- Joint Probabilistic Data Association Filter (JPDAF): At each scan, the joint probability for assigning new detections to trajectories, given the previous assignment, is considered. The JPDAF does not change the assigned trajectories for previous scans and expects a fixed number of trajectories.

- Integer Programming: In 1977, Morefield formulated the radar problem as a set packing task, and solved it using integer programming [105]. The set of all possible trajectories [1] is accumulated, along with the probability (or cost) for each trajectory. The trajectories in this set are not disjoint, as the same detection is assigned to multiple trajectories. Set packing then creates hypotheses, where the trajectories in each hypothesis are disjoint and all detections are explained. Integer programming is used to find the hypothesis with the highest probability. This technique performs an exhaustive search through the space of explanations.

- Multiple Hypotheses Tracking (MHT): Reid proposed a heuristic search using the multiple hypotheses tree (MHT) [116]. Reid's tree has a number of levels that equals the number of scans. At each level, the detections at the current scan are assigned to existing or new targets. For each branch in the tree, constrained explanations for the current scan are added as children nodes to the branch. Notice that the set of possible explanations differs between branches depending on previous scans. As the tree grows exponentially, it is pruned and the k-best explanations are retained at each level. This search is heuristic, as it cannot be guaranteed in advance that the correct assignment will remain within the k-best hypotheses as future scans are considered. Increasing k though increases the required calculations and memory resources. Cox re-formulated the problem, using an earlier work of Murty [106], to find the k-best hypotheses in polynomial time without enumerating all the assignments [29]. The technique uses the Hungarian algorithm and amends the cost matrix to block the best-solution's assignments.

Recent solutions to the radar multi-target problem use MCMC to find the optimal association. Oh, Russell and Sastry introduced MCMCDA (MCMC Data Association) for multi-target tracking [111, 112]. In Oh *et al.*'s work, given a set of detections Y, the

[1] given a maximum distance between detections in subsequent scans - this is known as gating

search is for the best association $\hat{\omega}$ that maximises the posterior $p(\omega|Y)$. By defining the set of associations Ω, a Makrov chain is constructed to sample the space of associations. At each step in the Markov chain, a new association is proposed by applying a move to the current association. MCMCDA is further explained in Section 4.4.1. The set of reversible moves proposed in the paper for multi-target tracking are shown in Figure 2.17.

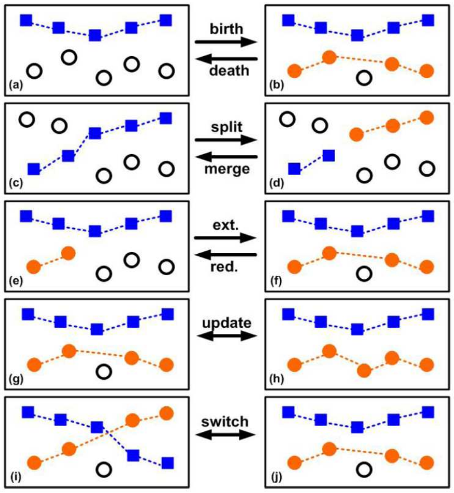


Figure 2.17: The set of reversible moves proposed by Oh et al. for the multi-target tracking problem. Diagram from [111]

2.3.2 Intra-camera global tracking

Visual tracking is the task of associating detections, retrieved from visual sensors like CCTV cameras, to form complete trajectories. It differs from the multi-target tracking problem introduced in Section 2.3.1 in that appearance can be used to relate detections, and distances are affected by the unknown depth of the view field. This section reviews techniques that employ global analysis to achieve better intra-camera tracking. Broken trajectories, tracklets and noisy detections have to be connected into complete trajectories. Traditionally, detections are associated by considering a couple of frames. A recent trend toward global solutions, despite the combinatorial complexity, uses approaches such as multiple-hypotheses trees [13, 24, 81], cost-flow networks [152], Bayesian network inference [79], Expectation-Maximisation [150, 156], quadratic Boolean optimisation [94], dynamic programming [14] and linear programming [126].

The closest form of intra-camera visual tracking to that of multi-target tracking is tracking ants and bees [85], because the detections are indistinguishable. Khan *et al.*'s work [84, 85] tested the ability to track ants and bees within a closed environment, where the number of targets is fixed, as well as an open environment where ants can leave the field of view via an opening and return again. MCMCDA was used to sample the space of global explanations. The recent work of Zou *et al.* [158] tries to establish 3D trajectories from stereo data of fruit fly swarms. A global approach is used where the trajectory is defined as a sequence of stereo correspondences between the image projections across the entire duration. To accommodate for the combinatorial complexity, the approach uses Gibbs sampling to sample the set of possible correspondences, and the optimal global explanation is found using dynamic programming.

Pedestrian tracking associates foreground segmentations, often represented by blobs, to form trajectories. Sampling the distribution of possible trajectory assignments has been increasingly employed in tracking pedestrians using importance sampling [147] or MCMC sampling [2,131,149,154]. Zhao and Nevatia's work is a novel work in this area, where the best interpretation of all detections in a video sequence is found by Bayesian inference [154]. The work reformulates the intra-camera tracking task as the estimation of the number of objects, the correspondence between the objects in consecutive frames, and the positions of those objects. The paper uses MCMC for sampling the possible explanations, and highlights the importance of 'informed' proposal distributions (referred to as 'weighted' proposal distributions in Section 4.4). The work assumes each blob belongs to a single trajectory, and each target is represented by a maximum of one blob at each frame.

Smith [131] uses Reversible Jump MCMC (RJMCMC) for the same task. Smith's thesis discusses how RJMCMC, proposed by Green [57], is suitable for sampling the joint distribution of target numbers and their positions. Tracking is performed in a sliding window, and the globally optimal trajectories are computed for each window independently. Building on this, Yu *et al.* [149] combine segmentation along with tracking. As the same target can be split into several blobs during tracking, or the same blob can be composed of multiple targets, this work merges and splits blobs to find global trajectories. They model both spatial and temporal moves (extending those of Smith), and search the space of explanations within a sliding window. Figure 2.18 shows the moves suggested in [149].

While all the presented techniques provide an explanation for all the detections, up to the current time stamp, some approaches postpone the decision until the data is dis-

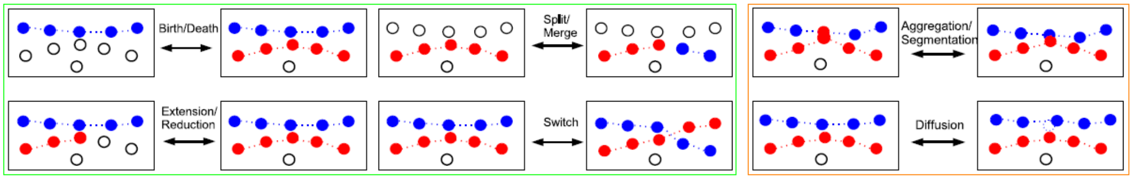


Figure 2.18: Spatial and temporal moves for intra-camera tracking. Diagram from [149]

ambiguated [124]. Ambiguous trajectories are flagged, and are only explained when the uncertainty can be confidently disambiguated.

2.3.3 Inter-camera global tracking

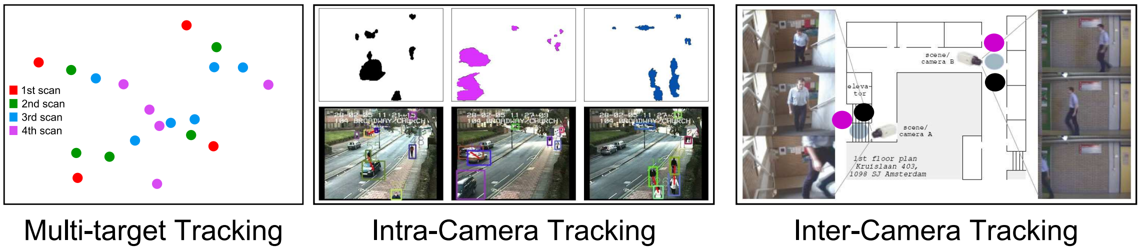
Global analysis for trajectories in non-overlapping cameras has previously been used to relate entry and exit points in camera views, and to track individuals across blind regions [14, 19, 76, 101, 155]. The work related to this problem can be divided into three categories. In the first category, the topology of a network of cameras is established without directly associating the detections [97, 101, 133]. This category does not include a data association task. The second category aims at establishing the correspondences for a given camera topology. The third hybrid category finds the topology along with establishing correspondences between detections.

For the second category, features of the pedestrians, referred to as passive [52] or soft-biometrics [141, 146], are compared to assess whether two detections correspond to the same person. Most of these features are session-based, i.e. they might differ for the same individual if observed at a later point in time. Clothing colour is a common matching feature to connect two trajectories as it is easy to retrieve [19,52,53,76,129,146]. Other passive features have been used, like texture [59], height [70, 99] and gait [63, 110]. In solving the data association task, one-to-one assignment has generally been assumed [19, 83, 151], and a greedy search [70, 129] or the Hungarian algorithm [83] have been employed to find the best assignment.

The work by Zajdel *et al.* is one example of the hybrid approach [150], as it finds the topology and connects the trajectories. It considers all the detections and builds a dynamic Bayesian network. Expectation-Maximisation (EM) is used to retrieve the BN structure that best suits the detections, and the parameters of that structure.

Inter-camera tracking becomes more complex when new people can appear anywhere across the network, and people can depart at any blind area. One of the earliest solutions to this complex inter-camera tracking was introduced by Huang and Russell [71], as part of ‘Roadwatch’ for tracking cars across wide-area traffic scenes. They assign each car seen upstream to its corresponding observation downstream, allowing for on-ramp and off-ramp detections. Their solution uses MHT, thus it cannot scale to tracking cars between more than two cameras due to the growing complexity. An MCMC sampling approach is proposed for a scalable solution [113].

Figure 2.19 provides an example that shows how multi-target tracking, intra-camera and inter-camera tracking can be perceived as different forms of the data association problem. Global optimisation techniques like multiple-hypotheses tracking or sampling using MCMC can be employed for data association. The search is for the best global explanation that associates all the detections.



Multi-target Tracking Intra-Camera Tracking Inter-Camera Tracking

Figure 2.19: The three different data association problems are shown. In each problem, the detections are partitioned into a previously unknown number of targets or considered false detections. Intra-camera diagram taken from [149] and inter-camera diagram from [150]

2.4 Summary

This chapter reviewed some of the previous work related to global analysis for activity recognition and data association. As the thesis is proposing global explanations for activity recognition, a quick overview of global analysis in computer vision was first presented. Global analysis assists resolving local ambiguities by considering hard and soft constraints.

A collection of previous frameworks for activity recognition was discussed. For each framework, the method to define the activities was first explained, followed by the recognition technique. Global explanations of activities require not only recognising all events, but also partitioning the detections into the activity's events. This is a data association task. A review of data association for tracking was presented for three tracking problems: multi-target tracking of radar detections, intra-camera visual tracking and inter-camera visual tracking. The next chapter introduces the framework presented in this thesis to find global explanations for activity recognition.

Chapter 3

Global Explanations for Activity Recognition

Analysing an activity involves recognising a consistent set of events. While most existing activity recognition techniques deal with a single event, realistic surveillance typically involves interleaved unordered events, extending over a long temporal duration. In these situations, the events are often mutually dependent. For example, a person entering a building can be observed departing only once at a later time. In visual analysis, these dependencies can be exploited to disambiguate uncertain visual data by seeking a global explanation.

This chapter presents a complete framework that starts with a general way to formalise the set of global explanations for a given problem using attribute multiset grammars. **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 approach that considers the prior probability along with the likelihoods of the recognised events. To find the best parse tree given a set of detections, the approach is accompanied with an algorithm that transforms the grammar and a finite set of detections into a Bayesian Network (BN). The set of possible labellings of the Bayesian network corresponds to the set of all parse trees for the given set of detections. The best global explanation is the Maximum a Posteriori (MAP) solution over the space of explanations.

3.1 Activities as hierarchies of events

As explained in Section 2.2, an activity is a set of related events, which can be recursively defined as sets of simpler events until primitive events are reached. For a chosen activity, the composition of the activity forms a hierarchy. Consider the activity in a car park, Figure 3.1 shows a plausible decomposition into event types. In addition to cars and people passing by, cars can be left in the parking area and retrieved later. Six types of primitive events are expected in this activity - these are the leaves of the tree in Figure 3.1. In addition to directly detecting these primitive events, compound events need to be recognised by grouping simpler events. For example, to combine detections of a car stopping with a person moving away as a ‘leaving-car’ event, the person must emerge close to the right frontal door of the car. Similarly, to combine a ‘leaving-car’ event with a ‘retrieving-car’ event, the same car (parked at the same spot for example) should be detected in both events. The hierarchy in Figure 3.1 represents possible event types. The activity will actually include multiple interleaved events of these types.

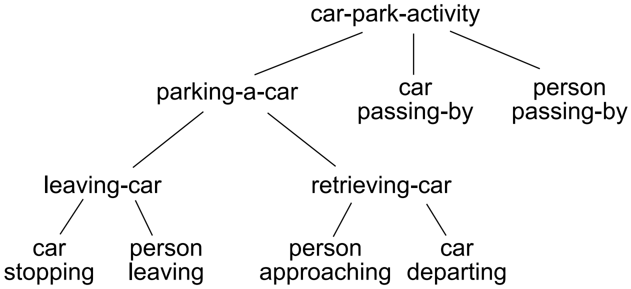


Figure 3.1: The activity in a car park is represented as a hierarchy of compound and primitive event types.

Figure 3.2 shows an illustrative timeline for a set of detections in a car park (5 car detections and 6 person detections). The bar shows the temporal extent of each detection, for example the temporal extent of a car stopping starts from the moment the car

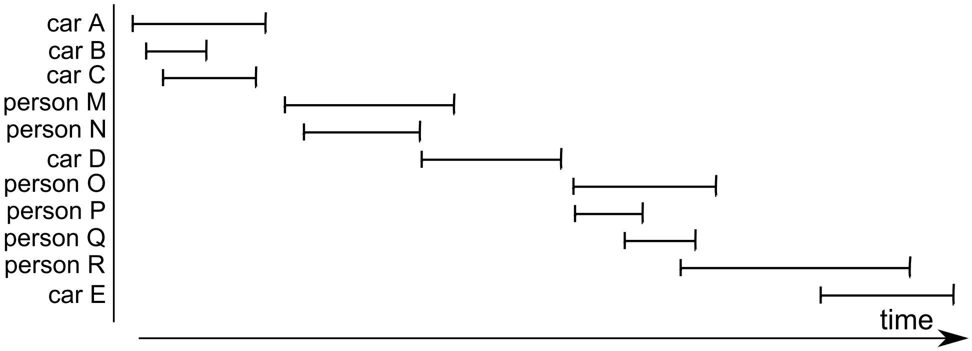


Figure 3.2: Five cars and six people detected in a car park.

appears until it fully stops. Given this set of detections, and the expected activity hierarchy (Figure 3.1), a global explanation partitions all detections into a consistent set of compound and primitive events. Figure 3.3 represents an example of such an explanation. This global explanation contains a set of five events, three of which are further defined as a set of simpler events. Some compound events might not be complete, like the third car parking, as there was no observation of the car being taken away.

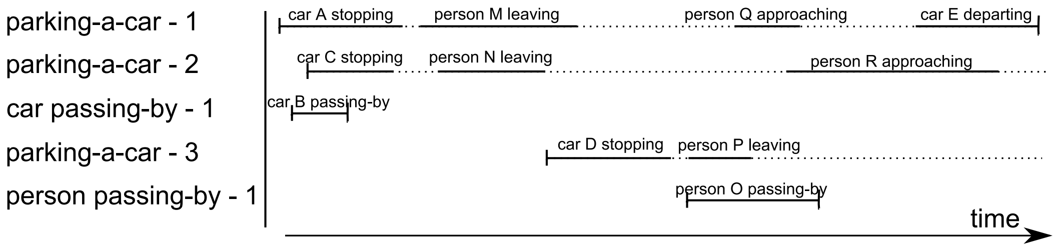


Figure 3.3: A global explanation for interleaved unordered events. Each row represents one event in the activity. Dotted lines show the temporal gaps between events.

Figure 3.4 expresses the global explanation in Figure 3.3 as a hierarchy. Each row in Figure 3.3 corresponds to one of the sub-hierarchies of the activity. The left-right order of the events in the tree is irrelevant. Accordingly, each node in the hierarchy is a set of its subordinates, rather than a tuple. Using sets, instead of tuples, simplifies the definition, as many compound events can be carried out in different orders. Defining the event as an ordered tuple would require multiple tuples for the different possible orders. When sets are used, only one set can represent the various cases. Temporal constraints can still be defined, but only when needed.

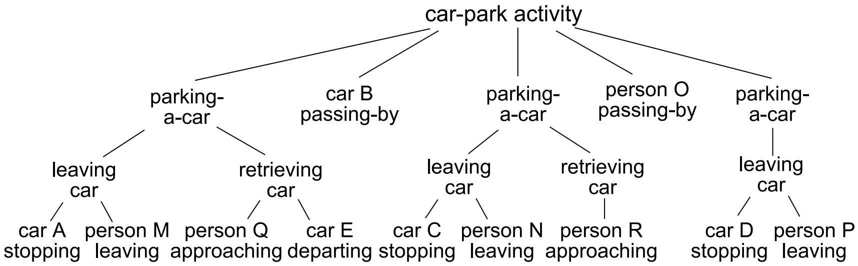


Figure 3.4: The global explanation is expressed as a hierarchy of events.

A set though, by definition, contains distinct objects. An activity can contain multiple instances of the same event. For example, the hand shaking act involves two people performing the same event. A multiset is better suited to represent the collection. 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. The global explanation thus represents the activity as a multiset of compound and primitive events. Each compound event should be further defined as a multiset of simpler events until primitive events are reached.

When recognising an activity, a collection of constraints on consistent events can be defined. For example, a car needs to stop before a person can leave the car. Failing to enforce these constraints results in inconsistent events. These constraints are *intra-activity constraints* as they govern the relationships between the events making up the same compound event. Temporal and spatial intra-activity constraints can be identified in the car parking activity.

Another set of constraints, often ignored in activity recognition, is referred to as the *inter-activity constraints*. While *intra-activity constraints* ensure each recognised event is internally consistent, *inter-activity constraints* ensure the complete set of recognised events is consistent. For example, dropping a person off by a car involves one person detection and one car detection. A person can be dropped off by only one car, while the same car can drop off multiple people. Allowing two cars to drop off the same person results in an inconsistent set of events, regardless of how close the person was to both cars. On the other hand, a solution that allows the car to drop off only one person is over-constrained. Explaining each event independently fails to take inter-activity constraints into consideration, and can result in an inconsistent set of events. This research makes a clear distinction between the two types of constraints due to two reasons. The first is that inter-activity constraints are often ignored in activity recognition, so are worth highlighting. The second is that the two types are defined in different ways as will be shown in Section 3.2.

The framework presented in this chapter attempts to define global explanations, where all detections are explained, maintaining intra- and inter-activity constraints. Section 3.2 proposes a grammatical representation to define consistent sets of events that satisfy the activity's constraints.

3.2 Attribute Multiset Grammars

A general way to formalise the set of globally consistent explanations for a given activity is not yet available, particularly in the formalisation of constraints within a structural representation. In this section, a grammar formalism is proposed for this task. The grammar is defined so the language it describes corresponds to the set of all global explanations.

Attribute Grammars as first introduced by Knuth [87] [1], also referred to as Feature-

[1] An inspiring reflective narrative about the historical origins of attribute grammars was written by Knuth [88]

Based Grammars [17] and Attribute-Value Grammars [3], add attributes to the terminal and nonterminal symbols of a grammar. These attributes can be used in three ways. The first is to propagate information towards the root of the parse tree; ancestors derive their attribute values from those of their descendants. The second is to propagate attribute values 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 string grammar rewrites a symbol into a sequence of symbols, multiset grammars rewrite a symbol into a multiset. Attribute Multiset Grammars (AMG) were introduced in [55] for representing the constituents and layout of a picture. They have also been referred to as Constraint Multiset Grammars [102]. Visual languages were later defined as graph grammars because connectors between neighbouring shapes require a formal definition of edges. A review of grammars for visual languages can be found in [10].

Conventional approaches to activity recognition expect an inherent order of events to define a compound event. Context-Free (string) grammars were thus used for the definition. When a compound event can be carried out in different orders, each order has to be defined separately. This research adopts the viewpoint that the compound event is made up of an (unordered) set of events. Temporal (i.e. causal) relationships between some of these events could be defined, but an ordering is not enforced when it does not exist naturally. The AMG formalism thus satisfies the requirements introduced in Section 3.1 for formally defining global explanations. It rewrites the activity as a multiset of events, which can be further defined as multisets of other events. Note that two event instances of the same type are considered identical, which triggered the usage of the multiset grammar. Moreover, attributes allow defining and constraining intra- and inter-activity relationships. The terminology used in the rest of the chapter follows the one introduced by Knuth in [87]. Here, 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 (S ∈ N), 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 descendants, and $A_1(X)$ is the set of *inherited* attributes which are calculated based on the attributes of the ancestors [87].

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 $m \in M$ 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.

Analogous to the types of attributes, an attribute rule $m \in M$ is synthetic (M_0) if it assigns a value to a synthetic attribute, and is an inherited attribute rule (M_1) otherwise. Similarly, there are two types of attribute constraints C; synthetic constraints (C_0) which specify allowed values for synthetic attributes and inherited constraints (C_1) which limit the values assigned to inherited attributes.

AMG can thus be used to define activities 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 features extracted for each primitive event or detection. These can be used to calculate attributes of compound events. For example, the temporal extent for each primitive event is retrieved directly by the detector. The temporal extent of a compound event is the union of all its primitive events' durations.
- Inherited attributes (A_1) are explanation-related. For example, the person who is part of a car-leaving event is a driver. Such attributes are not calculated from the input, but are assigned based on the explanation, and differ between explanations
- Synthetic rules (r) define the structure of the activity's hierarchy. The rule: A $\rightarrow a, b$ means the compound event A is made up of the primitive events a and b.
- Synthetic constraints (C_0) define intra-activity constraints. They limit the temporal and spatial relationships between the grouped events, as the time and location of the event are synthetic attributes.
- Inherited constraints (C_1) define inter-activity constraints. Sharing an event between two compound events can be forbidden by maintaining a count for the number of times each event is shared. Such a count is decided by the chosen explanation and varies between explanations. It thus is an inherited attribute.

To illustrate, consider the AMG grammar G_a defined next

Nonterimanls (N):	S	start symbol
	A	compound event 1
	B	compound event 2
Terminals (T):	a	primitive event 1
	b	primitive event 2
	c	primitive event 3

Attributes (A):

attribute name	**type**	**domain**	**defined for**
time	$\in A_0$	$\mathbb{Z}$	{a, b, c, A, B}
count	$\in A_1$	$\mathbb{Z}$	{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.time	=	a.time+B.time	a.time	$<$	B.time
				B.count	=	1	B.count	$\neq$	1
p_3	B	$\rightarrow$	b,c	B.time	=	c.time	b.time	$<$	c.time
				b.count	=	B.count	b.count	$\neq$	1

The sample AMG, G_a, defines three nonterminal symbols of which 'S' is the starting symbol of the grammar. It defines three terminals that can be detected directly from the input. One synthetic and one inherited attribute is defined along with three production rules. The first production rule p_1 rewrites the start symbol into the possible event types. The multiset $\{A^\star, B^\star, a^\star, c^\star\}$ indicates that the activity is a multiset of events of these four types. The star indicates the presence of zero or more events of each type in the multiset. A primitive event of type a can then be part of a compound event A, or not. Primitive events of type b on the other hand, cannot occur on their own.

The second production rule p_2 specifies the hierarchy of the compound event 'A'. Two attribute rules and two attribute constraints are defined for p_2. The first attribute rule $A.time = a.time + B.time$ is synthetic as it calculates the value of the attribute 'time' from some values of the descendants' attributes. The second attribute rule $B.count = 1$ is inherited as it assigns a value of 1 to the attribute 'count' of the descendent symbol B. The two attribute constraints are synthetic and inherited respectively. Notice that the event 'B' can participate in only one event of type A by setting the count to 1 when the rule is applied and constraining it to non-1 values by the inherited constraint.

Figure 3.5 shows the dependency graph corresponding to the attribute rules in G_a. A dependency graph [87] graphically represents the dependencies amongst attributes. In the graph, each symbol is surrounded by its attributes. Synthetic attributes are listed to the right while inherited attributes are to the left of the symbol. A dotted line shows the derivations of the syntactic rules, while an arrow denotes the attribute dependencies. The dependency $x \rightarrow y$ means attribute y is dependent on x. The value of attribute y cannot be known before x is assigned a value. Arrows pointing upwards indicate synthetic attribute rules, while downward arrows denote inherited attribute rules.

Though any set of attributes and attribute rules can give rise to an AMG, all the grammars used in this thesis are Ordered Attribute Grammars (OAG) [80]. An ordered grammar assumes a partial order over the attributes is defined. All the attributes can then be evaluated in this order in a finite number of passes. Kastens provides an algorithm to check whether an AMG is an ordered grammar [80]. This is not necessary if the attribute dependency graph clearly shows the non-circulatory nature of the grammars. This thesis assumes the AMG for activity recognition is an OAG, as other grammars cannot be evaluated into parse trees with values for all the attributes. This is though not a restriction to defining activities, because attributes are features and interpretations that should have values in all cases. When activities are represented by an AMG, and the attribute dependency graph is not obviously non-circular, the algorithm in [80] can check that the grammar is ordered.

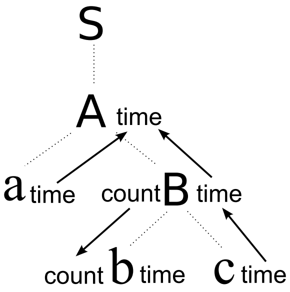


Figure 3.5: Attributes dependency graph showing synthetic and inherited attributes.

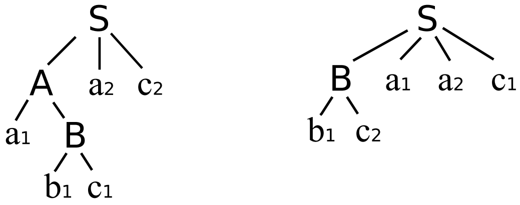


Figure 3.6: Two parse trees given a multiset of detections and AMG G_a.

For each input video, detectors are used to retrieve a multiset 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_a, is the set of all possible explanations for the input video. For the grammar G_a, suppose the detectors generated the following multiset D = $\{a_1$ (time=1), a_2 (time=2), b_1 (time=2), c_1 (time=3), c_2 (time=4)$\}$ - subscripts distinguish different instances of the same terminal. Values for the only synthetic attribute *time* are assigned by the detector for each detected instance of the terminal symbols. Figure 3.6 shows two possible derivations (i.e. parse trees). Starting from the start symbol ‘S’, the set of all distinct explanations equals the set of all possible parse trees. Recall that the order of branches in the tree is irrelevant.

Attribute constraints ensure that only consistent events are generated. For example,

Derivation	Check constraints	Apply attribute rules
B → b_1,c_1	$b_1.\mathit{time} < c_1.\mathit{time}$	B.time = c_1.time = 3 b_1.count = 1
A → a_3, B	$a_3.\mathit{time} >$ B.time	

Figure 3.7: An example of a violation constraint. The syntactic rule cannot be parsed.

given a new detection a_3 (time = 5), Figure 3.7 shows that the second production rule cannot be applied as the constraint was violated.

This section showed how Attribute Multiset Grammars (AMG) can represent the domain's knowledge for global analysis of activities. Syntactic rules of the grammar encode the hierarchical structure of the activity. Attribute rules and constraints enforce the natural constraints.

3.3 A Bayesian approach to finding the best parse tree

Section 3.2 detailed how global explanations for a multiset of detections arise as parses according to a given grammar. Given the same detections, the set of different possible parse trees corresponds to the set of all global explanations. To find the best parse tree given a multiset of detections in a Bayesian approach, all detections need to be assessed along with prior probabilities that would favour some parse trees over others. The probability distribution over the space of possible explanations is modelled using a Bayesian network.

In this Bayesian network, a Boolean node is added for each compound or primitive event in all the possible parse trees. Each global explanation is thus a labelling of the BN, so only the nodes corresponding to the set of events in this parse tree are labelled true. Finding the best explanation is then finding the Maximum a Posteriori (MAP) labelling of the Bayesian network. The joint probability of all the nodes in the BN is factorised. Conditional links are formed between events and their associated evidences, between compound events and their constituent events, and between related events when enforcing consistency in the parse tree. This is explained next in detail.

Bayesian Networks (BNs) are directed graphical models that convey the independence assumptions in a joint probability distribution [16]. In a BN, nodes represent random variables (RVs), while directed edges represent the dependency between these variables. A directed edge from node ‘a’ to node ‘b’ symbolises that the value of ‘b’ depends on the value assigned to ‘a’. This is often informally referred to as ‘a’ being the parent of node ‘b’. In Bayesian networks, the value assigned to a random variable is only dependent on the values of its parents, children and co-parents (referred to as the Markovian blanket of the RV). Three types of random variables are used in this section to build the BN:

- Observed Random Variables, or observables: denoted by shaded nodes, and represent discrete or continuous values that can be directly measured from input data.
- Hidden Random Variables, or latent variables: denoted by non-shaded nodes, and represent the explanation inferred from the node's Markovian blanket.
- Deterministic Random Variables: denoted by double-circled nodes, and represent variables that functionally depend on the values of its parents. A Boolean function decides the value of the deterministic random variable based on the values assigned to its parents.

A simple one-rule example is presented first. Given a pair of primitive events x and y, and one syntactic rule ($Z \rightarrow x,y$), Figure 3.8 (left) shows a Bayesian network with three evidences; o_x, o_y and o_z representing the set of synthetic attribute values for each symbol. The evidence o_z is the calculated synthetic attribute values associated with the syntactic rule. Three hidden random variables, (x,y,Z), explain the two primitive events and one compound event. The joint probability is factorised so the compound event is dependent on its constituent events. It is important to clarify that **the descendants in the parse tree are the parents in the Bayesian network**. Each hidden random variable is Boolean (t/f), where 't' represents the occurrence of the event, while 'f' indicates the event is not recognised. For each synthetic attribute, a conditional probability density function (cpdf) needs to be defined for each labelling. In this example, $p(o_x|x=t)$ and $p(o_x|x=f)$ are required, and similarly for the other two observed random variables. These cpdfs can be learned from labeled data as will be shown in Chapters 5 and 6.

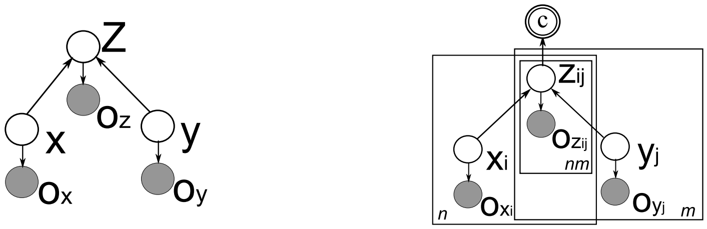


Figure 3.8: Directed graph for the production rule $Z \rightarrow x,y$ given two detections (left) and a plate representation for multiple events (right).

For a multiset of detections with n detections of type x and m detections of type y, then Figure 3.8 (right) shows a plate representation linking each x event to all possible y

events according to the production rule. Though each compound event is dependent on its constituent events, inter-activity constraints should also be governed. A deterministic random variable is added to link inter-dependent events. In the plate representation, all Z compound events are assumed inter-dependent and thus are linked to one deterministic random variable c. The inter-dependent nodes are those nodes whose production rules include inherited attribute constraints governing the same inherited attribute. This will be further explained later in this section. Figure 3.9 is an unrolled example for $n = 3$ and $m = 2$. The different kinds of nodes in the Bayesian network are labeled on the left hand side. Each pair of x and y RVs parents one compound event node Z. Figure 3.10 shows a parse tree and the corresponding labeled Bayesian network.

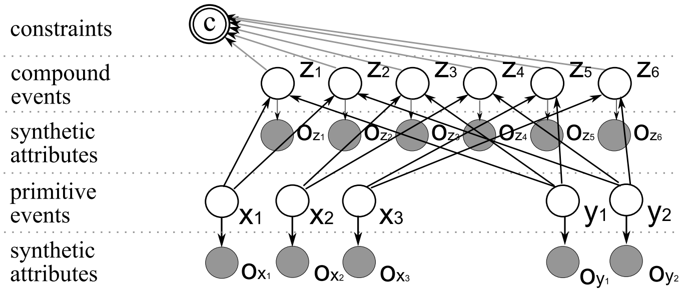

constraints
compound events
synthetic attributes
primitive events
synthetic attributes
c
z_1 z_2 z_3 z_4 z_5 z_6
o_{z_1} o_{z_2} o_{z_3} o_{z_4} o_{z_5} o_{z_6}
x_1 x_2 x_3 y_1 y_2
o_{x_1} o_{x_2} o_{x_3} o_{y_1} o_{y_2}

Figure 3.9: An unrolled Bayesian network for multiple events.

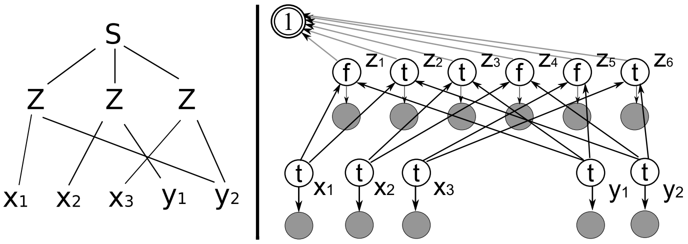

S
Z Z Z
x_1 x_2 x_3 y_1 y_2
1
z_1 z_2 z_3 z_4 z_5 z_6
f t t f f t
t t t t t

Figure 3.10: A sample explanation (left) and its corresponding labelling of the BN (right).

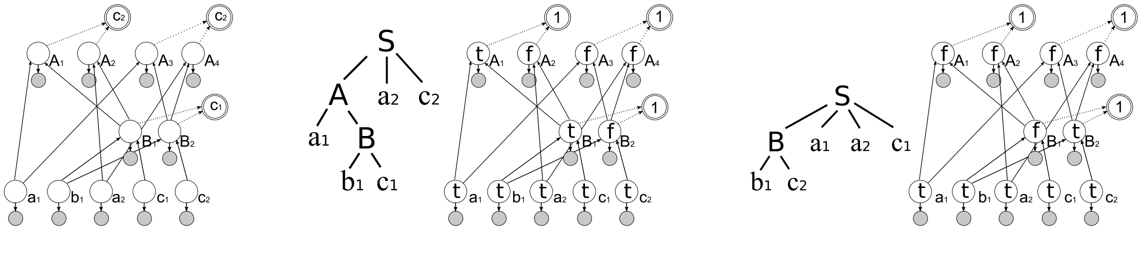


Figure 3.11: The Bayesian network for the grammar G_a along with two labellings that reflect the parse trees in Figure 3.6. A node is labeled true if it appears in the parse tree. The deterministic function evaluates to 1 for labellings that satisfy the inherited constraints.

In Section 3.2, an AMG was introduced as an example along with two parse trees for a multiset of detections. Figure 3.11 shows the Bayesian network for the specified detection multiset along with two labellings that reflect the parse trees in Figure 3.6. Notice that each possible nonterminal in the parse trees is represented by a hidden random variable (RV), and is labeled true if the nonterminal appears in the explanation's parse tree, and false otherwise.

The method for building this BN might not seem obvious. Algorithm 3.1 details the steps for building a BN out of a set of detections D and an AMG. Figure 3.12 traces the algorithm to build the sample BN in Figure 3.11. The set of production rules is ordered (p_3, p_2, p_1), then a hidden-and-observed RV pair is created for all the detections in D. The observed RV holds the value(s) of the synthetic attribute(s) for each detection.

Lines 7-23 in the algorithm build the BN's structure. For the first production rule (p_3 : B $\rightarrow$ b, c), the possible combinations (line 10) are

$$\text{comb} = \{\ (\text{b}_1, \text{c}_1), (\text{b}_1, \text{c}_2)\ \}$$

For each of these two tuples, the synthetic attribute constraint $b.time < c.time$ is checked (line 13). As the constraint is satisfied for both tuples, two hidden RVs of type B are created $\{\text{B}_1, \text{B}_2\}$. The synthetic attributes are calculated for each (B_1.time = 3, B_2.time = 4), and represented by a related observed RV. The dependency links are established between the compound event and its constituent events. Similarly, the second level of the BN is built for the rule $p_2 : A \rightarrow a, B$.

To accommodate for *direct recursion* in grammars, the loop (lines 11-23) checks if new tuples (lines 20-23) have been added. Direct recursion occurs when the multiset at the right hand side of the production rule contains an instance of the nonterminal at the left hand side, for example A → a, A, b. The algorithm cannot deal with indirect recursion, like

A → a, B
B → b, A
A → c

These cases can be checked while establishing the order of the production rules (line 2).

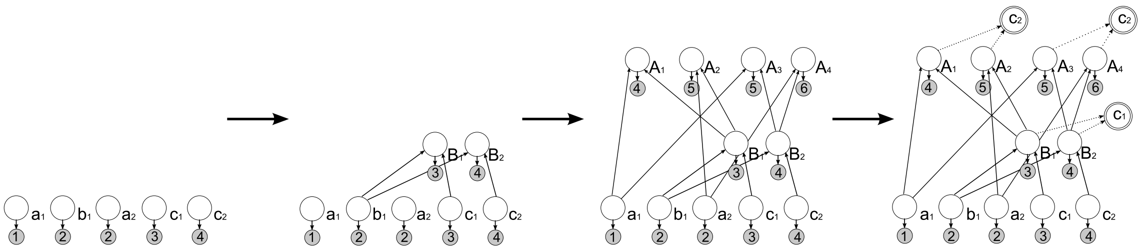


Figure 3.12: An example of constructing the BN from the AMG G_a and a set of detections, shown in steps.

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

initialise 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 synthetic attribute values
foreach rule $p \in P$ ($p.r : X_0 \rightarrow X_1, X_2, \ldots, X_n$)
    if $X_0 \neq S$ then
        Let I($X_i$) be the set of nodes in BN of type $X_i$
        $comb$ = I($X_1$) $\times$ I($X_2$) $\times$ ... $\times$ I($X_n$)
        while size of comb $> 0$ do
            foreach tuple $b \in comb$
                if $b$ satisfies synthetic attribute constraints $p.C_0$ 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
                            apply $m$ assigning a synthetic attribute value to $X_0$
                    add a related observed RV to hold synthetic attribute values
                    all nodes in the tuple $b$ parent the created hidden RV
            $comb$ = I($X_1$) $\times$ I($X_2$) $\times$ ... $\times$ I($X_n$) - $comb$
            foreach new tuple $b \in comb$
                if the primitive events of $b$ has redundancies then
                    remove $b$ from $comb$
Let Nodes$_n$ be the set of all hidden RVs associated with nonterminal symbols $N$
while $Nodes_n \neq \phi$ do
    find Nodes$_p$ with inherited constraints limiting the same inherited attribute values
    Nodes$_n$ = Nodes$_n$ - Nodes$_p$
    if size of $Nodes_p > 1$ then
        add deterministic RV $c$ to hold the inherited constraints
        all nodes in $Nodes_p$ parent the deterministic RV $c$
```

Algorithm 3.1: Mapping a multiset of detections D to the Bayesian network (BN) structure that represents the probability distribution over the set of possible parses, given an AMG G.

Grammars with indirect recursion would have ambiguous ordering of the rules. Handling indirect recursion could be done in principle, because the BN is based on a finite set of detections. A possible (yet inefficient) algorithm can loop through all rules until the BN is completely built. Designing an efficient algorithm is left for future work. This is not seen as a limitation to defining activities, because direct recursion is sufficient to define repetitive patterns in the grammar.

Lines 24-30 explain how inter-dependent nodes can be found and linked to deterministic random variables. First, the set of all nonterminal nodes along with the set of inherited attributes each one constrains, is accumulated. For the example BN, this set is $\{\mathrm{B}_1 \rightarrow \{b_1.\mathrm{count}\}, \mathrm{B}_2 \rightarrow \{b_1.\mathrm{count}\}, \mathrm{A}_1 \rightarrow \{\mathrm{B}_1.\mathrm{count}\}, \mathrm{A}_2 \rightarrow \{\mathrm{B}_1.\mathrm{count}\}, \mathrm{A}_3 \rightarrow \{\mathrm{B}_2.\mathrm{count}\}, \mathrm{A}_4 \rightarrow \{\mathrm{B}_2.\mathrm{count}\}\}$. Lines 24-25 iteratively find the sets of interdependent events. In this example, three sets of interdependent events are identified; $\{\mathrm{B}_1, \mathrm{B}_2\}$, $\{\mathrm{A}_1, \mathrm{A}_2\}$, $\{\mathrm{A}_3, \mathrm{A}_4\}$. For each set of inter-dependent events, one deterministic random variable is created. In this case, the deterministic functions check that a maximum of one node in each inter-dependent set is recognised at a time. Symbolically,

$$p(C|e_1, e_2) = \neg(e_1 \wedge e_2) \tag{3.1}$$

where e_1 is assigned true when the event is recognised and the logical expression evaluates to zero for false expressions and 1 for true ones.

3.3.1 Multi-labelled Bayesian networks

All the BN examples presented up till now assume a Boolean labelling which indicates whether an event is recognised. Let's take another AMG G_x, with the following synthetic rules,

S → B*, C*
A → a, b
B → a, b
C → A, c

In this example the same multiset of detections $\{a, b\}$ can be combined into two different event types. For example, a person can either get into a car or leave a car. Given the detections multiset $D_x = \{a_1, b_1, c_1, c_2\}$, then the BN would be presented in Figure 3.13. Two Boolean hidden RVs are created, one for event 'A' and another for event 'B', and are constrained. Alternatively, one multi-labeled hidden RV can be used. The more concise grammar G_y can be introduced.

S → D*, C* D.action = 'B'
D → a, b
C → D, c D.action = 'A'

The hidden RV 'D' has three possible labels, {A, B, f}. The Bayesian network is then represented in Figure 3.14. Algorithm 3.1 can still be used to generate the Bayesian network's structure. The set of possible labels allowed for each hidden RV needs to be specified. A special attribute, called ‘action’, is defined for these 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. The false label is assigned when the event is not recognised. While Boolean labelling is the default option, multi-labelling enables a more concise formulation and decreases the number of constraints as will be shown in the AMG for the *Bicycles* problem (Chapter 5) and the *Enter-Exit* problem (Chapter 6).

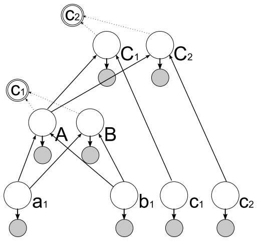


Figure 3.13: Boolean BN for the AMG G_x

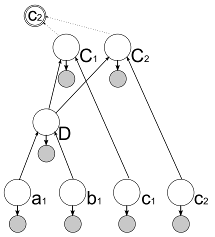


Figure 3.14: Multi-labelled BN for the AMG G_y

After building the topology of the BN, priors and conditional probabilities need to be specified. Priors are defined for primitive events. For each production rule, the conditional probability of the nonterminal at the left hand side given the multiset at the right hand side should be specified. For example, for the derivation $D \rightarrow a, b$, where the set of possible labels are as follows $D.action \in \{A, B, f\}, a.action \in \{t, f\}, b.action \in \{t, f\}$, then $p(D|a,b)$ can be defined by assigning a value to each conditional probability in the following table:

$p(D\|a,b)$	$D=A$	$D=B$	$D=f$
$a=t, b=t$			
$a=t, b=f$			
$a=f, b=t$			
$a=f, b=f$			

Notice that $p(D|a,b)$ should more precisely be written as $p(D.action|a.action, b.action)$ as the possible values of the attribute ‘action’ are the possible labels of the hidden random variable. In the rest of the thesis, for each symbol $X \in N \cup T$, X and X.action are used interchangeably, and X is often used for simplicity.

In our research, these conditional probabilities are estimated by an expert 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 training data and is a computationally hard optimisation problem due to the dependencies between the production rules that arise from the constraints. Abney [3] explained how the conditional probabilities can be correctly estimated from training data, using sampling and selecting features that incorporate the dependencies between the rules.

This section has shown that the search for the best parse tree can be performed by transforming the detections into a single Bayesian network (BN) that models the probability distribution over the space of all possible explanations for those detections. Each global explanation, represented by a parse tree corresponds to a labelling of the BN. Boolean and multi-labelled BNs have been discussed. The best parse tree then corresponds to the Maximum a Posteriori labelling of the BN. This section presented an algorithm that automatically performs this transformation from the AMG to the BN.

3.4 The posterior probability

The BN built in Section 3.3 models the probability distribution over the set of all global explanations given a multiset of detections. To find the best explanation, one needs to infer the Maximum A Posteriori (MAP) labelling $\omega^\star$ of all the hidden random variables, given all observed RVs Y;

$$\omega^\star = \arg\max_{\omega} p(\omega|Y) \tag{3.2}$$

For the simple AMG of one production rule in Figure 3.9, and multiset of detections $\{x_i\}, \{y_j\}$, the posterior is written as

$$p(\omega|Y) = \frac{1}{\mathscr{G}} \prod_i p(o_{x_i}|x_i)p(x_i) \prod_j p(o_{y_j}|y_j)p(y_j) \prod_{ij} p(o_{z_{ij}}|z_{ij})p(z_{ij}|x_i,y_j)p(c|\{z_{ij}\}) \tag{3.3}$$

The posterior can be re-arranged as (Appendix C)

$$p(\omega|Y) = \frac{1}{\mathscr{Z}} \prod_i p(x_i|o_{x_i}) \prod_j p(y_j|o_{y_j}) \prod_{ij} p(z_{ij}|x_i,y_j,o_{z_{ij}})p(c|\{z_{ij}\}) \tag{3.4}$$

where $\mathscr{Z}$ is the normalising factor that need not be evaluated when searching for the maximum. $p(x_i|o_{x_i})$ is the posterior of the label assigned to x_i given the evidence from the synthetic attribute values o_{x_i} and similarly for $p(y_j|o_{y_j})$. The deterministic function $p(c|\{z_{ij}\})$ evaluates the labels of all z linking nodes, and equals 1 if the labels are consistent, and zero otherwise. Accordingly, the posterior for inconsistent labelling evaluates to zero always.

The third factor in Equation 3.4 becomes intractable to compute as the number of detections increases. Fortunately, this can be avoided by computing a proportional quantity instead. This is derived as follows ($p(z_{ij}|x_i,y_j,o_{z_{ij}})$ is abbreviated to $p(z_i|\cdot)$ in the derivation)

$$
\begin{aligned}
\prod_i p(z_i|\cdot) &= \prod_{i:z_i=f} p(z_i=f|\cdot) \prod_{i:z_i=t} p(z_i=t|\cdot) && (3.5)\\
&= \prod_{i:z_i=f} p(z_i=f|\cdot) \prod_{i:z_i=t} p(z_i=t|\cdot) \frac{\prod_{i:z_i=t} p(z_i=f|\cdot)}{\prod_{i:z_i=t} p(z_i=f|\cdot)} && (3.6)\\
&= \prod_i p(z_i=f|\cdot) \prod_{i:z_i=t} \frac{p(z_i=t|\cdot)}{p(z_i=f|\cdot)} && (3.7)\\
&\propto \prod_{i:z_i=t} \frac{p(z_i=t|\cdot)}{p(z_i=f|\cdot)} && (3.8)
\end{aligned}
$$

This derivation specifically enables finding a quantity, proportional to the original posterior, that is independent of all false-labelled nodes (i.e. unrecognised events). Accordingly, evaluating the posterior of a single parse tree should only take into consideration the events recognised within the parse tree, and should not be concerned with the remaining unrecognised events. This uses the fact that labelling all the nodes as false is a fixed quantity. For nodes labelled true, the ratio of labelling a node as true to labelling it as false is sufficient to compare the posterior across various labellings of the Bayesian network.

Thus, the posterior $p(\omega|Y)$ is rewritten to be

$$
p(\omega|Y) = \frac{1}{\mathscr{Q}} \prod_i p(x_i|o_{x_i}) \prod_j p(y_j|o_{y_j}) \prod_{ij:Z_{ij}=t} \frac{p(z_{ij}=t|x_i,y_j,o_{z_{ij}})}{p(z_{ij}=f|x_i,y_j,o_{z_{ij}})} \prod_{ij} p(c|\{z_{ij}\}) \qquad (3.9)
$$

Notice the difference between the normalising factor $\mathscr{X}$ in Equation 3.4 and the normalizing factor $\mathscr{Q}$ in Equation 3.9. This is because the term in Equation 3.8 is only proportional, but not equal, to the term in 3.5. In Figure 3.15, the unrecognized events in the BN are drawn in light grey to show the compound events that are labeled true and their accompanying observed random variables. Only these nodes are required to calculate the posterior for this explanation. This shows that evaluating the posterior for a parse tree only consid-

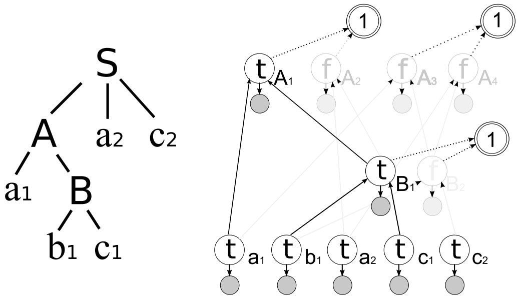


Figure 3.15: The highlighted nodes are the only nodes included in calculating the posterior in Equation 3.10 for the labeled explanation.

ers the non-terminal symbols in this tree. Accordingly, the posterior for the parse tree in Figure 3.15 equals

$$p(\omega|Y) = \frac{1}{\mathscr{Q}} p(a_1|O_{a_1}) p(a_2|O_{a_2}) p(b_1|O_{b_1}) p(c_1|O_{c_1}) p(c_2|O_{c_2}) \frac{p(B_1 = t|O_{B_1}, b_1, c_1)}{p(B_1 = f|O_{B_1}, b_1, c_1)} \frac{p(A_1 = t|O_{A_1}, B_1, a_1)}{p(A_1 = f|O_{A_1}, B_1, a_1)} \quad (3.10)$$

This is extensible to multi-labeled BN (Section 3.3.1). The posterior would still be independent of all false labelling. Recall that in both Boolean and Multi-labelled BN, ‘false’ is a possible label for all hidden RVs. Using the posterior from Equation 3.9 decreases the number of likelihoods calculated to evaluate each global explanation.

Chapter 4 explains that exact inference is intractable in most cases, and presents heuristic search techniques to find the MAP labelling of the BN.

3.5 Synthetic attributes

In the previous sections, the synthetic attributes for each symbol were already known and encoded in the AMG. The choice of the synthetic attributes was not discussed, and is the topic of this section. These synthetic attributes are features selected for each detection. The features should be selected to help distinguish the different events.

Some synthetic attributes of the primitive event are used to calculate attribute values for the compound events. For example, in the rule $Z \rightarrow x, y$, the compound event Z can be measured by the spatial proximity between x and y. Accordingly, the locations of x and y have to be measured, and o_z is the distance between these locations, calculated by a function defined in the grammar. Thus, some synthetic attributes distinguish the occurrence of primitive events, and others are used to calculate the values of synthetic attributes for more complex events.

Selecting which feature best distinguishes whether an event occurred or not can be performed manually or automatically. Learning varies between supervised, semi-supervised and unsupervised methods. For the cases studies in Chapters 5 and 6, features that could distinguish the event types are manually selected. This avoids features that are specific to the training data, because they are based on the expert’s knowledge. The expert selects these features while defining the AMG. The framework though is general and is independent of the choice of the features. One can replace these features with different or multiple features, and follow the same recognition procedure.

If multiple synthetic attributes are chosen to distinguish whether an event occurred (e.g. location and time), independence is assumed given the features are retrieved independently from the data. The cpdf is then the product of the likelihoods of those features:

$p(o_r|r) = \prod_k p(r.a_k|r)$. Given training data for different labels of r, a conditional pdf can be learnt over each attribute $r.a_k$ and each label x: $p(r.a_k|r = x)$.

3.6 Conclusion

This chapter explained how AMG can be used to present the domain's activities as hierarchies of compound and primitive events, along with intra- and inter-activity constraints. In an AMG, terminals represent primitive events that directly correspond to detections, and nonterminals represent compound events. Each symbol (i.e. terminal or nonterminal) has synthetic and inherited attributes. Each production rule in the grammar rewrites a nonterminal into a multiset of symbols. A production rule is accompanied by attribute rules that traverse values up and down the parse tree, and attribute constraints that ensure the natural constraints are satisfied.

Parsing a multiset of detections by the AMG generates a global explanation that covers all the detections, and satisfies all constraints. The set of all possible parse trees represents the set of global explanations for the detections. The chapter presents an algorithm to transform the multiset of detections, given the AMG, into a Bayesian network structure. The set of labellings of the BN corresponds to the set of all parse trees. After setting the priors and the conditional probabilities for the BN, the MAP solution represents the best explanation for the detections. The next chapter explains tractable techniques to search the BN for the MAP explanation.

Chapter 4

Searching for the Best Explanation by Optimising a Bayesian Network

Chapter 3 shows how to build a Bayesian Network (BN), given a set of detections, that models the probability distribution over the space of global explanations. The complete set of labellings of the Bayesian network corresponds to the set of all explanations. The Maximum a Posteriori (MAP) explanation is the best explanation according to the probability distribution. This chapter presents an exhaustive method for finding the MAP solution that is tractable in certain cases. It also presents three heuristic methods that are tractable in general.

The three heuristic search techniques are: greedy search (Section 4.2), Multiple Hypotheses Tree (MHT) (Section 4.3) and sampling the distribution using Reversible Jump Markov Chain Monte Carlo (RJMCMC) with Simulated Annealing (SA) (Section 4.4). The RJMCMC section introduces general reversible moves that can traverse the space of binary event hierarchies. Finding the solution using Integer Programming (IP) is the proposed exhaustive search method, and is explained in Section 4.5.

This chapter motivates the usage of these techniques, that have previously been proposed in the literature for similar problems. It also explains each technique and details how it can be applied to search the BN of global explanations. The search techniques introduced in this chapter are compared experimentally in Chapters 5 and 6 for the two case studies.

4.1 The complexity of the search space

The size of the search space can be estimated from the number of nodes in the BN and the different labellings of each node. First a detections multiset D = $\{$a$_1$, a$_2$, ..., a$_{n_a}$, b$_1$, b$_2$, ... , b$_{n_b}\}$ is acquired using the detectors, where each a_i is a different detection of type a and similarly for b_j. For an AMG G, and the detections multiset D, the number of hidden RV nodes in the BN cannot be calculated in advance, as synthetic constraints govern the ways nodes are combined. An upper bound on the number of nodes can though be calculated. Assuming all rules rewrite a nonterminal into two symbols (i.e. binary parse trees), h is the maximum depth of the parse tree, and n is the maximum number of detections of the same type in D, then the number of nodes is of order $O(n^h)$. The number of constrained labellings representing explanations cannot be calculated either, as it depends on the inter-activity constraints. For a Boolean BN, the upper bound on the number of explanations is $O(2^{n^h})$. This is a multi-dimensional assignment problem, which is an NP-hard combinatorial optimisation problem [114].

Production Rules (P):

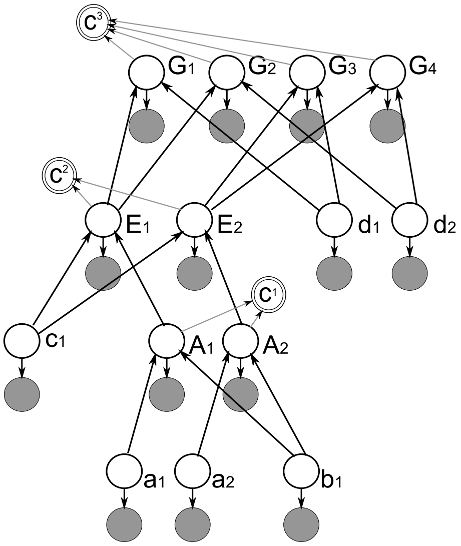
rule	Syntactic Rule (r)			Attribute Rules (M)			Attribute Constraints (C)
p$_1$	S	$\rightarrow$	G*, E*, A*, a*, b*, c*, d*				
p$_2$	A	$\rightarrow$	a, b	A.O_A b.count	= =	f$_A$ (a.O_a, b.O_b) 1	b.count $<$ 1
p$_3$	E	$\rightarrow$	c, A	E.O_E c.count	= =	f$_E$ (c.O_c, A.O_A) 1	c.count $<$ 1
p$_4$	G	$\rightarrow$	E, d	G.O_G d.count	= =	f$_G$ (E.O_E, d.O_d) 1	d.count $<$ 1

Figure 4.1: The production rules of a sample AMG.

To explain the different search techniques, the production rules of a sample grammar are specified in Figure 4.1. Given the following detections $\{a_1, a_2, b_1, c_1, d_1, d_2\}$, Figure 4.2 presents the Boolean BN. This sample BN will be searched using the different techniques. Recall that the search is for the complete labelling of the Bayesian network $\hat{\omega}$ that maximises the posterior probability, given the observations Y. Figure 4.3 shows the exponential relationship between the number of primitive events and the number of hidden RVs for this example.

4.2 Greedy search

A simple technique to find a good global explanation given the Bayesian network is to work from the bottom layer up, incrementally assigning labels to the hidden random vari-



Conditional Probability		Priors	
$p(A=t\|a=t,b=t)$	0.7	$p(a=t)$	1.0
$p(E=t\|A=t,c=t)$	0.8	$p(b=t)$	1.0
$p(G=t\|E=t,d=t)$	0.5	$p(c=t)$	0.9
		$p(d=t)$	0.8

Observations cpdf			
$p(O_{a_1}\|a_1=t)$	0.6	$p(O_{a_1}\|a_1=f)$	0.7
$p(O_{a_2}\|a_2=t)$	0.4	$p(O_{a_2}\|a_2=f)$	0.1
$p(O_{b_1}\|b_1=t)$	0.8	$p(O_{b_1}\|b_1=f)$	0.4
$p(O_{c_1}\|c_1=t)$	0.7	$p(O_{c_1}\|c_1=f)$	0.2
$p(O_{d_1}\|d_1=t)$	0.1	$p(O_{d_1}\|d_1=f)$	0.8
$p(O_{d_2}\|d_2=t)$	0.9	$p(O_{d_2}\|d_2=f)$	0.3
$p(O_{A_1}\|A_1=t)$	0.6	$p(O_{A_1}\|A_1=f)$	0.6
$p(O_{A_2}\|A_2=t)$	0.4	$p(O_{A_2}\|A_2=f)$	0.9
$p(O_{E_1}\|E_1=t)$	0.8	$p(O_{E_1}\|E_1=f)$	0.5
$p(O_{E_2}\|E_2=t)$	0.9	$p(O_{E_2}\|E_2=f)$	0.2
$p(O_{G_1}\|G_1=t)$	0.1	$p(O_{G_1}\|G_1=f)$	0.7
$p(O_{G_2}\|G_2=t)$	0.2	$p(O_{G_2}\|G_2=f)$	0.8
$p(O_{G_3}\|G_3=t)$	0.4	$p(O_{G_3}\|G_3=f)$	0.9
$p(O_{G_4}\|G_4=t)$	0.8	$p(O_{G_4}\|G_4=f)$	0.02

Figure 4.2: A Boolean BN along with a chosen set of priors, conditional probabilities, and the observations likelihoods.

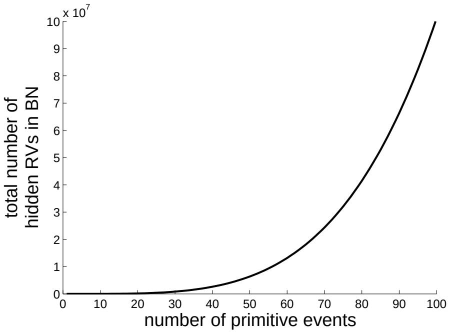


Figure 4.3: The number of nodes in the BN increases exponentially with the number of primitive events.

ables, and checking constraints at each stage. Algorithm 4.1 details how the greedy search is performed for a hierarchical Bayesian network. First, for each primitive event x, the posterior ratio l_x is evaluated,

$$l_x = \frac{p(o_x|x=t)p(x=t)}{p(o_x|x=f)p(x=f)} \tag{4.1}$$

The node is labeled true if $l_x \geq 1$, and false otherwise. This is shown for the sample BN in the first step of Figure 4.4.

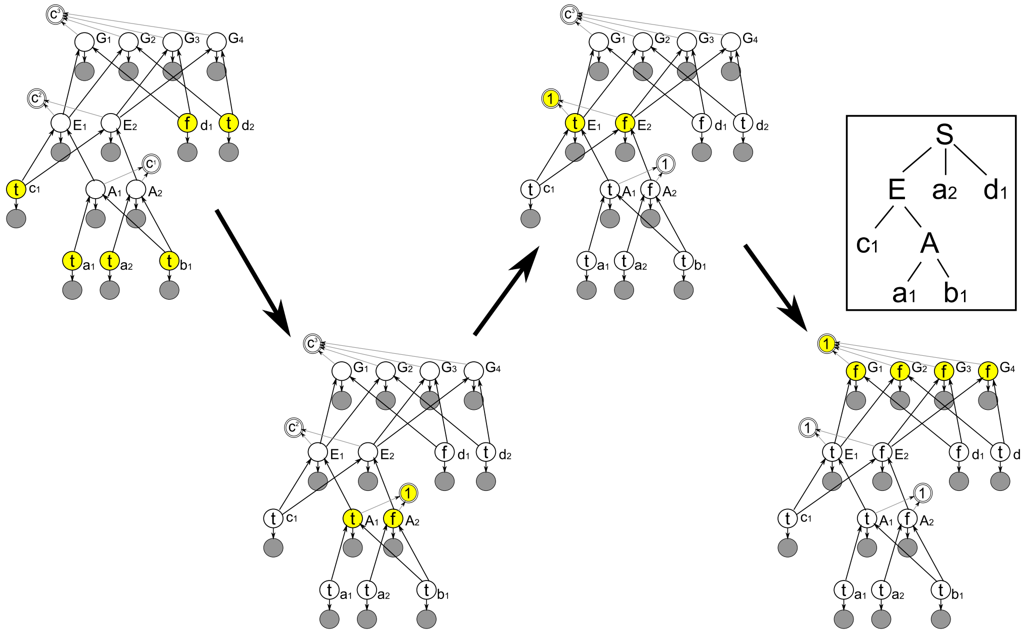


Figure 4.4: Searching the BN in Figure 4.2 using greedy search. Yellow shading of hidden RVs is used to highlight the set of nodes labeled at each step. The resulting parse tree is shown at the end, and corresponds to the fully labeled BN.

```
input  : Bayesian Network BN
output : $\omega_{greedy}$: labelling of the BN
while more nodes to be labeled do
    Let $\{X_i\}$ be the sequence of unlabeled nodes with all parents already labeled (or
    without any parents), in descending order of the ratio $l_{X_i} = \frac{p(X_i=t|o_{X_i},pa_{X_i})}{p(X_i=f|o_{X_i},pa_{X_i})}$, where $pa_{X_i}$
    are the parents of $X_i$
    while more nodes in $\{X_i\}$ are to be labeled do
        Let $X_u$ be the first unlabeled node in $\{X_i\}$
        if $l_{X_u} \geq 1$ then
            label $X_u$ in $\omega_{greedy}$ as $t$
            if $X_u$ is constrained then
                propagate labelling according to the constraint in $\omega_{greedy}$
        else
            label all remaining unlabeled nodes in $\{X_i\}$ in $\omega_{greedy}$ as $f$
```

Algorithm 4.1: Greedy search for labelling a BN

Next, the hierarchy $A \rightarrow a, b$ is assessed. Two nodes A_1 and A_2 are considered.

$$l_{A_1} = \frac{p(A_1 = t|o_{A_1}, a_1, b_1)}{p(A_1 = f|o_{A_1}, a_1, b_1)} \tag{4.2}$$

If $l_{A_1} \geq 1$ then A_1 is labeled true, and similarly for A_2. Yet, if $l_{A_1} \geq 1$ and $l_{A_2} \geq 1$, only the one with the higher ratio is labeled true to satisfy the constraint $c1$ [1]. The evaluation continues up the hierarchy until all nodes are labeled. Figure 4.4 shows how the greedy search can be performed for the sample BN. This is though not necessarily the MAP solution of the BN. This is because each node is evaluated given the pre-labeled parents and those cannot be changed. The greedy search is used as a baseline to compare the results found by the other search techniques.

4.3 Multiple hypotheses tree

The Multiple Hypotheses Tree (MHT) algorithm, first used by Reid for multi-target radar tracking [116], propagates a tree of multiple hypotheses (explanations). It assumes the

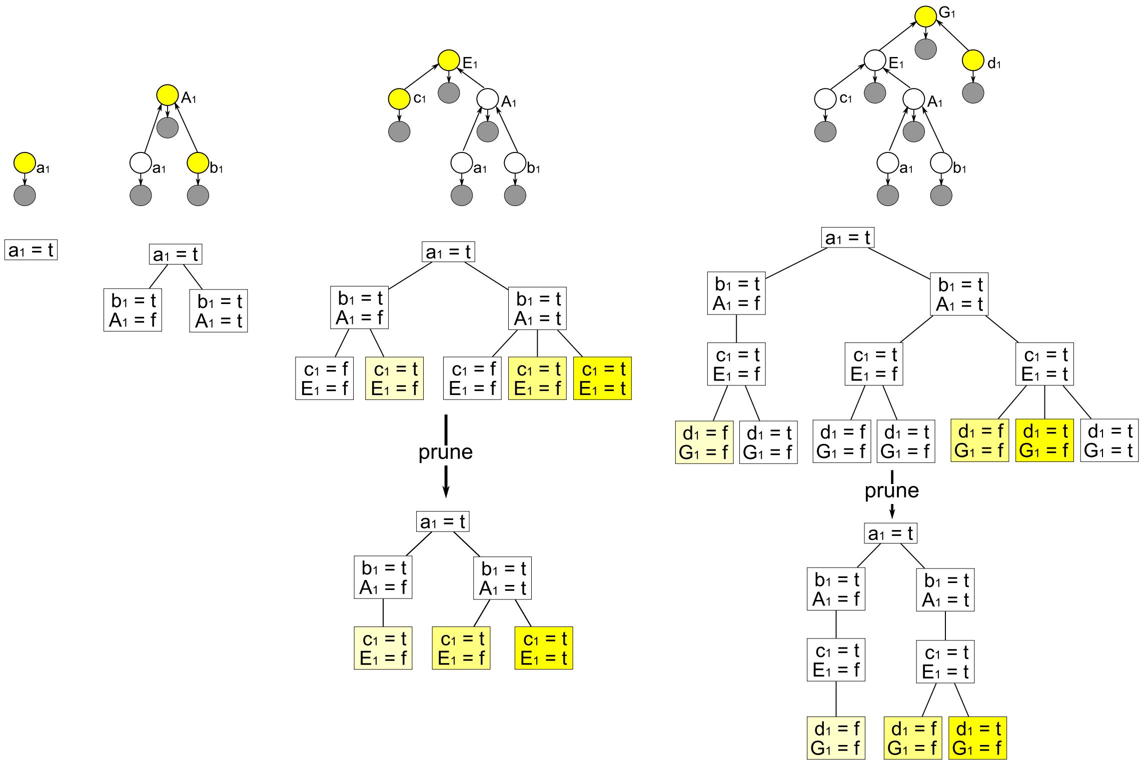


Figure 4.5: MHT considers one detection at a time. The BN (for the detections up to that level) is shown at the top with yellow shading for the detection and all related hidden RVs to be labeled at that level. All feasible labellings are added to the current tree branches. Feasible labellings differ between branches depending on the already labeled nodes at each branch. If the number of branches exceeds k ($k = 3$ in this example), the tree is pruned. Shaded nodes in the tree represent the leaves of the highest k posterior branches with the darkest representing the highest posterior up to that level.

[1] For the constraint that allows only one of the inter-dependent events to be recognised, line 8 in the algorithm labels all conflicting nodes as f. This is the most common constraint in the AMGs presented in this thesis.

detections have an ordering (usually temporal) and starts from the first detection working through to the last. Each level in the tree is thus expanded into nodes representing the different hypotheses explaining the detection in hand. Each path, from root to leaf, in the 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 tree as it propagates into the future. Yet it would be impractical to maintain the complete tree, due to the number of possible hypotheses for all but the simplest cases. The tree is pruned at each step to keep the search tractable by retaining only the best k hypotheses. This is a beam search [123]. The number of retained branches, k, is selected based on a trade-off between number of calculations and accuracy. If $k = 1$, this search becomes a best-first search [123].

Figures 4.5 and 4.6 show how the sample BN (Section 4.1) can be searched using MHT. The search is split into two figures for clarity. k was set to 3, and the following ordering of detections $\{a_1, b_1, c_1, d_1, a_2, d_2\}$ was assumed. The resulting explanation depends on the ordering and might differ between orderings. At each step, a detection is considered along with all 'related' event nodes. The related event nodes are the ancestors

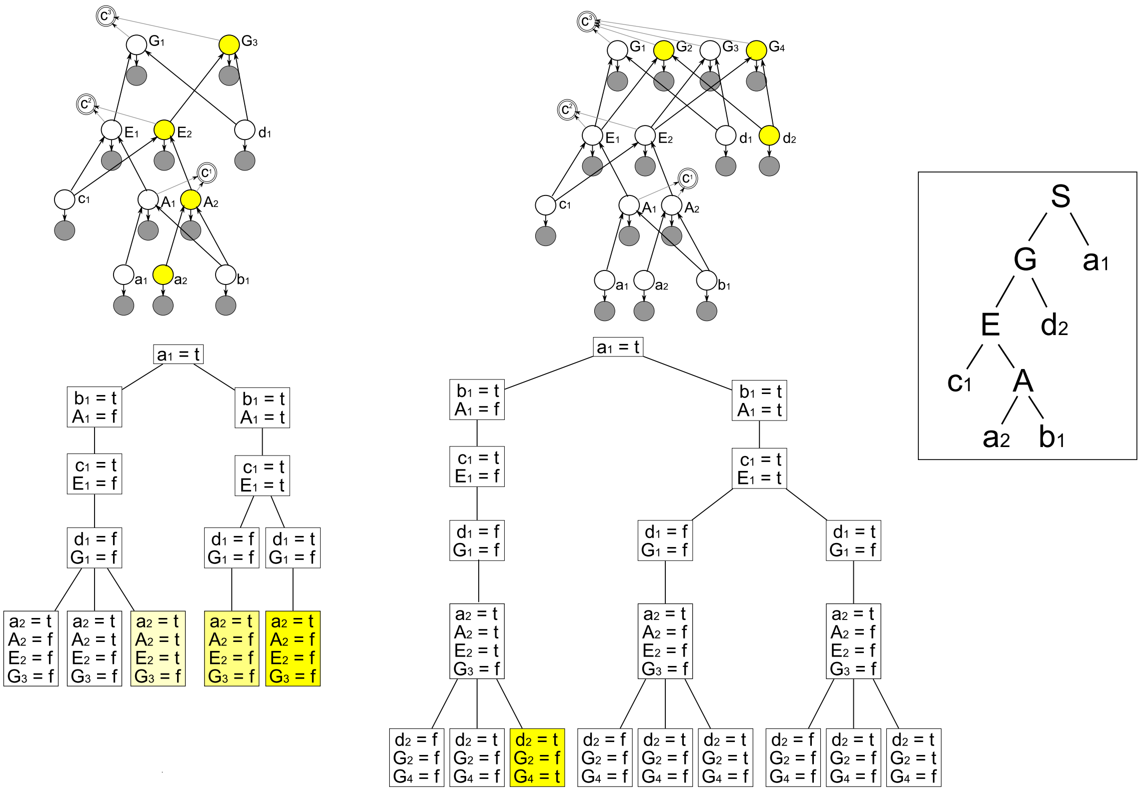


Figure 4.6: MHT search is continued from Figure 4.5. The BNs and MHTs are shown. The parse tree with the maximum retained posterior is shown in a box on the right. Notice that the branch with the maximum posterior changes as the last observation d_2 is added.

of the considered node which have all their other children already labeled. All consistent labels of these nodes are evaluated at this level of the MHT. Considering consistent labellings only increases the speed, as all inconsistent labellings evaluate to a posterior of zero. It should be noted that the consistent labels differ between tree branches depending on the previously labeled nodes in each branch.

After all consistent labels are added to each branch in the MHT, the tree is pruned to retain k branches only. This is accomplished by evaluating each branch, and keeping the k branches with the highest k posteriors. As Figure 4.6 shows, the branch with the highest posterior might change when more evidence is considered. If k was small, and that branch was not retained, the MAP solution cannot be found. It is though not possible to estimate the optimal k in advance, as it depends on the ambiguity in the detections. Algorithm 4.2 shows a pseudo-code for searching a BN using MHT

```
input  : Bayesian Network BN, ordering of detections D, number of branches $k$
output : $\omega_{MHT}$: labelling of the BN

initialise tree $t$ with one empty branch
foreach primitive event $d \in D$
    Let $\{X_i\}$ be the list of nodes related to $d$ of size $m$
    Let $L_{X_i}$ be the set of possible labels of node $X_i$
    $L^m = L_{X_1} \times L_{X_2} \times \ldots \times L_{X_m}$
    foreach branch $b \in$ tree branches
        foreach labelling $l \in L^m$
            if $l$ is consistent with explanation $b$ then
                add node $l$ to branch $b$
        if no labelling is consistent then
            remove branch $b$
    prune tree (i.e. keep $k$-best branches)
$\omega_{MHT}$ = labelling of branch with maximum posterior
```

Algorithm 4.2: Multiple Hypotheses Tree (MHT) search for labelling a BN

4.4 Markov chain Monte Carlo sampling

Instead of exhaustively searching the space, MCMC samples the posterior distribution $\pi(\omega) = p(\omega|Y)$ using a Markov chain. The set of possible states in the Markov chain Ω is the set of all global explanations, and 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 [8]. For readers who are not familiar with it, MCMC and the Metropolis-Hastings algorithm are explained in Appendix A.

4.4.1 Markov chain Monte Carlo data association

The work of Oh, Russell and Sastry [111] proposed using Markov Chain Monte Carlo (MCMC) for data association because it scales better than Multiple-Hypotheses Trees (MHT) when the probabilities of different explanations are very close, and the MAP explanation is unlikely to reside amongst the k-best explanations for a reasonable beam width k (Section 4.3). The space of possible explanations Ω is a discrete space, thus moves are designed to change a certain explanation ω into a slightly different one. Each move amends part of the explanation ω, preserving the constraints. After each move is applied, the resulting explanation should still be a valid global explanation. These moves need to be carefully designed to traverse the whole space of possible explanations. They can be simple or complex moves, although complex moves can be achieved via applying a sequence of simpler moves. MCMCDA then starts from any valid global explanation and produces a sample from the posterior distribution of explanations [2]. The sample size equals the length of the Markov chain (n_{mc}).

Assume ξ is the set of all move types. MCMCDA (Algorithm 4.3) amends the general Metropolis-Hastings algorithm (Appendix A.2) to include a prior step for selecting the move type m. Due to the nature of the explanation and its constraints, not all move types are allowed given a certain explanation, thus ξ_i refers to the set of allowed move types given the current explanation ω_i. The algorithm requires a choice of the sample size n_{mc}, as well as an initial element ω_0. At each step, a new explanation is proposed and the acceptance probability α is computed. A sample u is drawn from $\mathscr{U}[0,1]$; the uniform distribution in the closed interval from 0 to 1. The proposed explanation is accepted in the sample if $\alpha > u$.

As data association aims to find the best explanation, rather than sample the distribution of explanations, the best explanation $\hat{\omega}$ is maintained throughout the Markov chain. At each iteration, the chosen sample is compared to the best explanation found so far. The required solution is thus chosen from amongst the sample elements.

$$\hat{\omega} = \arg\max_{i=1..n_{mc}} p(\omega_i|Y) \tag{4.3}$$

[2] The chain should be long enough to guarantee convergence (Appendix A)

```
initialise $\omega_0$
$\hat{\omega} = \omega_0$
for $i$ = 1 to $n_{mc}$ do
    sample m from $\xi_i$
    sample $\omega^\star$ from $Q_m(\omega^\star|\omega_{i-1})$
    calculate $\alpha(\omega^\star|\omega_{i-1}) = \min\left\{1, \frac{\pi(\omega^\star)Q(\omega_{i-1}|\omega^\star)}{\pi(\omega_{i-1})Q(\omega^\star|\omega_{i-1})}\right\}$
    sample $u$ from $\mathscr{U}[0,1]$
    if $u < \alpha(\omega^\star|\omega_{i-1})$ then
        $\omega_i = \omega^\star$
        if $\frac{\pi(\omega_i)}{\pi(\hat{\omega})} > 1$ then
            $\hat{\omega} = \omega_i$
    else
        $\omega_i = \omega_{i-1}$
```

Algorithm 4.3: Markov Chain Monte Carlo Data Association Algorithm

There are two obvious obstacles when using MCMCDA. The first is calculating the proposal distribution Q at each configuration. This is because the choice of the next step is split into selecting a move-type, followed by selecting a specific move of that type. Reversible Jump MCMC (RJMCMC), explained in Section 4.4.2, allows clearer formulations for the proposal distribution and the acceptance probability.

The second obstacle is expecting MCMCDA to find the best explanation while being a sampling technique. Adding simulated annealing is a minor modification, explained in Section 4.4.5, and is tailored to locate the best explanation rather than sample the distribution of explanations. RJMCMC and the addition of simulated annealing have not featured in most of the literature that adopts MCMCDA for radar and visual surveillance.

4.4.2 Reversible jump Markov chain Monte Carlo

Green suggested using MCMC for sampling the joint distribution of both the model dimension and the model parameters [57]. This technique, first called trans-dimensional MCMC and later referred to as Reversible Jump MCMC (RJMCMC), can be used to solve a wide variety of problems where the joint distribution of model dimension and model parameters needs to be optimised to find the best pair of dimension and parameters that suits the observations.

By analogy, given a set of detections Y, the search is for the number of events and which detections belong to each event. There are two ways for using MCMC to find the best explanation. The first approach is to use within-model MCMC where one chain is run for each possible number of events. Within-model MCMC is preferred when the numbers are limited and separate optimisation for each can improve the efficiency. Alternatively, across-model MCMC is expected to converge faster, especially when the number of dimensions is huge [54]. Reversible Jump MCMC (RJMCMC) applies across-model and within-model reversible moves.

Several applications of RJMCMC have been proposed in the literature - for example finding the number and parameters of Gaussians in a Gaussian Mixture [117]. A thorough review of alternatives to RJMCMC can be found in [54]. The main drawback of RJMCMC is the difficulty in designing the move types. Though some moves are general across a collection of applications, most moves are application-specific. It has been conjectured that some reported inefficiencies of RJMCMC have been due to poor design of the reversible moves [8].

RJMCMC generalises the acceptance probability formula in Algorithm 4.3 to include the probability of selecting the move type, and a move-specific probability [58].

$$\alpha(\omega'|\omega) = \min\big(1, \frac{\pi(\omega')}{\pi(\omega)} \frac{j_{m^R}(\omega')}{j_m(\omega)} \frac{g_{m^R}(u')}{g_m(u)} \big| \frac{\partial(\omega',u')}{\partial(\omega,u)} \big|\big) \tag{4.4}$$

In Equation 4.4, j refers to the probability of selecting a move-type. 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. $j_m(\omega) = 0$ for impossible move types that would result in an inconsistent set of events. 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 return to the previous explanation. $\frac{j_{m^R}(\omega')}{j_m(\omega)}$ is the ratio of the probability of selecting the reverse move type (back from the new explanation ω' to ω) to that of selecting the move type from the current explanation.

Using Green's formulations of RJMCMC, each move type m has its own 'within-move' proposal distribution g_m. In Equation 4.4, u refers to the random variables used to transform the current explanation ω to the new explanation ω' using the move type. Some move types result in a change in the explanation's dimension. This is when the new explanation has a different number of recognised events than the previous one. If d is the dimension of the explanation ω, d' is the dimension of the new explanation ω', r is the dimension of the random vector u and r' is the dimension of the random vector required for the reverse move u', then the transformation from (ω,u) to (ω',u') is a diffeomorphism if $d+r = d'+r'$. The last factor in Equation 4.4 is the absolute determinant of the Jacobian matrix of this diffeomorphism. This section will not explain further how the determinant of the Jacobian matrix is handled for the proposed discrete moves. The reader can refer to Smith [131] for proofs.

4.4.3 Designing reversible moves

When using RJMCMC to traverse the space of explanations, a different explanation is proposed at each step along the Markov chain based on the current one. For discrete search spaces, multiple types of moves are needed to traverse the search space [58]. For binary event hierarchies where each production rule in the AMG replaces a symbol by a multiset of two symbols, 4 move types were designed to traverse the search space (Figure 4.7). These connect or disconnect a pair, change one of the linked events or switch two pairs. It should be noted that this is not the minimal set of move types. A change move for example can be constructed from a disconnect move followed by a connect move. Disconnecting would often decrease the posterior probability significantly, which makes it a less probable move along the chain. Accordingly, change and switch move types enable efficient search of the space and faster convergence. Other complex moves can be constructed from a sequence of these moves. The change and switch moves are self-reversible, while the connect and disconnect moves form a reversible pair. They alter the dimension of the explanation by changing the number of compound events. The AMGs introduced in this thesis define binary hierarchies, so these moves were sufficient for the purpose.

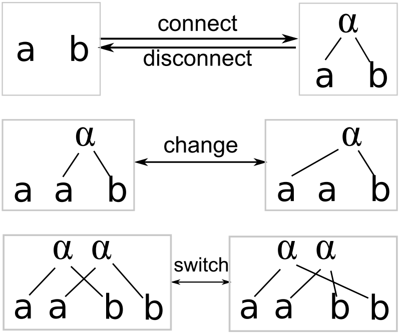


Figure 4.7: Four move types are proposed to link events, break links, change linked events and switch linkages.

RJMCMC splits sampling from the proposal distribution to propose a new explanation into two steps: choosing the move type j_m then choosing a specific move g_m. Uniformly choosing a move type from the set of possible moves ξ_i does not efficiently search the space of explanations. The weighted distribution j_m is thus estimated from the number of distinct moves of each type that can be applied to the current explanation ω_i. Accordingly

$$j_m(\omega_i) = \frac{f(m, \omega_i)}{\sum_{\gamma \in \xi_i} f(\gamma, \omega_i)} \tag{4.5}$$

where $f(m, \omega_i)$ is a function that maps the move type and an explanation to the number of possible moves of that type that can be applied to the explanation. Calculating the number of possible moves does not require enumerating the actual moves, but is estimated from the number of recognised events of each type within the explanation ω_i.

Next, a specific move of that type is chosen and applied to the current explanation. This ‘within-move’ proposal distribution g_m can also be uniform. Alternatively, a customised ‘within-move’ proposal distribution can be designed for each proposed move type. These are application-specific and depend on the expected ambiguities in the observations. Further explanation of these within-move proposal distributions will be given along with the two case studies in Chapters 5 and 6.

4.4.4 Example of searching using RJMCMC

This section explains by example how the space of global explanations can be searched using RJMCMC. The set of discrete moves to traverse the space were introduced in Figure 4.7. For the given BN in Figure 4.2, three layers of compound events are present. This section labels these layers as ‘A’, ‘E’ and ‘G’ based on the compound event they recognise. For simplicity, within-move proposal distributions g_m (see Section 4.4.2) are uniform distributions over the possible moves of each type. The Markov chain can start from any global explanation.

For an initial configuration ω_0, Figure 4.8 shows a 4-steps Markov chain. At each step, a list of move types with the number of possible moves of each type is shown as a label on the arrow. A move type is not mentioned if no possible moves can be found of that type. In the figure, a subscript indicates the layer at which the move is applied. $disconnect_A$, for example, disconnects an a and a b detection that are connected to a compound event A. The proposal distribution $j(\omega_0)$ is a weighted distribution by the number of possible moves of each type. The weighted distribution is randomly sampled and a move type is chosen (bounded by a rectangle). Figure 4.8 shows a sequence of applied moves, regardless of the acceptance probability. In presenting the figure, the parse tree is shown rather than the labeled BN. This presentation suits the moves better. Recall that there is a 1-1 mapping between a labeled BN and a parse tree.

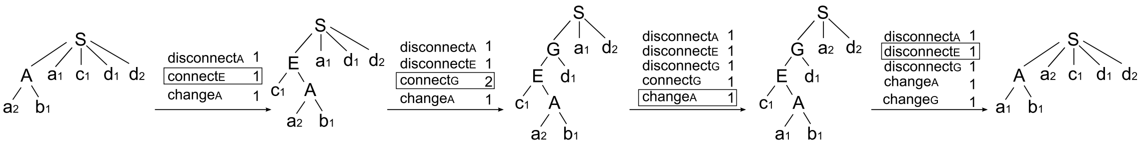
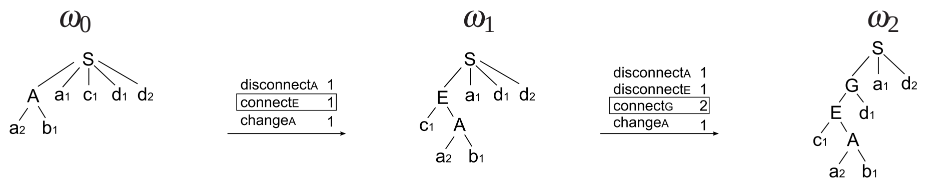


Figure 4.8: Four moves are applied in sequence. The label at each arrow shows the number of possible moves of each type. The rectangle indicates the chosen move type.

Next, Figure 4.9 shows the posterior calculations, along with the acceptance probability α for the first two moves. The figure shows two possible moves and evaluates the acceptance probability for each. For the first move, the ratio $\frac{\pi(\omega_1)}{\pi(\omega_0)}$ shows an increase in the posterior probability. When evaluating the ratio $\frac{j_{m_R}(\omega_1)}{j_m(\omega_0)}$, the numerator is evaluated to $j_{disconnect_E}(\omega_1)$ which is the probability of choosing the reverse move type given the explanation is ω_1. $j_{disconnect_E}(\omega_1)$ equals $\frac{1}{5}$ and can be calculated from the label on the arrow departing from ω_1. The denominator $j_{connect_E}(\omega_0)$ equals $\frac{1}{3}$ given 3 moves are only feasible. As only one move of each type is available, $\frac{g_{m_R}(\omega_1)}{g_m(\omega_0)} = 1$. These calculations guarantee the detailed balance explained in Appendix A.2. The acceptance probability is 1 as the minimum function compares to a ratio higher than 1. According to the RJMCMC algorithm, the move is certainly made, and ω_1 is the next sample element in the Markov chain.



$$\frac{\pi(\omega_1)}{\pi(\omega_0)} = \frac{p(E=t|A=t,c=t)p(O_E|E=t)}{p(E=f|A=t,c=t)p(O_E|E=f)} = \frac{0.8\times 0.9}{0.2\times 0.2}$$

$$\frac{j_{m^R}(\omega_1)}{j_m(\omega_0)} = \frac{j_{disconnect_E}(\omega_1)}{j_{connect_E}(\omega_0)} = \frac{1/5}{1/3}$$

$$\frac{g_{m^R}(\omega_1)}{g_m(\omega_0)} = \frac{1}{1}$$

$$\alpha(\omega_1|\omega_0) = \min\left(1, \frac{\pi(\omega_1)}{\pi(\omega_0)}\frac{j_{m^R}(\omega_1)}{j_m(\omega_0)}\frac{g_{m^R}(\omega_1)}{g_m(\omega_0)}\right) = 1$$

$$\frac{\pi(\omega_2)}{\pi(\omega_1)} = \frac{p(G=t|E=t,d=t)p(O_G|G=t)}{p(G=f|E=t,d=t)p(O_G|G=f)} = \frac{0.5\times 0.4}{0.5\times 0.9}$$

$$\frac{j_{m^R}(\omega_2)}{j_m(\omega_1)} = \frac{j_{disconnect_G}(\omega_2)}{j_{connect_G}(\omega_1)} = \frac{1/5}{2/5}$$

$$\frac{g_{m^R}(\omega_2)}{g_m(\omega_1)} = \frac{1}{1/2}$$

$$\alpha(\omega_2|\omega_1) = \min\left(1, \frac{\pi(\omega_2)}{\pi(\omega_1)}\frac{j_{m^R}(\omega_2)}{j_m(\omega_1)}\frac{g_{m^R}(\omega_2)}{g_m(\omega_1)}\right) = \frac{4}{9}$$

Figure 4.9: The acceptance probabilities α for the first two moves from Figure 4.8 are detailed. The first move is certainly accepted. The second move's acceptance depends on the random uniform sample u (see Algorithm 4.3).

The second move does not increase the posterior. It is accepted with a probability equal to α. When sampling u from the uniform distribution, the move to ω_2 is made if $\alpha > u$. Alternatively, the next sampled element will be ω_1 again.

4.4.5 Adding simulated annealing

MCMC is a sampling technique that aims at producing a sample that approximates the target distribution. MCMCDA (Section 4.4.1) uses sampling to find the global maximum of the target distribution [111], and so is the case with some applications of RJMCMC [117]. Although MCMC ensures more sample elements are chosen from the peak(s) of the distribution, it does not guarantee the maximum is found. Using MCMC for global optimisation is theoretically an approximation, hoping one element in the sample will match the distribution's highest peak.

An alternative method to find the maximum is adding simulated annealing. Simulated Annealing (SA) is a global optimisation technique that simulates the physical process of pre-heated and controlled slow cooling of material crystals. This physical process ensures finding the crystal with the largest size and fewest defects. The SA algorithm by analogy introduces a fictional temperature T, and updates it at each iteration T_i via a cooling schedule. The Markov chain with SA is non-homogeneous and its invariant distribution at each iteration i equals

$$\varphi(\omega_i) = \pi(\omega_i)^{\frac{1}{T_i}} \tag{4.6}$$

With each iteration, the temperature and the target distribution are updated. As T_i decreases, the SA algorithm slowly restricts accepting the move to a lower π value, so it would reach the maximum. SA requires a choice of the cooling schedule. Figure 4.4 displays the MCMC-SA general algorithm.

```
initialise $\omega_0$
initialise $T_0$, $T_{n_{mc}}$
define cooling schedule cool$(T_0, T_{n_{mc}}, i)$
$\hat{\omega} = \omega_0$
for $i$ = 1 to $n_{mc}$ do
    sample m from $\xi_i$
    sample $\omega^\star$ from $Q_m(\omega^\star|\omega_{i-1})$
    update $T_i = \mathrm{cool}(T_0, T_{n_{mc}}, i)$
    calculate $\alpha(\omega^\star|\omega_{i-1}) = \min\left\{1, \left(\frac{\pi(\omega^\star)}{\pi(\omega_{i-1})}\right)^{\frac{1}{T_i}} \frac{Q(\omega_{i-1}|\omega^\star)}{Q(\omega^\star|\omega_{i-1})}\right\}$
    sample $u$ from $\mathscr{U}[0,1]$
    if $u < \alpha(\omega^\star|\omega_{i-1})$ then
        $\omega_i = \omega^\star$
        if $\frac{\pi(\omega_i)}{\pi(\hat{\omega})} > 1$ then
            $\hat{\omega} = \omega_i$
    else
        $\omega_i = \omega_{i-1}$
```

Algorithm 4.4: Markov Chain Monte Carlo with Simulated Annealing Algorithm

The differences between MCMC and MCMC-SA are:

- MCMC guarantees a representative sample of the target distribution, but does not search for the global maximum. MCMC-SA aims at finding the global maximum, but the resulting sample does not approximate the target distribution.

- MCMC only requires choosing the suitable proposal distribution Q. MCMC-SA also expects a suitable cooling schedule. The choice of the cooling schedule is essential for finding the global maximum.

- For multi-peak distributions, the chance of jumping between peaks remains steady in MCMC. In MCMC-SA, the chance of jumping between peaks is higher at the start and decreases constantly as time passes.

When adding simulated annealing and searching using MCMC-SA, the probability of accepting the move from ω to ω' changes along the Markov chain according to the cooling schedule. Assume the ratio $\frac{\pi(\omega')}{\pi(\omega)} = \frac{1}{2}$, Figure 4.10 plots the ratio after applying the cooling $\left(\frac{\pi(\omega')}{\pi(\omega)}\right)^{\frac{1}{T_i}}$ along the chain, using different cooling schedules with $T_0 = 1.5$ and $T_{n_{mc}} = 0.01$. The figure compares the linear, exponential and sigmoid cooling schedules (Equations 4.7 - 4.9).

1. Linear cooling schedule

$$T_i = T_0 - i\left(\frac{T_0 - T_{n_{mc}}}{n_{mc}}\right) \tag{4.7}$$

2. Exponential cooling schedule

$$T_i = T_0\left(\frac{T_{n_{mc}}}{T_0}\right)^{\frac{i}{n_{mc}}} \tag{4.8}$$

3. Sigmoid cooling schedule

$$T_i = \frac{T_0 - T_{n_{mc}}}{1 + e^{0.3(i - n_{mc}/2)}} + T_{n_{mc}} \tag{4.9}$$

When $T_i > 1$, the probability of accepting the move increases. Alternatively, when $T_i < 1$, the algorithm becomes more restrictive to accept moves that decrease the posterior.

4.4.6 Online RJMCMC

RJMCMC can be modified to consider new detections. The global explanation calculated up to now is used to initialise the Markov chain. This means the initial solution ω_0 is the MAP explanation for all the previous detections, along with any consistent labelling of the new ones. It should not be misunderstood that the previous observations cannot be

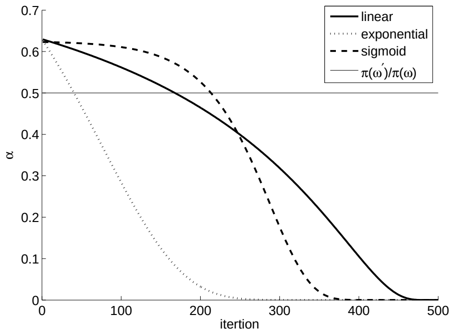


Figure 4.10: Four cooling schedules are compared by plotting the ratio $\left(\frac{\pi(\omega')}{\pi(\omega)}\right)^{\frac{1}{T_i}}$ *across 500 iterations of the Markov chain. The horizontal line shows the original* $\frac{\pi(\omega')}{\pi(\omega)}$*. When the result is higher than this line, the move has a higher acceptance probability. As the Markov chain progresses, the chance of accepting this move decreases. In the figure* $T_0 = 1.5$ *and* $T_{n_{mc}} = 0.01$.

re-considered (as in the case of MHT). The same set of moves is applied and could affect any of the new or original detections.

To speed convergence, the Markov chain can be run in two phases. The first phase is confined to moves that involve new detections. This phase locates a local optimum involving the new detections. The second phase runs the RJMCMC in the normal fashion introducing changes to the global explanation of all detections. This technique is similar to the burn-in sampling idea used in Markov chains, where initial samples affected by the starting position are discarded to speed convergence [8]. If only the second phase was used, the chance of the moves involving new detections decreases as more detections are added. The first phase cannot be run alone to achieve the global maximum. This is because the best explanation can introduce changes to previous events, which might accordingly introduce more changes to other events. During experimentation, the Markov chain had to be run for much longer when only one phase was run. As the number of detections increased, the chance of proposing a move that involves the new data tended to decrease, and the length of the required Markov chain had to be increased. The two-phase solution was able to solve this problem. The length of the Markov chain in the first phase is set to a factor γ of the Markov chain's length. γ was set to 0.25 in all the experiments.

4.5 Integer programming

While the three methods explained above are heuristic, i.e. they cannot guarantee the MAP is found. Integer Programming (IP) is an exhaustive technique that finds the MAP explanation. The next subsection explains the basics of integer programming, while Section 4.5.2 illustrates how IP can search the BN of global explanations for the MAP explanation.

4.5.1 Introduction to integer programming

An integer program is generally given in the following format [91] [3]

Given a matrix $A \in \mathbb{R}^{m \times n}$, and two column vectors $b \in \mathbb{R}^m$, $v \in \mathbb{R}^n$
Find max $v^T x$ such that
$Ax \leq b$, and
$x \in \mathbb{Z}^n$

All combinatorial optimisation problems can be formulated as integer programs [91], yet these are NP-hard problems, which cannot be solved in polynomial time. A linear (rather than integer) solution can be found if the integrality constraint is dropped. The problem would thus be $\{\max v^T x : Ax \leq b\}$. This is referred to as the linear relaxation of the integer program. Polynomial-time algorithms have been developed for solving linear programs [125]. For the linear program, a feasible solution is $x \in \mathbb{R}^n$ such that $Ax \leq b$. The space of feasible solutions is the intersection of many half spaces, given a finite number of linear inequalities. This set is a polyhedron.

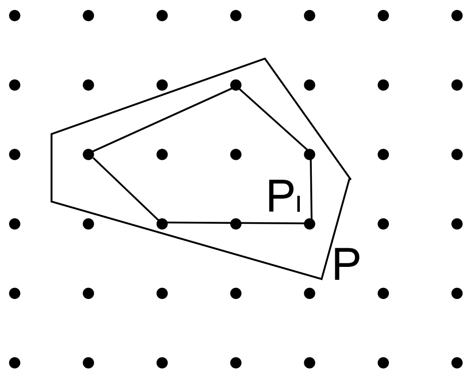


Figure 4.11: For a solution space, the polyhedron P is the solution space for the linear program found by relaxing an integer program, while P_I is the convex hull of P and represents the solution space for the integer program. Diagram from [91]

Figure 4.11 shows that once the polyhedron which represents the solution to the linear program P is found, the solution to the corresponding integer program is the convex hull of integer vectors P_I. Techniques for finding the convex hull given the polyhedron P have been proposed, such as branch and bound and cutting planes [91]. In branch-and-bound techniques, the solution to the linear program is assessed for integrality. For each

[3] The cost vector v is usually represented by the symbol c. This was not used here to avoid ambiguity with the detections.

$0 < x_i < 1$, the tree is branched with two options $x_i = 0$ and $x_i = 1$ [143]. Two new solutions are investigated, and further branching is done until the integer solution is found. Alternatively, cutting planes finds a polyhedron P' by cutting off certain parts of P, maintaining $P_I \subset P' \subset P$. If $\{\ \max v^T x : x \in P'\}$ is an integral vector, the sought solution to the integer program is found. Alternatively, P' is cut again to obtain P'' until the integral solution is found [91].

4.5.2 IP for searching a constrained BN

To use a linear solver, one must formulate the problem as an integer program. Morefield [105] first formulated the multi-target tracking problem as an integer program using implicit enumeration. In implicit enumeration, the list of all partial explanations $\mathcal{F}$ is accumulated, and a solution to the problem is an integer vector of all these partial explanations. Assume there are r partial explanations in $\mathcal{F}$, the explanation ω is then an r-dimensional vector of 0s and 1s. If ω_i (the ith component of ω) is set to 1 then the corresponding partial explanation $\lambda_i \in \mathcal{F}$ is part of the chosen set of consistent events making the explanation ω. Alternatively, a component of ω set to 0 corresponds to a possible partial explanation that is not considered. To illustrate, assume there are 5 partial explanations, then the vector ω = [0 1 1 0 1]^T means the second, third and fifth partial explanations make up the global explanation.

To understand this representation, one must explain what a partial explanation is. In the case of global explanations for activities, a partial explanation is one event from the possible set of events. Recall from Section 3.2 that the first production rule rewrites the start symbol S as a multiset of other terminal and nonterminal symbols, for example: S $\rightarrow$ A*, B*, $a^\star$, $b^\star$. The given options are the types of events in this activity. The set of all possible A compound events, B compound events, along with any primitive events that can be left ungrouped equals the set of all partial explanations $\mathcal{F}$. For the detections 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:

$\lambda_0 : a_1$

$\lambda_1 : a_2$

$\lambda_2 : c_1$

$\lambda_3 : c_2$

$\lambda_4 : B_1, b_1, c_1$

$\lambda_5 : B_2, b_1, c_2$

$\lambda_6 : A_1, a_1, B_1, b_1, c_1$

$\lambda_7 : A_2, a_2, B_1, b_1, c_1$

$\lambda_8 : A_3, a_1, B_2, b_1, c_2$

$\lambda_9 : A_4, a_2, B_2, b_1, c_2$

The probability of each partial explanation can be calculated independently. Assume v is an r-dimensional real-valued vector where v_i is the $\log(p(\lambda_i))$ of the partial explanation λ_i. The search for the MAP using implicit enumeration would be to find $\max v^T \omega$. This is because

$$v^T \omega = \sum_{i:\omega_i=1} v_i = \sum_{i:\omega_i=1} log(p(\lambda_i)) \tag{4.10}$$

Accordingly, ω_1 = [0 1 0 1 0 0 1 0 0 0]^T and ω_2 = [1 1 1 0 0 1 0 0 0 0]^T correspond to the two parse trees in Figure 4.12. The posterior of each explanation is simply $v.\omega_1$ and $v.\omega_2$.

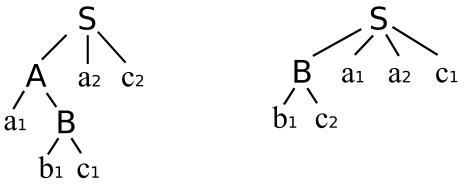


Figure 4.12: Two parse trees given a multiset of detections and AMG G_a.

While maximising $v^T.\omega$, some of the r-dimensional vectors are an inconsistent or incomplete set of events, like the vectors ω_3 = [1 0 1 0 0 0 0 0 0]^T and ω_4 = [1 1 0 0 1 1 1 0 0]^T. ω_3 is incomplete, as each explanation should explain all the detections. ω_4 is inconsistent as it violates the constraints in the sample AMG. The solution to the IP problem should include constraints that ensure the resulting set of events makes up a global explanation. Three types of constraints can be defined for the global explanations:

- All terminals should be explained - referred to as the 'terminal constraints'.
  τ is a matrix of size $\mid D \mid \times r$. Each cell τ_{ij} has the value 1 if the terminal i is explained in the partial explanation j. To ensure each terminal is explained at least once, the constraint $\tau.\omega \geq \mathbf{1}$ should be maintained, where $\mathbf{1}$ is a vector of 1s of dimension $\mid D \mid$.

- A maximum of one of the inter-dependent nodes, that constrain a common inherited attribute, is allowed - referred to as the 'consistency constraints'.
  θ is a matrix of size $m \times r$ where m is the number of inter-dependent node sets, and equals the number of deterministic nodes in the BN. Each cell θ_{ij} is of value 1 if one of the inter-dependent nodes of set i is explained in the partial explanation j. The constraint would then be $\theta.\omega \leq \mathbf{1}$.

- Nodes should have the same label in all the different partial explanations - referred to as the 'conflict constraints'.

κ is the node labelling constraint. This matrix of size $n \times r$ where n is the number of nodes in the BN. Each cell κ_{ij} has a value 0 if the node i is not labeled in explanation j, 1 if it has a true label, and 2 if it's labeled false. Note that extra possible values can be added for multi-labelled BNs. To ensure the node is labeled correctly, the following non-linear constraint should be added for each node i in the BN

$$\sum_{j=1}^{r} \sum_{k=j+1}^{r} (\kappa_{ij} \neq \kappa_{ik}).(\kappa_{ij} \neq 0).(\kappa_{ik} \neq 0).\omega_j.\omega_k = 0 \tag{4.11}$$

The constraint in Equation 4.11 is non-linear, and cannot be solved by a linear solver. This constraint can be converted to a set of linear constraints. For each $\delta_{jk} = \omega_j \omega_k$, then three linear constraints can ensure δ_{jk} equals 1 only when both ω_j and ω_k equal 1.

$$\delta_{jk} \leq \omega_j \tag{4.12}$$

$$\delta_{jk} \leq \omega_k \tag{4.13}$$

$$\delta_{jk} \geq \omega_j + \omega_k - 1 \tag{4.14}$$

For each δ_{jk}, a constraint would check that

$$\sum_{i=1}^{n} (\kappa_{ij} \neq \kappa_{ik}).(\kappa_{ij} \neq 0).(\kappa_{ik} \neq 0).\delta_{jk} = 0 \tag{4.15}$$

Algorithm 4.5 shows the steps of generating the set of partial explanations $\mathcal{F}$, and the three constraints matrices: τ, θ, κ. Though these constraints make the set of all needed constraints, the complete set of these constraints has redundancies. If a terminal a is constrained to be consumed once, then no conflict would be expected, and both the first and the second constraints can be substituted by $\tau_a \omega = \mathbf{1}$. Similarly, if a nonterminal is constrained to one, then it can be dropped from the check for conflict constraint. This decreases the number of constraints significantly.

Next, Algorithm 4.6 shows how the integer program can be formulated and solved. Instead of finding an integer solution (0s and 1s), linear relaxation substitutes this with a linear constraint $0 \leq \omega_i \leq 1$. After finding the linear answer, techniques such as branch-and-bound can correct non-integer values in the solution. In solving the problem, two solvers were employed. The first is part of the Optimisation Toolbox of MATLAB. It is based on branch-and-bound algorithm [103]. The second solver, XPRESS-MP, tries a collection of breadth-first, depth-first, best-first branch-and-bound techniques along with

```
input  : Bayesian Network BN, AMG G = (N,T,S,A,P), detections multiset D
output : Partial Explanations $\mathcal{F}$, Terminal constraints $\tau$, Consistency constraints $\theta$,
         Conflict constraints $\kappa$
Let $H$ be the set of all deterministic random variables in BN
Let $P_s : S \rightarrow X_1, X_2, ..., X_{n_{ps}}$ be the production rule rewriting the start symbol S
Let i = 0 be the counter for partial explanations
foreach $x \in X_1, X_2, ..., X_{n_{ps}}$
    Let Nodes$_x$ be the set of all nodes in BN of type $x$
    foreach $y \in Nodes_x$
        $\lambda_i$ := y $\cup$ pa$_y$, where $pa_y$ is the set of all ancestors of $y$
        $v_i = \log(p(\lambda_i))$
        foreach $d \in D$
            if $d \in \lambda_i$ then
                $\tau_{di}$ = 1
            else
                $\tau_{di}$ = 0
        foreach $h \in H$
            if $\lambda_i$ constrained by $h$ then
                $\theta_{hi}$ = 1
            else
                $\theta_{hi}$ = 0
        foreach Node $n \in BN$
            if $n$ labeled true in $\lambda_i$ then
                $\kappa_{ni}$ = 1
            if $n$ labeled false in $\lambda_i$ then
                $\kappa_{ni}$ = 2
            if $n$ not labeled in $\lambda_i$ then
                $\kappa_{ni}$ = 0
        i = i + 1
$\mathcal{F} = \cup_i \lambda_i$
```

Algorithm 4.5: Integer Programming (IP) - implicit enumeration

advanced cutting-plane strategies [45]. To use XPRESS-MP, the integer program is formulated using the modelling language MOSEL. The MOSEL program for the problem in Section 4.1 is shown in Appendix D.

4.6 Comparing the search techniques

Table 4.2 compares the techniques introduced in this chapter based on four aspects. The first aspect is the type of search. The first four techniques are heuristic, as they do not

```
input  : Partial Explanations $\mathcal{F}$, Terminal constraints $\tau$, Consistency constraints $\theta$,
         Conflict constraints $\kappa$, Cost vector $v$
output : $\omega_{IP}$: labelling of the BN
Let r be the number of partial explanations $\mathcal{F}$
Let $\omega$ be an r-dimensional vector of 0s and 1s
$\max v^T.\omega$
$\tau.\omega \geq \mathbf{1}$
$\theta.\omega \leq \mathbf{1}$
foreach $j$ = 1..r
    foreach $k$ = j+1..r
        $\delta_{jk} \leq \omega_j$
        $\delta_{jk} \leq \omega_k$
        $\delta_{jk} \geq \omega_j + \omega_k - 1$
        $\sum_i (\kappa_{ij} \neq \kappa_{ik})(\kappa_{ij} \neq 0)(\kappa_{ik} \neq 0)\delta_{jk} = 0$
% finding $\omega_{IP}$
run linear solver
foreach $j$ = 1..r
    if $\omega_j$ = 1 then
        $\lambda_j$ is part of $\omega_{IP}$
```

Algorithm 4.6: Formulating the problem as an integer program

search the full space of explanations. Integer Programming, on the other hand, is exhaustive as it searches for the set of partial explanations that maximise the posterior while satisfying the constraints.

The second aspect is the randomness. While greedy, MHT and IP produce the same result every time they are run, RJMCMC and RJMCMC-SA have an element of randomness that might change the obtained explanation between different runs of the algorithm.

The table also compares the ability of the technique to search in an ‘online’ fashion. The algorithm is online if it builds on the already-found explanation when new detections are added. The greedy algorithm is not ‘online’, as all the detections are evaluated before the next layer is considered. The MHT algorithm is in essence online. This is because it considers the detections in a sequence, and builds on previously labelled RVs. The RJMCMC can be online as described in Section 4.4.6, and similarly for the simulated annealing addition (RJMCMC-SA). The IP algorithm is offline as it re-evaluates the complete solution when new detections are added.

The table also details any parameters the algorithms require. Greedy and IP searches do not require any parameters. MHT is pruned to the k-best branches, and the choice of k represents a trade-off between accuracy and resources. RJMCMC expects the length of the Makrov chain to be known, an initial explanation, and specifying the ‘within-type’ proposal distributions which are application-dependent. RJMCMC-SA requires the same parameters as RJMCMC in addition to a choice of the cooling schedule.

	type	random?	online/offline	parameters
Greedy	heuristic	no	offline	-
MHT	heuristic	no	online	k
RJMCMC	heuristic	yes	online/offline	n_{mc}, x_0, g_m
RJMCMC-SA	heuristic	yes	online/offline	n_{mc}, x_0, g_m, cool (T_0, $T_{n_{mc}}$, i)
IP	exhaustive	no	offline	-

Table 4.2: Comparing different search techniques presented in the chapter

Section 4.1 presented a sample BN. As a precursor to the comparison on real data in the next two chapters, the quantitative results of searching this BN using all the techniques are compared here. Table 4.3 shows $-\log(p)$ for the recorded results; maximising the posterior is equal to minimising $-\log(p)$. The table shows that the greedy search was unable to find the MAP, and that RJMCMC-SA finds the MAP with $\sigma = 0.0$ which is a better result than sampling using RJMCMC [4].

	Greedy	**MHT**	**RJMCMC**		**RJMCMC-SA**		**IP**
		k=3	μ	σ	μ	σ	
MAP	13.80	9.88	10.32	1.29	9.88	0.0	9.88

Table 4.3: $-\log(p)$ for the MAP solution for the sample BN searched using the different search methods discussed in this chapter.

4.7 Conclusion

The space of global explanations was transformed into a Bayesian network in Chapter 3. The set of labellings with a positive posterior corresponds to the space of explanations. Enumerating all labellings to find the Maximum a Posteriori (MAP) solution is intractable, in most cases. Thus, tractable methods are needed to search the space and find the MAP explanation.

This chapter presented four techniques to search the space of global explanations. For each technique, an algorithm is presented and applied to a sample BN of exponential complexity. The search techniques were: greedy, MHT, RJMCMC-SA and IP. The result of the greedy search forms a baseline for heuristic search techniques. Multiple-Hypotheses Tree (MHT) retains the best k explanations as detections are considered sequentially. Section 4.4 explains MCMCDA and the Reversible-Jump formulations. It shows how adding

[4] 100 runs of 10 parallel independent chains ($n_{mc} = 30$)

simulated annealing targets finding the MAP solution rather than sampling the distribution, and proposes general moves that can traverse the space of binary event hierarchies.

For comparison with exhaustive search techniques, finding the MAP explanation was formulated as an integer program. Section 4.5 reviews integer programming and presents an algorithm to transform the BN inference to an integer program, where the set of partial explanations is first enumerated, along with the posterior of each explanation. Each partial explanation is internally consistent, and the global explanation is one that explains all the detections satisfying the inter-activity constraints. The section presents an algorithm for transforming an AMG given a set of detections into an integer program. Linear solvers can then find the best global explanation.

The search techniques presented in this chapter are experimentally compared in the next two chapters for the *Bicycles* problem and the *Enter-Exit* problem.

Chapter 5

Case I: The *Bicycles* Problem

This chapter presents the first of two case studies of the framework presented in Chapter 3. The first case study, the *Bicycles* problem, concerns activity in a bicycle rack over a full day. The activity is first described textually, and in Section 5.2 it is formulated as an attribute multiset grammar. The AMG combines detections of two types, people and bicycle-clusters, into a two-layer hierarchy. Next, the Bayesian network structure is built from the AMG given a set of detections as explained in Chapter 3. Priors and conditional probabilities are based on expert knowledge, and adapted to a training set.

In accordance with the proposed framework, the set of synthetic attributes required to recognise each event is calculated from certain visual features. Section 5.3 shows how these features are obtained and how the likelihoods are trained. The method was tested extensively on 5 full days from two different sites. One testing site was located in the campus of the University of Leeds, and the other one was outside Cambridge railway station. The dataset is described in Section 5.5. The Maximum a Posteriori solution of the BN is obtained using the various search techniques from Chapter 4. The results (Section 5.6) demonstrate the ability of the framework to recognise the activity in a bicycle rack.

5.1 The *Bicycles* problem

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

Ambiguities in the recognition process increase with occlusion when multiple individuals approach the racks. Due to occlusion and clutter, one cannot always be sure about the event in which each person participated. Yet some evidence can be gathered from the change in foreground blob size along the person’s trajectory, the changes within the rack area and spatial proximities. These time intervals, during which one or more people are simultaneously inside the rack area, are called **activity units**, consistent with the terminology in [56] for plane refueling scenes. Figure 5.1 illustrates an example of an activity unit by highlighting the detected people and the bicycle-clusters. Within each activity unit, the explanation is constrained so each person is linked to one bicycle-cluster at most. This emerges from the natural constraint that a person cannot drop/pick more than one bicycle per visit to the racks. On an even higher level, if both a drop and a pick of the same bicycle are observed, the solution should link the drop event to its subsequent pick event generating a higher-level compound event. Each drop can be connected to one pick at most from a later activity unit, and vice versa.

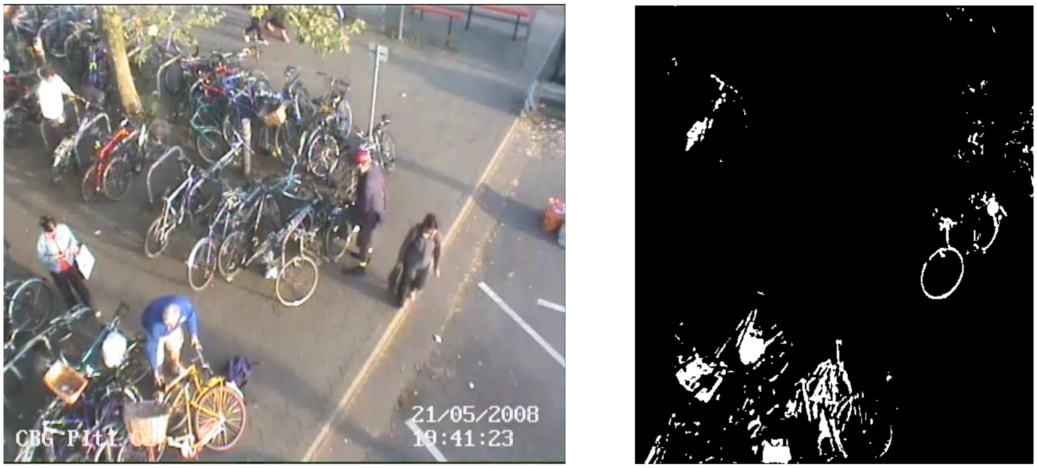


Figure 5.1: An example of an activity unit showing 5 individuals (left) and several bicycle-clusters (right).

It should be mentioned that the *Bicycles* problem is harder than other parking scenarios like those in car parks. This is because bicycles are parked very close to each other, and are sometimes ‘piled’ on top of one another. This makes the *Bicycle* problem a challenging one, which would benefit from pursuing global explanations. The complexity of this problem demonstrates the generality and capabilities of the framework.

For a given input video, this section explains how the detections are collected. Notice that if a detector fails to detect a person or a bicycle-cluster, that detection will not be included in the global explanation.

5.1.1 Detecting people

The input data for the *Bicycles* problem is video recorded from one static camera that is assumed to be mounted high above the ground. Figure 5.2 shows an example of such a viewpoint. An off-the-shelf blob tracker is used [100]. This tracker uses a per-pixel background model, based on the work of Stauffer and Grimson [134], together with a simple foreground model. It assigns a unique identifier to each object moving over a continuous trajectory. It requires an estimate of object size in addition to extra parameters that are tuned depending on the noise level in the image sequence. Examples of the retrieved trajectories are found in Figure 5.3. The tracker incorporates shadow removal by dropping any pixel with colour similar to the background model at the pixel.

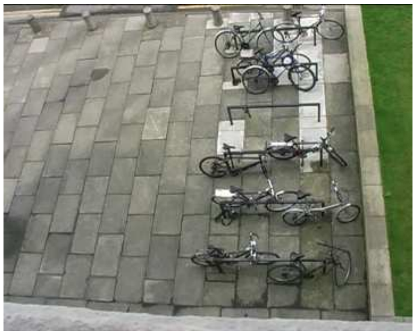


Figure 5.2: Example of the camera’s viewpoint.

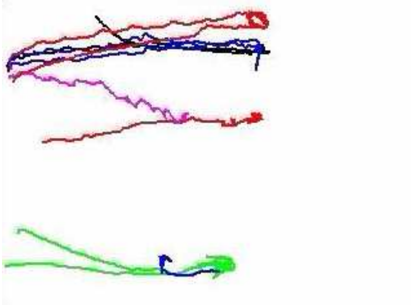


Figure 5.3: Retrieved trajectories for the viewpoint in Figure 5.2.

For each person, the foreground pixels’ position and colour are retrieved for each frame during the time the person is visible. Only people that enter the manually bounded rack area for longer than a certain duration, are considered. The extent of the rack area is represented by a convex polygon [1]. It is assumed that each individual can be tracked separately for some time. Tracked groups cannot be segmented, and they would be identified as a single detection.

[1] An efficient implementation to find whether a point is inside a polygon can be found at [48]

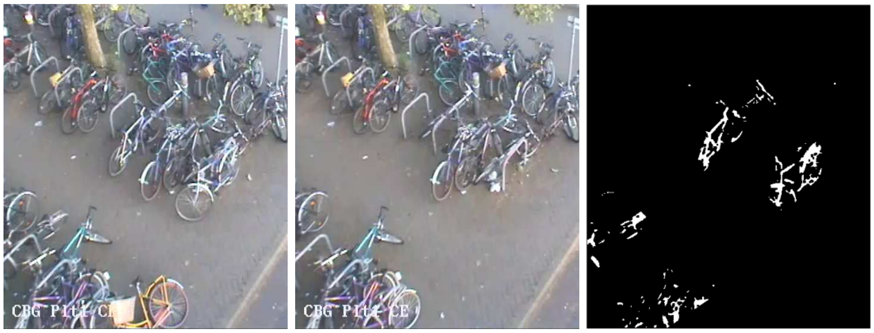
The tracker has been extended for this application to deal with obvious errors in the trajectories. Trajectories generated are often broken during occlusion or when individuals are walking in close proximity. Moreover, the trajectory of a person dropping a bicycle is often broken after the bicycle is left in the rack area. This is because the foreground blob representing the person and the bicycle is split into two, and the tracker assigns a new id to one of the two blobs. Broken trajectories that are similar in their colour and reappear within allowable spatial and temporal ranges are merged. Trajectories with spatial jumps are split if the colour profile is dissimilar before and after the discontinuity.

It should be mentioned that a person detection starts from the first appearance of a moving blob within the camera's field of view and ends when the person departs the scene or is fully occluded. If the same person returns again to the field of view, it is considered a new detection by the tracker. This is because the tracker does not maintain the identity after the person leaves the field of view. Thus, a *person detection* is in effect a single continuous appearance of the person. If a person appears multiple times, different unconnected detections are retrieved by the tracker.

5.1.2 Detecting bicycle-clusters

The motion tracker cannot be used to identify static objects. Therefore to detect bicycles, 'before' and 'after' reference images of the rack area are compared, thereby revealing changed pixels, representing objects that have been deposited and removed. This is in fact a 'change detector' as it simply records the change within the rack area between two points in time. The 'before' reference image is automatically stored whenever the tracker identifies a person approaching the rack area. A flag is set to automatically record the 'after' reference image once the rack area is cleared again. If one or more people enter the area prior to the departure of the first person, the 'after' reference image is only taken after all have departed. The reference images thus record the upper and lower limits of the activity units. Figure 5.4 shows the 'before' and 'after' reference images and the differences by subtracting the pixels, along with some morphological operations like erosion, dilation and closing. The morphological operations attempt to enclose the bicycle's pixels in one cluster. Notice that the changed pixels can signify a dropped or a picked bicycle.

The changed image pixels are then grouped into connected regions representing several clusters. Multiple bicycles can be dropped/picked within one detected cluster. The risk of changes due to noise or lighting effects is minimised by taking reference images before a person enters the rack area and after departing. It cannot be completely ignored though. Figure 5.5 shows some cases where a cluster contains multiple bicycles or no bicycles. A *bicycle-cluster detection* is thus a connected component of changed pixels containing an unknown number of dropped or picked bicycles.



(a) (b) (c)

Figure 5.4: Before (a) and after (b) reference images, revealing changed pixels (c) that signify 3 picked bicycles and one dropped bicycle.

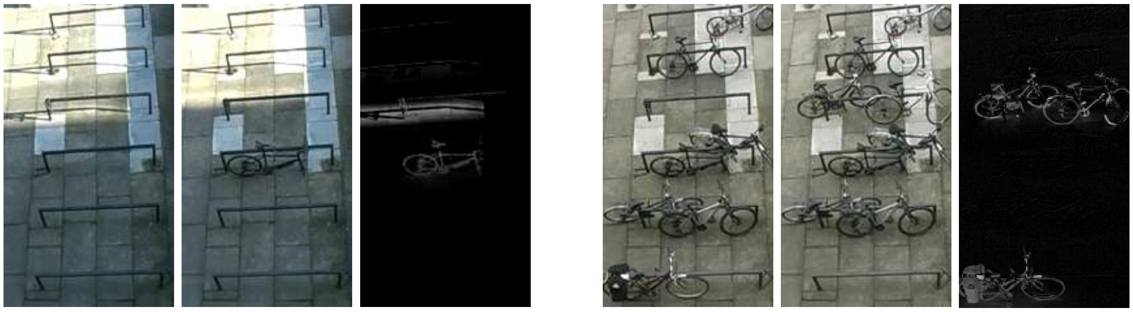


Figure 5.5: The left example shows a noise blob caused by lighting changes. The right example shows one bicycle-cluster made up of 2 bicycles.

It should be mentioned that object detection based on appearance could be used to detect bicycles in static images, using supervised or semi-supervised learning. The PASCAL challenge, for example, presents a suitable dataset of bicycles [41]. This approach was not tried because the camera's viewpoint results in very different bicycle appearances, and the cluttered environment makes it difficult to recognise individual bicycles. Figure 5.6 shows a collection of viewpoints and cluttered scenarios contained in the dataset.

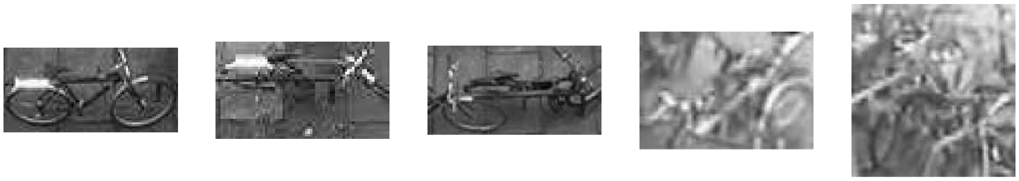


Figure 5.6: A collection of bicycles detected from different viewpoints.

5.2 An AMG for the *Bicycles* problem

This section formally defines an Attribute Multiset Grammar for the activity in a bicycle rack area. The terminal and nonterminal symbols, along with attributes for each symbol, are listed. The attributes are explained and grouped into synthetic and inherited attributes. Functions defined by the AMG are listed before the set of production rules. Refer to Section 3.2 for the AMG formulation and notations.

Terminals (T):	x	person detection
	y	bicycle-cluster detection
	u	an unobserved drop or pick event
Nonterimanls (N):	S	Start symbol representing the global explanation
	V	Drop-Pick: relates a drop event to a later pick
	Z	Drop or pick: relates a person to a bicycle-cluster

Attributes (A):

symbol	att. name	type[2]	domain	description	pdf[3]
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	
	n	A_0	$\mathbb{Z}$	number of frames with the person visible	
	traj	A_0	$\mathbb{Z}^{4n}$	bounding boxes representing the extent of the person in each frame	
	sizeRatio	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	p(x.sizeRatio|x)[4]
	count	A_1	{0,1}	number of events in which the person participates	
	action	A_1	{drop (d), pick (p), pass-by (s)}		
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	
	edgeRatio	A_0	$\mathbb{R}$	ratio of new to removed edges within the cluster	p(y.edgeRatio|y)
	count	A_1	$\mathbb{Z}^*$	inferred number of bicycles in the bicycle-cluster	
	action	A_1	{drop (d), pick (p), noise (n)}		
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	
	edgeRatio	A_0	$\mathbb{R}$	= y.edgeRatio	
	fMap	A_0	Image	= y.fMap	
	dist	A_0	$\mathbb{R}$	spatial proximity between a person and a bicycle-cluster	p(Z.dist|Z)
	count	A_1	{0,1}	number of drop-pick events in which this event participates	
	action	A_1	{drop (d), pick (p), f}		
V	clustOverlap	A_0	$\mathbb{R}$	pixel overlap between the dropped and the picked bicycle-clusters	p(V.clustOverlap|V)

[2] A_0 are synthetic attributes, while A_1 are inherited attributes.

[3] pdf: the probability density function for the synthetic attribute values given the possible actions. Training is required for these pdfs.

[4] This should be written as p(x.sizeRatio|x.action) but x was used for a more concise representation.

pos	A_0	$\mathbb{Z}^4$	bounding box of the intersection area between the dropped and the picked bicycle-clusters	
psDropDist	A_0	$\mathbb{R}$	post-segmented distance for the drop event	p(V.psDropDist$\|$V)
psPickDist	A_0	$\mathbb{R}$	post-segmented distance for the pick event	p(V.psPickDist$\|$V)
psDropEdges	A_0	$\mathbb{R}$	post-segmented edge ratio for the drop event	p(V.psDropEdges$\|$V)
psPickEdges	A_0	$\mathbb{R}$	post-segmented edge ratio for the pick event	p(V.psPickEdges$\|$V)
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 (Section 5.3.3)
$\psi_{co}(Z_1.fMap, Z_2.fMap)$	calculates the overlap in foreground map between the dropped and the picked bicycle-clusters (Section 5.3.4)
$\psi_{edgeRatio}(y.edgeRatio, y.pos)$	calculates the ratio of new to removed edges within a particular rectangular area (Section 5.3.5)

Production Rules (P)

	Syntactic Rule (r)		Attribute Rules (M)		Attribute Constraints (C)		
p_1	S $\rightarrow$ V*, x*, y*	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.clustOverlap	=	ψ_{co} (Z$_1$.fMap, Z$_2$.fMap)			
		V.pos	=	Z$_1$.pos $\cap$ Z$_2$.pos			
		V.psDropDist	=	ψ_{dist} (Z$_1$.traj, V.pos)			
		V.psPickDist	=	ψ_{dist} (Z$_2$.traj, V.pos)			
		V.psDropEdges	=	$\psi_{edgeRatio}$ (Z$_1$.edgeRatio, V.pos)			
		V.psPickEdges	=	$\psi_{edgeRatio}$ (Z$_2$.edgeRatio, 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.psDropDist	=	Z.dist			
		V.psPickDist	=	1			
		V.psDropEdges	=	Z.edgeRatio			
		V.psPickEdges	=	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.psDropDist	=	1			
		V.psPickDist	=	Z.dist			
		V.psDropEdges	=	1			
		V.psPickEdges	=	Z.edgeRatio			
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.edgeRatio	=	y.edgeRatio
Z.fMap	=	y.fMap
Z.dist	=	ψ_{dist} (x.traj, y.pos)
x.count	=	1
y.count	=	y.count+1

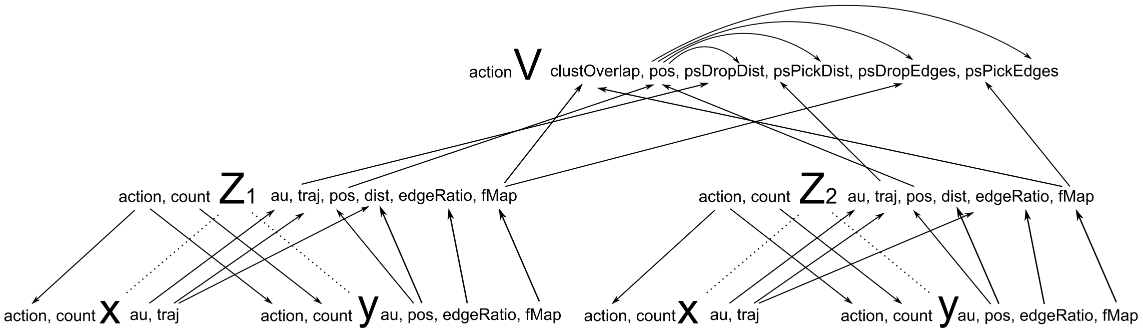

action V clustOverlap, pos, psDropDist, psPickDist, psDropEdges, psPickEdges
action, count Z1 au, traj, pos, dist, edgeRatio, fMap
action, count Z2 au, traj, pos, dist, edgeRatio, fMap
action, count x au, traj
action, count y au, pos, edgeRatio, fMap
action, count x au, traj
action, count y au, pos, edgeRatio, fMap

Figure 5.7: The attribute dependency graph for the Bicycles problem AMG.

Figure 5.7 presents the attribute dependency graph for the AMG. After presenting the AMG, Algorithm 3.1 is used to build the Bayesian network given the set of detections. Figure 5.8 represents this two-layered activity for 3 people and 3 bicycle-clusters. Events within each activity unit are surrounded with a dotted frame for clarity. The AMG specifically constrains drop and pick events between people and bicycle-clusters detected within the same activity unit ($x.au = y.au$ in p_5). Moreover, possible drops are only linked to picks in later activity units ($Z_1.au < Z_2.au$ in p_2).

The Boolean unobserved node ‘u’ is labeled true if an open world assumption is considered. Alternatively, if ‘u’ is labeled false, all drop and pick events are forced to be linked and the world is assumed closed. This would be used if the input video starts from an empty rack area and ends in an empty rack area again, which is an unrealistic assumption in real datasets. Some drops remain unlinked, indicating the bicycle is still within the racks, and some picks are related to drops that occurred before the observation period. While introducing this node might be seen as hallucinating connections that do not exist, it provides a more specific parse tree, and enables switching between open and closed world assumptions. Connecting a drop event to an unobserved pick indicates that either the pick did not occur yet, or the relevant detections were not retrieved by the detector. An alternative approach is to rewrite the activity (represented by the start symbol S) into drops and picks without introducing the unobserved event. This is left for the designer, and here the unobserved node was added for an explicit modelling of unobserved connec-

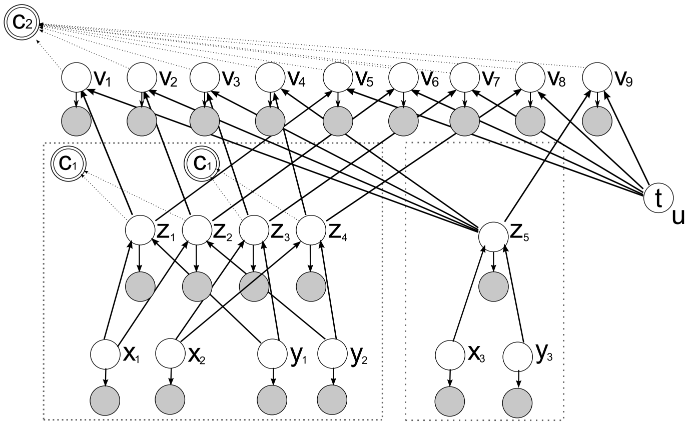


Figure 5.8: The structure of the Bayesian network for the Bicycle Problem given a set of detections. Dotted boxes surround activity units (not to be confused with plate diagrams). Detected people (x) and bicycle-clusters (y) are linked within activity units to explain drops and picks. Events are linked in a second layer to explain drop-pick events. Explanations at each layer are constrained by deterministic RVs.

tions. Figure 5.9 shows a parse tree of the AMG along with a labeled Bayesian network.

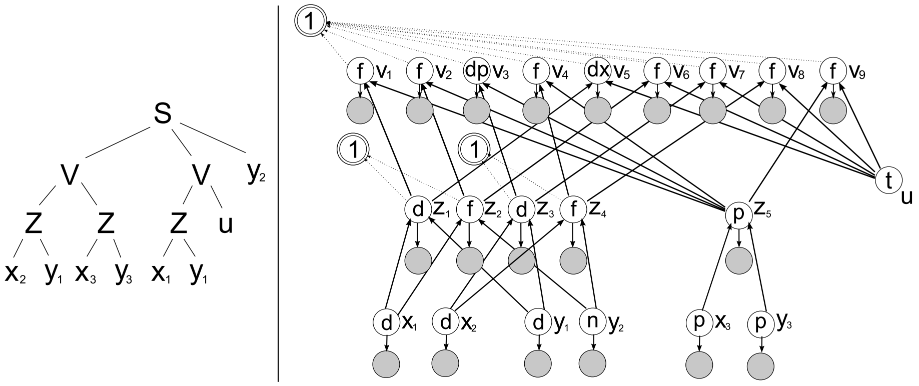
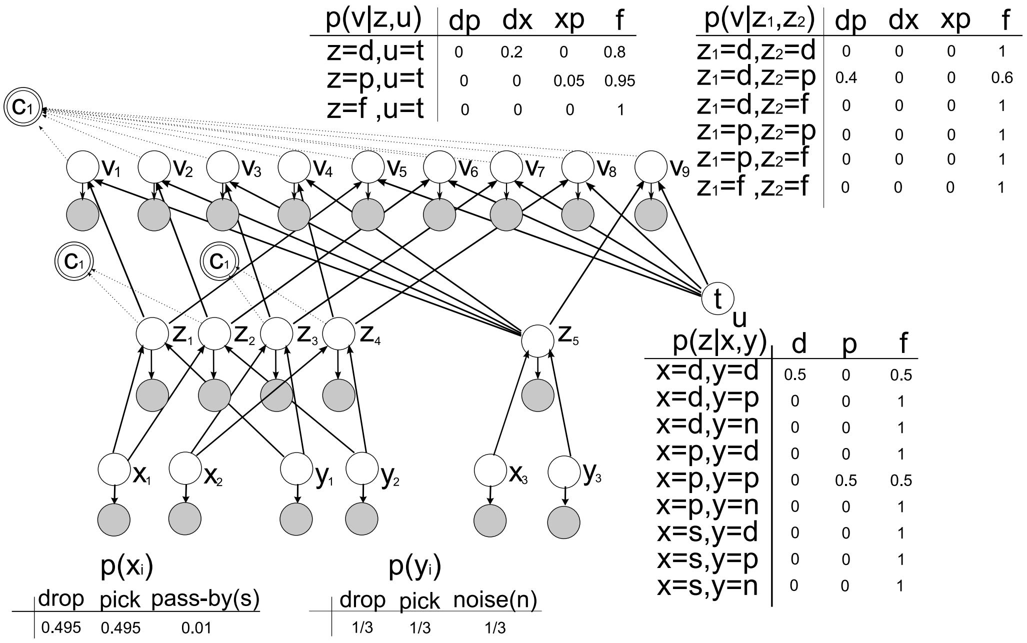


Figure 5.9: A sample parse tree and the corresponding labelled BN.

Figure 5.10 presents the complete Bayesian Network (BN) showing priors and conditional probabilities. These have been estimated using expert knowledge from the training sequence and the corresponding hand-generated ground truth. They were kept constant for all other sequences (Section 5.6).

To realise the size of the search space, one can evaluate the number of hidden random variables for a given set of people and bicycle-cluster detections. For each activity unit $k = 1,2,..,n$, assume α_k is the number of people detected in this activity unit, and β_k is the number of bicycle-clusters. The number of hidden Random Variables (RV) in the activity unit equals $\alpha_k + \beta_k + \alpha_k \beta_k$. The product $\alpha_k \beta_k$ equals the number of drop and pick events Z_k within the activity unit. For the next hierarchical level, each Z_k can connect to



p(v|z,u)	dp	dx	xp	f
z=d,u=t	0	0.2	0	0.8
z=p,u=t	0	0	0.05	0.95
z=f ,u=t	0	0	0	1

p(v|z$_1$,z$_2$)	dp	dx	xp	f
z$_1$=d,z$_2$=d	0	0	0	1
z$_1$=d,z$_2$=p	0.4	0	0	0.6
z$_1$=d,z$_2$=f	0	0	0	1
z$_1$=p,z$_2$=p	0	0	0	1
z$_1$=p,z$_2$=f	0	0	0	1
z$_1$=f ,z$_2$=f	0	0	0	1

p(z|x,y)	d	p	f
x=d,y=d	0.5	0	0.5
x=d,y=p	0	0	1
x=d,y=n	0	0	1
x=p,y=d	0	0	1
x=p,y=p	0	0.5	0.5
x=p,y=n	0	0	1
x=s,y=d	0	0	1
x=s,y=p	0	0	1
x=s,y=n	0	0	1

p(x$_i$)

drop	pick	pass-by(s)
0.495	0.495	0.01

p(y$_i$)

drop	pick	noise(n)
1/3	1/3	1/3

Figure 5.10: Priors and conditional probabilities for the Bicycles problem.

all Z_l variables where $k < l$, plus it has one more connection to the unobserved terminal u. Accordingly, the number of hidden random variables at this level is,

$$\sum_{k=1}^{n} \alpha_k \beta_k \big(\sum_{j=k+1}^{n} (\alpha_j \beta_j) + 1 \big) \tag{5.1}$$

To simplify Equation 5.1 further, assume μ is the average ambiguity within all activity units where $\mu = \frac{\sum_{k=1}^{n} \alpha_k \beta_k}{n}$, from 5.1,

$$\sum_{j=k+1}^{n} \alpha_j \beta_j = \sum_{j=1}^{n} \alpha_j \beta_j - \sum_{j=1}^{k} \alpha_j \beta_j = n\mu - k\mu \tag{5.2}$$

Substituting 5.2 into 5.1,

$$\sum_{k=1}^{n} \alpha_k \beta_k (n\mu - k\mu + 1) = n^2 \mu^2 - \sum_{k=1}^{n} k \alpha_k \beta_k + n\mu \tag{5.3}$$

The second term in Equation 5.3 ($\sum_{k=1}^{n} k \alpha_k \beta_k$) cannot be simplified further using μ. The summation would be higher if the ambiguity is higher at earlier activity units. One can though find the lower bound of Equation 5.3, where $max(\alpha_k \beta_k) = \mu$ to be $n^2 \mu^2$. This reveals exponential complexity in the number of hidden Random Variables in the Bayesian network of the *Bicycles* problem.

The posterior probability can be retrieved from the BN, and rewritten, according to

Equation 3.8, to be independent of false links.

$$p(\omega|Y) = \frac{1}{\mathcal{Q}} \prod_i p(x_i|o_{x_i}) \prod_j p(y_j|o_{y_j}) \prod_{k:z_k \neq f} \frac{p(z_k|o_{z_k},x,y)}{p(z_k = f|o_{z_k},x,y)} \prod_{l:v_l \neq f} \frac{p(v_l|o_{v_l},z_1,z_2)}{p(v_l = f|o_{v_l},z_1,z_2)} \prod_n p(c_n|pa_{c_n}) \tag{5.4}$$

In Equation 5.4, pa_{c_n} is the set of parent nodes for the deterministic random variable c_n, and represents a set of interdependent events.

5.3 Feature selection and supervised training

Eight conditional probability density functions (pdf) have been listed in the last column of the attributes in the AMG (Section 5.2). These specify the likelihood of attribute values given an occurring event. Recall that the attributes are calculated from features of the detections. This section explains how these features are obtained and their ability to distinguish events of different types at the various hierarchical levels of the explanation. A training set was manually labeled to generate parameterised likelihood distributions.

It should be noted here that the framework is totally independent from the choice of the features. Other features can be specified in the AMG and used instead. Also, multiple features can be added. When multiple features are used to distinguish the same event, the likelihoods are just multiplied in the posterior calculations, assuming independence. The remainder of this section explains the selected features.

5.3.1 $p(x.sizeRatio|x.action)$

This conditional pdf uses the change in blob size across the person's trajectory to distinguish people dropping a bicycle, picking one up or passing by. Finding a visual feature that is able to distinguish this from the person's detection only was not easy. Attempts to use common pedestrian recognition techniques [33] failed to distinguish between pedestrians and cyclists. Other simple features like speed could not be used either, as cyclists slow down or even drag their bicycle as they approach the rack area.

The attribute *sizeRatio* describes the change in the foreground blob size before entering the rack area and after exiting it. A significant change in the blob size usually occurs for a person involved in a drop or a pick event. Figure 5.11 shows three graphs where the blob size before the rack area and after it are plotted, with a break that indicates the time spent within the racks. The blob size within the rack area has been ignored due to two reasons. The first is that the person bends to perform the locking or unlocking actions which results in smaller blobs. Secondly, as the person pauses to perform the action, the adaptive background tracking procedure dissolves the person's pixels into the background. These pixels are later retrieved into the foreground when the person moves again. This makes the blob size within the rack area ambiguous and noise-prone.

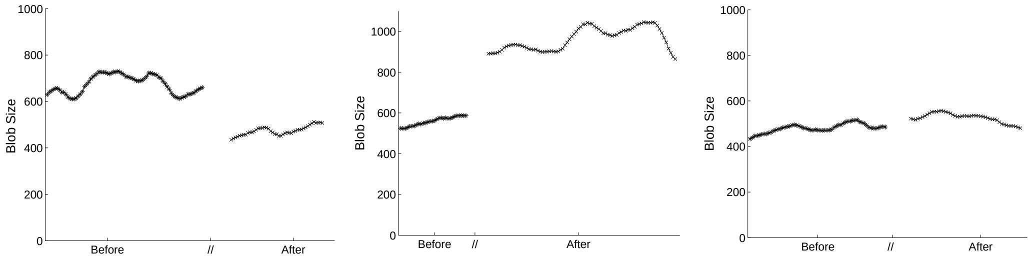


Figure 5.11: Three examples of blob size changes through time, representing a drop (left), pick (middle) and pass by (right). The three examples are selected from the training sequence (1^{st} sequence). The blob sizes have been smoothed (window size = 10).

The change in blob size can be used in any drop/pick scenario where the possessed object is comparable to the individual's projected area, as in the case of bicycles. For each person, the blob sizes from the first appearance up to entering the defined rack area are calculated and smoothed. A fixed smoothing window of size 10 was chosen throughout all the experiments. The same is accomplished for the frames between the exit from the rack area and the last appearance. The ratio of the mean blob size before and after the racks is used to assign a probability to the three possible event types: dropping, picking and passing by.

A training set is obtained where people are categorised, according to the ground truth, into the three event types. Maximum likelihood estimation (MLE) is used to estimate Gaussian class conditional densities [5]. Figure 5.12 (left) shows the three Gaussians trained for the Leeds dataset (Section 5.5.1) obtained from the ground truth of the first sequence. As expected, the Gaussian for $x.sizeRatio$ of dropping people has a mean higher than one, because the blob size before entering the racks combines that of the pedestrian and the dragged/drove bicycle, while the blob size after departing the racks represents the pedestrian only. The picking person tends to have a mean less than one, while a passing by person has a mean as close to 1 as possible. For the Cambridge dataset (Section 5.5.2), different training was required due to the difference in depth between the two entrance/exit spots for the rack area. Figure 5.12 (right) shows MLE estimates based on data from one hour of training. The situation is clearly more ambiguous. Training a mixture of Gaussians based on the different entrance and exit locations would have resulted in a more discriminative feature in this case.

[5] Appendix B explains the usage of Z-score to calculate the area under Gaussians for constrained domains as $x.sizeRatio \geq 0$.

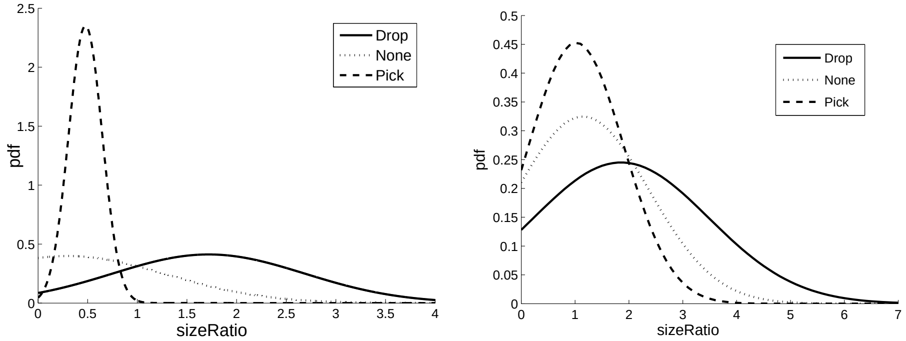


Figure 5.12: MLE for sizeRatio. The trained Gaussians are presented for the Leeds dataset (left) and the Cambridge dataset (right).

5.3.2 $p(y.edgeRatio|y.action)$

For each bicycle-cluster detection, this feature compares the intensity edges in the ‘before’ and ‘after’ reference images. Edges are retrieved by the Sobel detector [132] and are masked by the changed pixels, then removed and new intensity edges can be identified. A removed edge is one that appears only in the ‘before’ reference image, while a new edge is introduced in the ‘after’ reference image. By assuming the background is relatively free of edge features, the ratio of new to removed edges gives an estimate of whether the cluster included dropped or picked bicycles. Figures 5.13 and 5.14 show examples of removed and new edges. By plotting the removed versus new edges for the training set, two thresholds were defined that split the space into three regions: dropped, picked and noisy or multiple bicycle-clusters (Figure 5.15). For the first two regions, the cluster is classified into drop and pick respectively. Alternatively, the bicycle-cluster detection is duplicated so one can represent a dropped cluster and the other can represent a picked cluster. The two thresholds are the lines with slopes 0.5 and 2.0.

The ratio of new to removed edges was not probabilistically modelled due to the effect of the viewpoint on how many edge pixels are introduced/removed. Figure 5.6 showed how bicycles can be added in different ways, which affects the number of new or removed edges. A higher ratio of new to removed edges does not indicate higher confidence in the event’s occurrence. Training a single Gaussian would, mistakenly, favor the bicycles parked in common ways. In the experiments, all bicycle-clusters with a significant *edgeRatio* (above the threshold), are equally treated as clusters of dropped bicycles.

5.3.3 $p(Z.dist|Z.action)$

Given the temporal constraint, a person can only drop/pick a bicycle to/from a cluster detected within the same activity unit. Yet, multiple bicycle-clusters can actually be de-

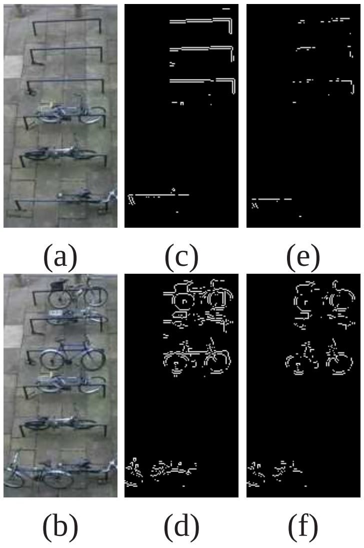


Figure 5.13: Two reference images are compared, before (a) and after (b) the activity unit. Edges, masked by the changed pixels between the before (c) and after (d) images, are compared to decide on the removed (e) and new edges (f). Notice that dropping a bicycle results in concealing edges in the background.

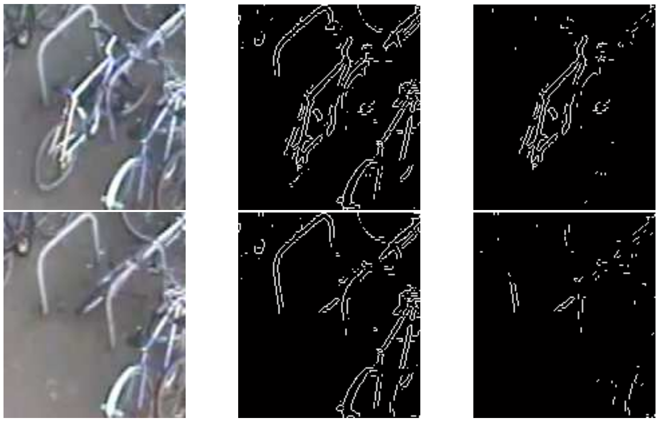


Figure 5.14: Another example from the Cambridge dataset. The first column shows the before and after reference images, the middle column shows the Sobel edges, while the removed and new edges are shown in the right-most column.

tected within a single activity unit. The feature *dist* is used to assess the probability of linking the person to a bicycle-cluster, on the assumption that the person comes close to the cluster when interacting with it. The plausibility of a link between the person and a bicycle-cluster is calculated from the maximum degree of overlap between the bounding box of the cluster and the bounding boxes of the foreground regions representing the person across the whole trajectory. For a person x and a bicycle-cluster y, where $x.traj$ represents the bounding boxes of the foreground regions across $x.n$ frames and $y.pos$ represents the bounding box of the detected cluster, then the maximum overlap is calculated

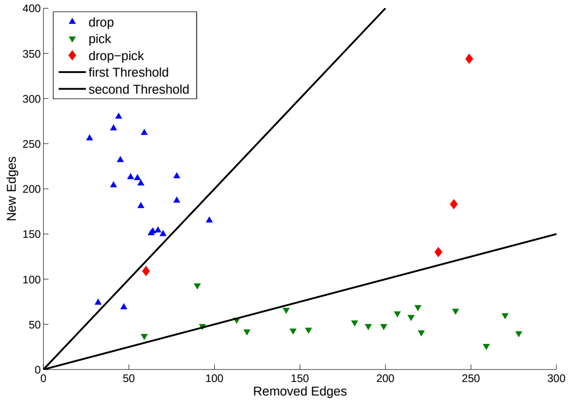


Figure 5.15: Removed versus new edges are plotted for manually labeled bicycle-clusters. All clusters with ratio < 0.5 *are clusters of picked bicycles, while those with ratio* > 2 *are clusters of dropped bicycles. The ambiguous area contains heterogeneous clusters.*

using function $\psi_{dist}(x.traj, y.pos) \in [0,1]$ as defined in Equation 5.5.

$$\psi_{dist}(x.traj, y.pos) = \max_{i\in\{1...x.n\}} \left(\frac{A(x.traj(i)\cap y.pos)}{\min(A(x.traj(i)), A(y.pos))}\right) \tag{5.5}$$

In Equation 5.5, A($\cdot$) gives the area of the given rectangle, $x.traj(i)$ is the bounding box at frame i, and $\cap$ is the (rectangular) intersection between the two bounding boxes.

A training set was created by computing the overlap for all correct and incorrect distances between people and bicycle-clusters (within the same activity unit) in the dataset. Figure 5.16 shows the histograms created from this training set, and the estimated cpdf composed of half Gaussians. The centre of the full Gaussians is fixed at 0 and 1 for incorrect and correct half Gaussians respectively.

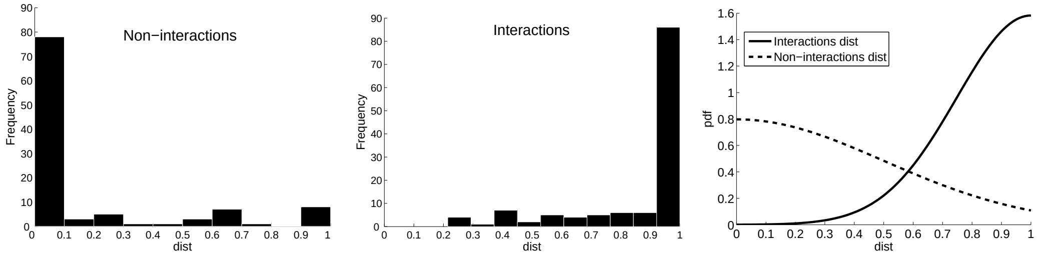


Figure 5.16: Non-interaction (left) and interaction (middle) distance histograms show how this feature can assess the likelihood of the event involving the person and the bicycle-cluster. The cpdfs (right) are trained using half Gaussians.

This feature is though not ideally informative since the person can pass close to several clusters before performing the event. This has clearly been noticed in the Cambridge dataset (Section 5.5.2).

5.3.4 $p(V.clustOverlap|V.action)$

A feature is needed to connect the drop event to its subsequent pick. Each drop of a bicycle needs to be connected to the pick of that bicycle regardless of the person performing the event. A measure is thus needed to compare bicycle-clusters. This match function accommodates any object type, and assumes objects do not change their shape or position between being dropped and picked. For two bicycle-clusters y_1 and y_2, the overlap is measured as

$$\psi_{co}(y_1.fMap, y_2.fMap) = \frac{M(y_1.fMap \& y_2.fMap)}{\min(M(y_1.fMap), M(y_2.fMap))} \tag{5.6}$$

In Equation 5.6, the function M(·) returns the number of non-zero pixels in the binary image, and the operator & is the 'Boolean and' of two images resulting in overlapping pixels between the two bicycle-clusters.

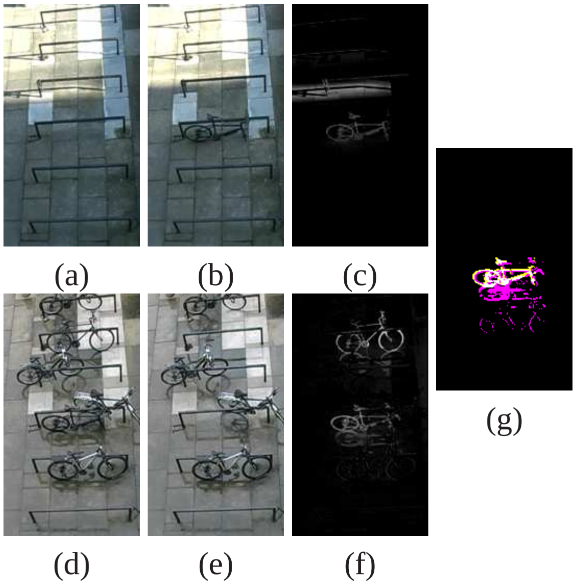


Figure 5.17: Two consecutive reference images (a) and (b) are compared to reveal changes (c) representing a dropped bicycle, and a noise cluster. Later, two consecutive reference images (d) and (e) are also compared to reveal two picked bicycles (f). By comparing the changed blobs (g), the clusters overlap gives a high likelihood and a pixel match of 0.86 (calculated using Equation 5.6). Yellow pixels represent the dropped clusters while pink pixels represent the picked cluster. White pixels signify the overlapped pixels.

Figure 5.17 shows an example of a drop and a pick that were correctly connected by comparing the changed blobs despite the temporal gap between the two events. Figure 5.18 shows another example from a more challenging dataset. Figure 5.19 presents the trained Gaussians.

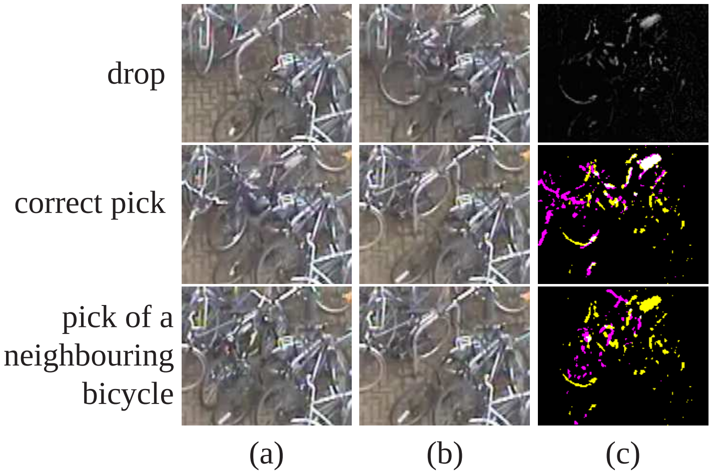


Figure 5.18: A harder example of bicycle-clusters pixels overlap. 'Before' (a) and 'after' (b) reference images are compared for three activity units. Pixel-to-pixel matches (c) are capable of detecting the correct pick with higher pixel overlap. This example is from the Cambridge dataset where clutter and ambiguity are significantly high.

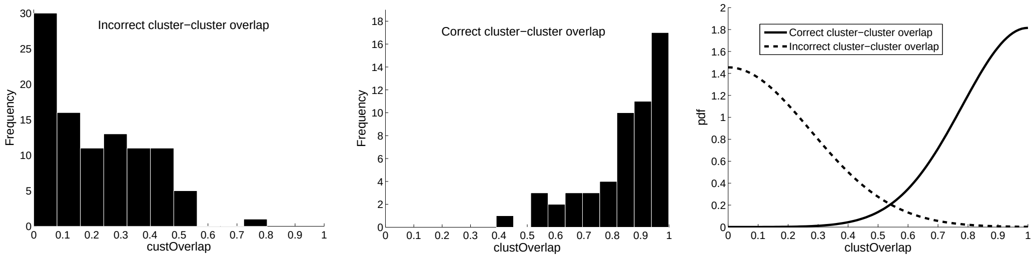


Figure 5.19: Training labelled data is fitted into half-Gaussians with a fixed mean of 1 for correct values and a fixed mean of 0 for incorrect connections.

5.3.5 Post-segmentation

Assume several people dropped bicycles within the same bicycle-cluster. The distance between a person and the bicycle-cluster will produce a high likelihood between all these people and the combined cluster. When one bicycle amongst the cluster is later picked, the pixel overlap helps refining the bounding box estimate of the bicycle object. This can be clarified by an example shown in Figure 5.20.

Notice that this assists segmenting the bicycle from a bicycle-cluster composed of several bicycles for both the drop and the pick events. This is referred to as 'post-segmentation' because the bicycle is segmented after the drop-pick link is established. The post-segmented position $V.pos$ is the intersection between the dropped and the picked clusters. After this position is determined, the distance and the ratio of edges can be revisited. The maximum overlap between the person's trajectory $traj$ and the post-segmented position $V.pos$ is calculated using the function ψ_{dist} for both the dropping and picking

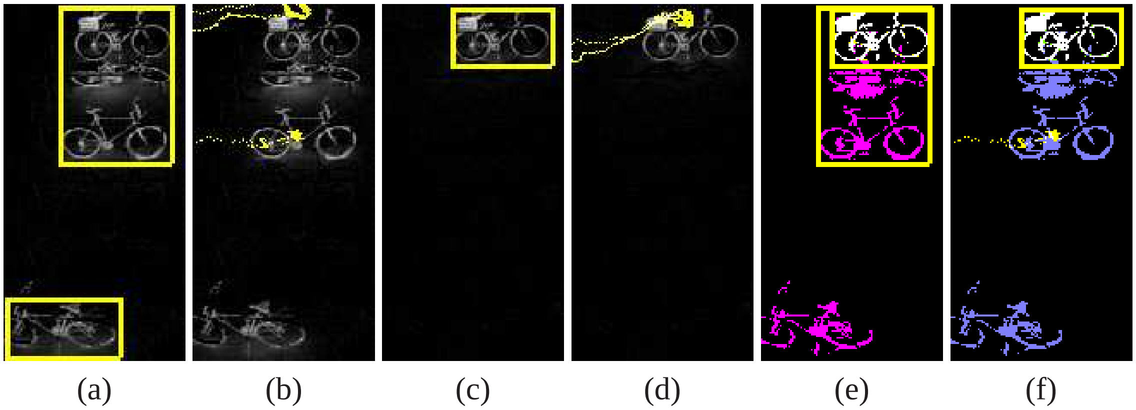


Figure 5.20: Two bicycle-clusters are identified during one activity unit (a). The top cluster combines three bicycles that cannot be individually segmented. Two people's trajectories ($x_1(top), x_2(bottom)$) are displayed (b) and are linked to the cluster. During a later activity unit, one bicycle was picked (c) by a person (d). Matching pixels of the dropped and picked clusters enables segmenting the bicycle and provides a better estimate of its location and bounding box (e). Only one of the people shown in (b) can now be linked to this refined boundary, due to the post-segmentation information. (f) shows the person x_2 cannot be part of this drop-pick event.

trajectories, and is referred to as $psDropDist$ and $psPickDist$. Similarly $\psi_{edgeRatio}$ is calculated, so the new and removed edges are limited within the new boundary. The latter is efficiently performed using integral histograms [4]. Post-segmentation is incorporated into the grammar as synthetic attributes.

This section has reviewed all the likelihoods required for the *Bicycles* problem BN. Figure 5.21 labels the example BN with the likelihood functions for completeness.

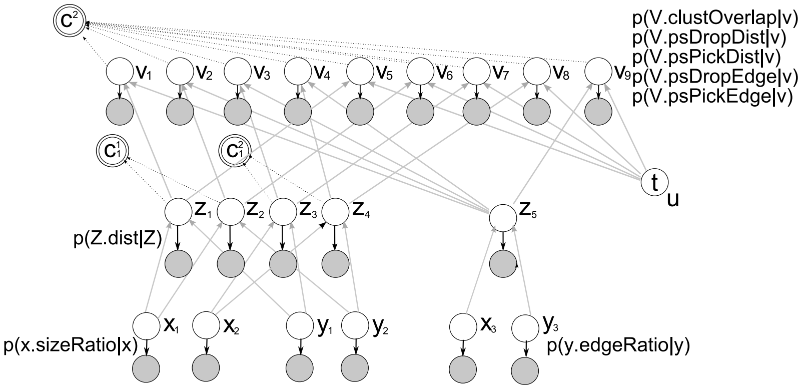


Figure 5.21: The different likelihoods/features shown on the BN structure.

5.4 Reversible moves for the *Bicycles* problem

After the BN is built, the MAP solution is sought using the techniques from Chapter 4. This section explains how the general RJMCMC moves can be applied to the *Bicycles* problem. The four move types introduced in Section 4.4.2 are duplicated for the two layers of binary event linkage (Figure 5.22). The subscript for the move type indicates the layer. In the initial explanation ω_0, all people are passing by the rack area and all bicycle-clusters are noise. This is a valid explanation, though unlikely to result in the MAP solution. At each step of the Markov chain, a move is applied to the current explanation. Figure 5.23 shows a sequence of moves. For each move, the reversible move is shown to indicate the chain can run both ways. Each applied move creates a new explanation ω', and can change multiple labels in the Bayesian network. Moves of type change$_v$, for example, change the labels of four hidden RVs of type V (Figure 5.24).

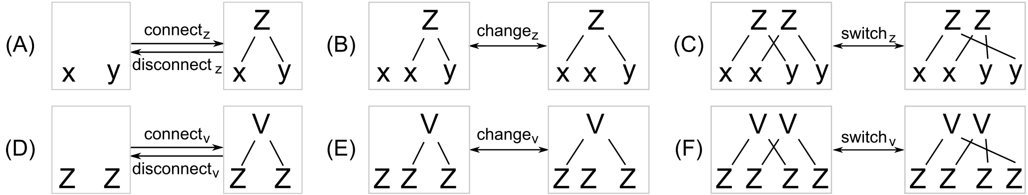


Figure 5.22: Generalised reversible moves, for both layers of the Bicycles problem.

Next, one needs to define the proposal distribution for the Markov chain $Q(\omega'|\omega)$. RJMCMC uses two proposal distributions to propose a new explanation: one for choosing the move type j_m, and another for choosing a specific move g_m. Randomly choosing a move type does not efficiently search the space of explanations. Section 4.4.2 suggested estimating the number of distinct moves of each type that can be applied to the current explanation. For example, the number of possible 'disconnect$_z$' moves equals the total number of drop and pick events in the current explanation. These counts are used as weights in choosing the move type.

Next, a specific move of that type is chosen and applied to the current explanation. This 'within-move' choice can also be performed uniformly at random. Alternatively,

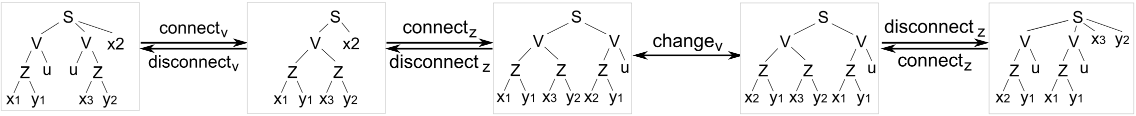


Figure 5.23: A sequence of $\{$connect$_v$ $\rightarrow$ connect$_z$ $\rightarrow$ change$_v$ $\rightarrow$ disconnect$_z\}$ moves was applied. The last move affects both layers as disconnecting a pick cancels the drop-pick linked to that pick. The subscript next to the move type indicates the compound event for which the move is applied.

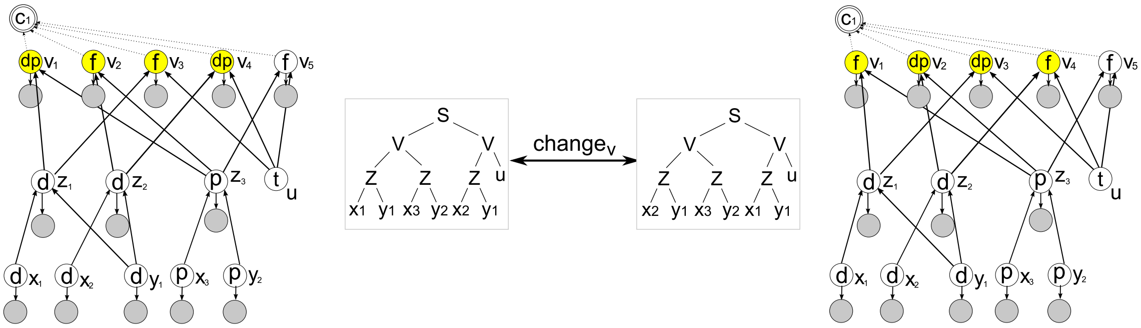


Figure 5.24: The effect of the third move in the sequence is shown on the BN. Shadowed hidden RVs have been affected by the move. This move changes the labels of four hidden RVs in the BN.

one can design a customised proposal distribution for each move type. Section 4.4.2 explained that these proposal distributions are application-dependent. This section lists a measurement δ_t for each move type t that weights the preference for choosing moves of that type. The proposal distribution g_m is then a weighted distribution from which a move is selected at random. For example, the 'connect$_z$' move type prefers connecting people to bicycle-clusters without existing links. The chosen weights for all move types are described next. In the coming equations, $B(x_i)$ yields the set of clusters that could be connected to person x_i, while $T(y_j)$ yields the set of people that could be connected to cluster y_j.

Move type (A) connect$_z$/disconnect$_z$

The ambiguity related to each person is calculated from the number of bicycle-clusters to which the person can be connected, and the ambiguity related to each of these clusters. For person x_i, the measurement for weighting moves of type *connect$_z$* is defined in Equation 5.7.

$$\delta_{connect_z}(x_i) = \sum_{y_j \in B(x_i)} \frac{1}{|T(y_j)|} \tag{5.7}$$

The measurement for the *disconnect$_z$* move type is the inverse of that for *connect$_z$*.

$$\delta_{disconnect_z}(x_1) = \frac{1}{\sum\limits_{y_j \in B(x_i)} \frac{1}{|T(y_j)|}} \tag{5.8}$$

Recall that $\delta_{connect_z}$ is defined for all passing-by people, while $\delta_{disconnect_z}$ is defined for all dropping and picking people.

Move type (B) change$_z$

This move type is defined for all Z events, and is self-reversible. For each connected person and bicycle-cluster, δ_{change_z} tests whether the cluster is better connected to another person, or the person is better connected to another cluster.

$$\delta_{change_z}(x_i, y_j) = \sum_{x_k \in T(y_j) - \{x_i\}} \frac{\psi_{dist}(x_k.traj, y_j.pos)}{\psi_{dist}(x_i.traj, y_j.pos)} + \sum_{y_k \in B(x_i) - \{y_j\}} \frac{\psi_{dist}(x_i.traj, y_k.pos)}{\psi_{dist}(x_i.traj, y_j.pos)} \tag{5.9}$$

δ_{change_z} sums the relative weight of all alternative links. A relative weight of 1 is given to all equally-likely connections, < 1 for less-likely connections, and > 1 for more-likely connections.

Move type (C) switch$_z$

$$\delta_{switch_z}(x_i, y_j, x_k, y_l) = \frac{\psi_{dist}(x_i, y_l)\,\psi_{dist}(x_k, y_j)}{\psi_{dist}(x_i, y_j)\,\psi_{dist}(x_k, y_l)} \tag{5.10}$$

δ_{switch_z} weights switching two drop/pick events based on the ratio of the new connection likelihoods to the current connection likelihoods. Notice that this weighting does not take into consideration the changes that can be introduced to any related drop-pick events. This will be evaluated when the move is actually applied. For example δ_{switch_z} can be greater than 1 for a specific move, yet it would result in a lower posterior. Refer to 4.4.2 for the explanation of how RJMCMC formulations can preserve the detailed balance condition, and thus convergence.

Move type (D) connect$_v$/disconnect$_v$

Proposing to connect an unconnected drop to a later pick is weighted by the bicycle-clusters overlap

$$\delta_{connect_v}(Z_i, Z_j) = \psi_{co}(Z_i.fMap, Z_j.fMap) \tag{5.11}$$

The disconnect move measurement is the inverse

$$\delta_{disconnect_v}(Z_i, Z_j) = \frac{1}{\psi_{co}(Z_i.fMap, Z_j.fMap)} \tag{5.12}$$

Notice that while $\delta_{connect_z}$ calculates the number of ambiguous alternative connections, this does not suit the *connect*$_v$ move type. Introducing a similar measure would favour connecting older drops, which cannot be justified.

Move type (E) change$_v$

$$\delta_{change_v}(Z_i, Z_j) = \frac{\max\big(\max\limits_{\{k:Z_k=d\}-\{i\}} \psi_{co}(Z_k, Z_j), \max\limits_{\{k:Z_k=p\}-\{j\}} \psi_{co}(Z_i, Z_k)\big)}{\psi_{co}(Z_i, Z_j)} \tag{5.13}$$

δ_{change_v} gives a weight of 0 when there are no alternative drops or picks, < 1 if the current connection has the highest likelihood, and > 1 if a better connection is available.

Move type (F) switch$_v$

For two pairs of drop-picks (Z_i, Z_j) and (Z_k, Z_l), the measurement for switching is dependent on the ratio of the new cluster-overlap likelihoods to the old likelihoods.

$$\delta_{switch_v}(Z_i, Z_j, Z_k, Z_l) = \frac{\psi_{co}(Z_i, Z_l)\psi_{co}(Z_k, Z_j)}{\psi_{co}(Z_i, Z_j)\psi_{co}(Z_k, Z_l)} \tag{5.14}$$

In addition to the within-move proposal distributions, it should be mentioned that, when a move is applied, $\frac{\pi(\omega')}{\pi(\omega)}$ can be simplified based on knowing the move type. Thus, the full posterior need not be evaluated at each step of the Markov chain. For example, for the connect$_z$ move type, where person x_i drops a bicycle into the bicycle-cluster y_i that was initially a noise cluster, the similar terms in $\pi(\omega)$ and $\pi(\omega')$ cancel each other resulting in the ratio

$$\frac{\pi(\omega')}{\pi(\omega)} = \frac{p(x_i = d|o_{x_i})p(y_j = d|o_{y_j})p(z_{ij} = d|x_i, y_j, o_{z_{ij}})}{p(x_i = f|o_{x_i})p(y_j = f|o_{y_j})p(z_{ij} = f|x_i, y_j, o_{z_{ij}})} \tag{5.15}$$

Only these 6 terms are evaluated when applying a move of this type. The remaining simplified ratios for all the move types are not listed here to avoid redundancy.

5.5 Datasets

Two locations have been chosen for recording. The first is within the University of Leeds. It is referred to as the ‘Leeds’ dataset, and consists of 37 hours of recording. Another dataset was obtained from National Express. This was recorded outside Cambridge train station. It is referred to as the ‘Cambridge’ dataset, and consists of 30 hours of recorded video. Table 5.5 contains a summary of statistics for both datasets.

	sequences						
	Leeds					Cambridge	
	1	2	3	4	5	6	7
Duration	1h	1h	11h	12h	12h	15h	15h
Activity Units	35	17	19	118	96	87	132
$\|\{x\}\|$	58	27	128	126	137	112	197
$\|\{y\}\|$	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
Simulated Thefts	7	0	7	1	0	0	0

Table 5.5: Dataset statistics; $|\{x\}|$: number of detected people, $|\{y\}|$: number of detected bicycle-clusters.

5.5.1 Leeds dataset

A rack area containing 6 racks was chosen for recording. The camera was mounted in a third floor window to capture the full rack area and a leading area showing people approaching and departing. Two 1 hour sequences were recorded during busy periods (1-2). Three full days (8am to 7/8pm) were also recorded to test long duration (3-5). Table 5.5 details the number of events of each type in the ground truth of these five sequences. It also lists the number of detected people and bicycle-clusters.

This dataset provided a thorough test, and was recorded on separate days between Oct 2006 and May 2008. It proves the ability of the prototype to work under severe weather conditions (rain, hail, shadowed and sunny periods are all part of the dataset). No recording was done at night as the tracker fails in dim lighting. All sequences were recorded in a 360×288 screen size at full frame rate (25fps). This enabled a real-time performance of the tracker. The location of the rack area was manually selected, as shown in Figure 5.25.

For this dataset, the participants were regular staff and students that would use the rack, as well as actors to simulate extra complexity like people returning with different clothing or simulated thefts. As indicated in Table 5.5, some simulated thefts were recorded to ensure the system succeeds in linking drops and picks when different individuals perform the events. This is also used to assess the ability of the prototype to detect thefts as will be explained in Section 5.6.3.

5.5.2 Cambridge dataset

Figure 5.26 shows the viewpoint from the Cambridge dataset along with the manually defined rack area. The provided videos were recorded from 6am to 9pm on the 17^{th} and

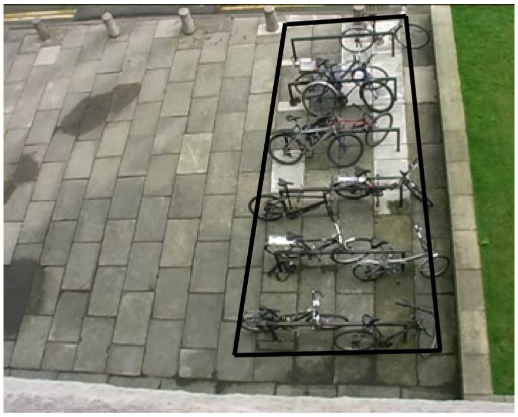


Figure 5.25: Viewpoint of the Leeds dataset. A manually defined convex polygon delimits the rack area.

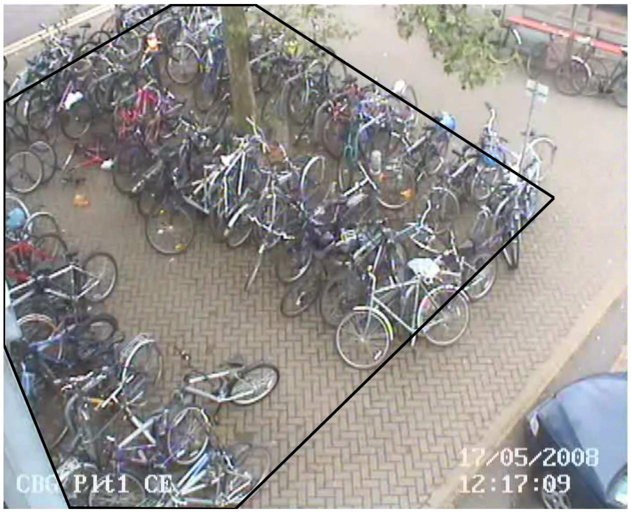


Figure 5.26: The Viewpoint of the Cambridge dataset.

21st of May 2008. A Pan-Tilt-Zoom (PTZ) camera, outside Cambridge train station, was fixed for collecting this dataset. The resolution of 704×576 was retrieved from the source at full frame rate (25 fps). After receiving this dataset, it was noticed that many bicycles were kept in the racks for long durations. The number of drop and pick events is thus less than anticipated when viewing the cluttered bicycles in Figure 5.26. This dataset differs from the one recorded in Leeds in the following aspects:

- The rack area occupies most of the viewpoint, leaving little space for the leading area. This affected the ability to observe the change in the blob size before approaching the area and after departing (Figure 5.12).
- The recording quality is lower, introducing more noise and aliasing effects.
- No actors were involved, and no thefts were recorded.

- The racks are not fully visible. Some racks are hidden behind the tree and others are not within the camera's viewpoint. People passing with the bicycle in front of the camera, but parking the bicycle outside the field of view (or behind the tree), were labeled as passing-by individuals in the ground truth.
- Due to the cluttered bicycle racks, a higher number of bicycles were shifted from their position while another bicycle is being dropped or picked. This increased the size of detected bicycle-clusters.
- Due to the over-cluttered racks as well, a considerable number of individuals tried to squeeze their bicycles in, and then departed without leaving a bicycle behind. Such events result in a change within the rack area, through the attempts of squeezing the bicycle. As the solution detects changes within the rack as bicycle-clusters, and associates these with people using spatial proximity, this resulted in a decrease in the explanation's accuracy.

Despite the challenge of the Cambridge dataset, the prototype was used without change following development on the Leeds dataset. The duration and number of events in this dataset has been presented in Table 5.5 under the 6^{th} and 7^{th} columns. To simplify the results, the Leeds dataset sequences are numbered 1 to 5, while 6 and 7 denote the two days of the Cambridge dataset.

5.6 Results

This section shows the results of searching for the MAP solution, which corresponds to the best explanation, using different search methods for the dataset sequences. Upon achieving the maximum a posteriori explanation $\hat{\omega}$, the accuracy is calculated by comparison to the ground truth. Finally, this section discusses an application of this activity recognition task to bicycle theft detection. Although this application requires further research related to passive biometrics and risk management, the global explanation forms the basis for its solution.

5.6.1 MAP explanation results

The search is for the global explanation $\hat{\omega}$ that maximises the posterior probability. Thus, comparing two search algorithms is based on comparing the posterior probabilities of the explanations found by the algorithms. This is done independently of the accuracy attained, which is considered in Section 5.6.2. In what follows, the negative log is minimised rather than the probability maximised. This is because small numbers cause overflowing.

The MAP is compared across all sequences for greedy, MHT, RJMCMC and IP searches. The greedy search is performed as explained in Section 4.2. MHT is compared for k = 50, 100 and 500 branches as explained in Section 4.3. Ten parallel chains (n_{mc} = 5000) are run for each of RJMCMC and RJMCMC-SA recording the MAP amongst all chains. These are run starting at the greedy solution and offline. Linear cooling is used for SA. The length of the Markov chain and linear cooling were chosen based on experiments on the training sequence (1^{st} sequence). It was noticed that the chains converge after 3000 steps or so, n_{mc} was accordingly chosen to be 5000. Similar performance was recorded for linear and exponential cooling, while sigmoid cooling performed slightly worse. The RJMCMC search is run 40 times, recording the mean and the standard deviation. IP is run on both the MATLAB and XPRESS-MP solvers. Table 5.6 shows the complete MAP results for all the dataset sequences. In all cases, the greedy search could not find the MAP explanation. RJMCMC achieved better results than MHT in four out of the seven sequences, and comparable results in the remaining three sequences. RJMCMC-SA achieved the best results amongst heuristic methods. Integer Programming shows the MAP solution by exhaustive searching.

	Greedy	**MHT**			**RJMCMC**		**RJMCMC-SA**		**IP**	
		k=50	**k=100**	**k=500**	μ	σ	μ	σ	**MATLAB**	**XPRESS-MP**
1	102.25	58.78	58.78	57.86	57.90	0.11	57.86	0.00	57.86	57.86
2	23.54	4.64	4.64	4.64	4.64	0.00	4.64	0.00	4.64	4.64
3	609.66	493.18	468.80	468.80	429.30	3.23	423.98	2.36	416.64	416.64
4	6272.69	6149.95	6144.98	6144.30	6079.88	3.43	6078.40	3.23	6065.0	6065.00
5	5034.46	4998.39	4982.86	4975.82	4943.71	3.59	4939.33	1.87	4937.1	4937.08
6	860.37	812.96	812.96	812.96	814.71	1.69	811.50	2.36	797.29	797.29
7	934.36	608.92	607.39	-	451.92	9.29	433.50	7.76	-	283.51

Table 5.6: $-\log(p)$ *compared across greedy, MHT (k = 50, 100, 500), 40 runs of RJMCMC, 40 runs of RJMCMC-SA and Integer Programming. The results are not available for MHT (k=500) or MATLAB linear solver on sequence 7 due to the implementation running out of memory.*

The comparison is also presented visually in Figure 5.27. The MAP (presented as $-\log(p)$) is compared across the sequences, where the posterior found using MHT (k=50) is vertically aligned for all sequences. For RJMCMC and RJMCMC-SA bars, the height of the bar represents mean of the different runs, and a vertical line presents the standard deviation .

To visualize the different explanations during a Markov chain, Figure 5.28 demonstrates a diagram for the explanation every 250 steps in the Markov chain. These diagrams are for one run of RJMCMC-SA on the 3^{rd} sequence. Starting from passing-by events for

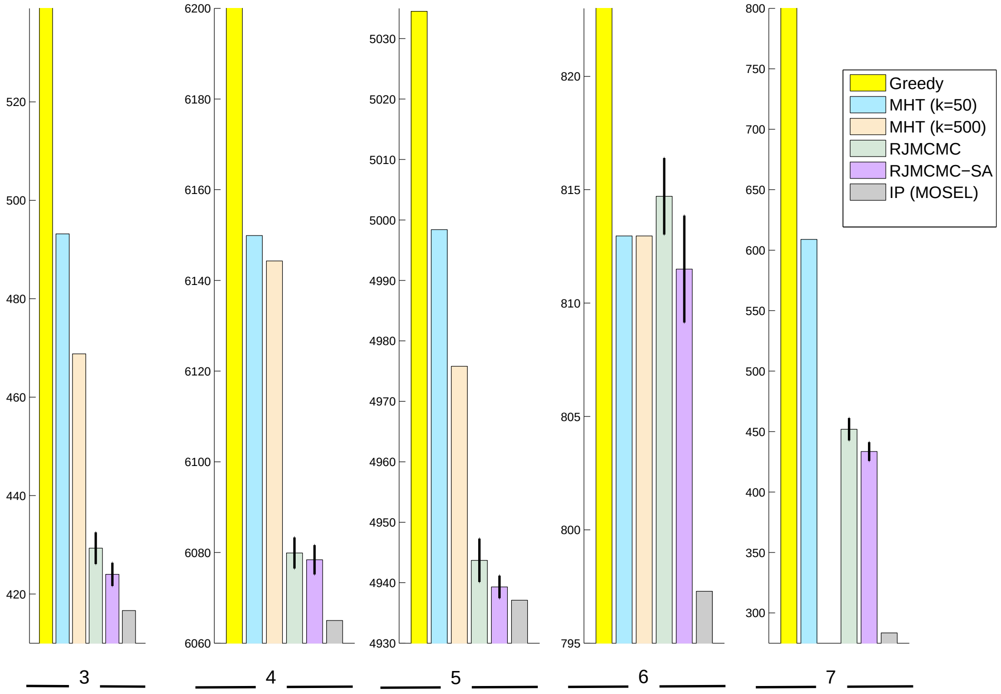


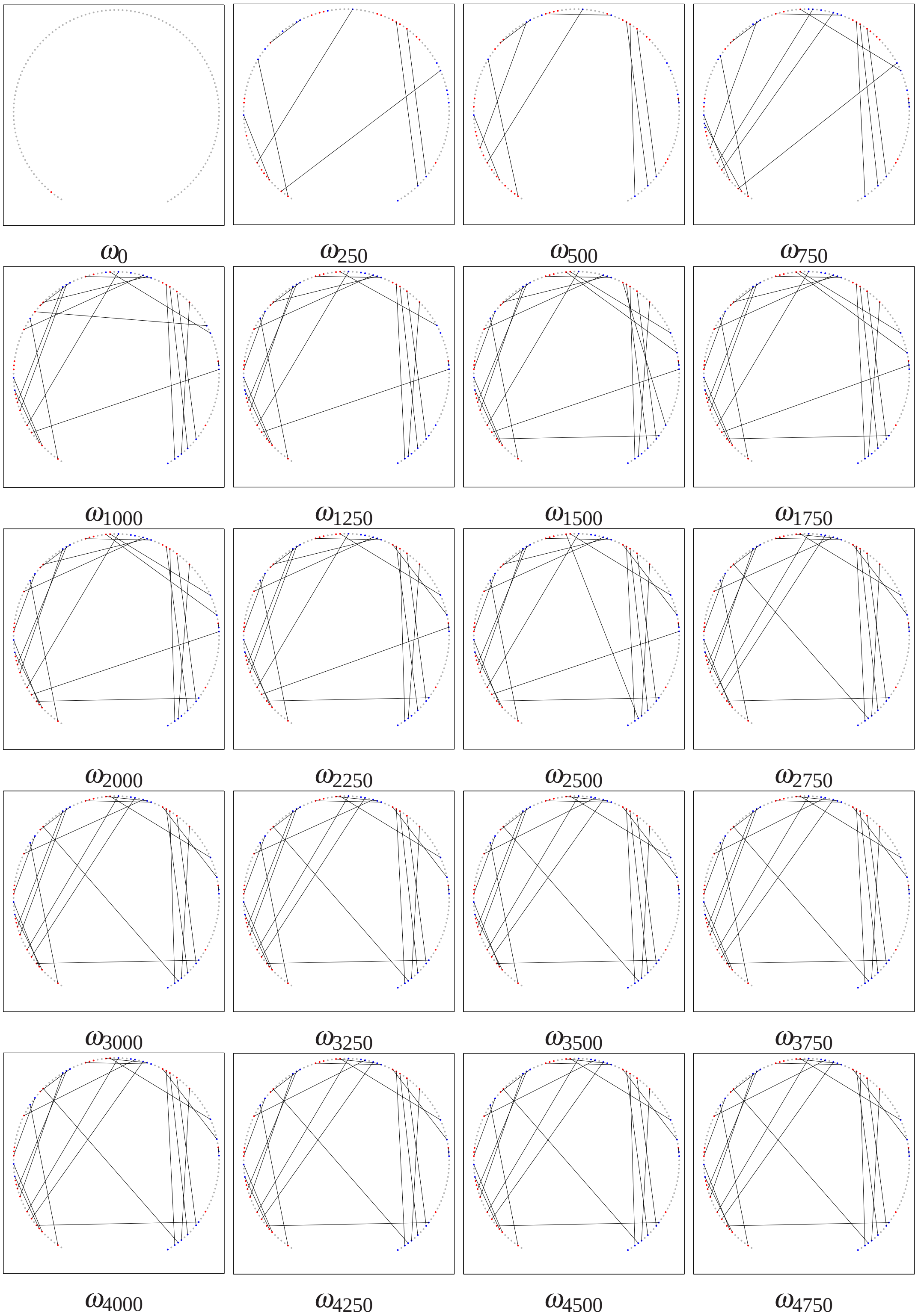
Figure 5.27: MAP is compared for the full day sequences (3-7) showing RJMCMC-SA achieves the best heuristic search results. The vertical line represents the standard deviation σ for RJMCMC and RJMCMC-SA.

all people detections, drop and pick events are recognised and linked (represented by a line connecting the pair of events). The visualization shows convergence to the best explanation at the end of the sequence. The diagram shows the complexity of the solution and the interleaved unordered events.

When comparing the time required for each of the search techniques, it is worth mentioning that each run of the RJMCMC chains executes within 3-7 minutes for the sequences in the *Bicycles* problem. This is an unoptimised code implemented using Java^{TM}, and run on a 4GB server. The time needed to run MHT depends on the number of branches k and was around 20minutes for $k = 500$. IP using the linear solvers takes between 5 and 30 minutes with the varying complexity in the code, run on a server of 10GB memory. Note that the code was not optimised for performance comparison.

For the Integer Programming results, Table 5.7 shows the number of partial explanations $\mathcal{F}$ for each of the 7 sequences.

After comparing the different search techniques, results are shown for different ways of searching using RJMCMC and RJMCMC-SA. Results are also available for the online search and starting from a completely unconnected explanation. Table 5.8 shows the com-



ω_{0} ω_{250} ω_{500} ω_{750}

ω_{1000} ω_{1250} ω_{1500} ω_{1750}

ω_{2000} ω_{2250} ω_{2500} ω_{2750}

ω_{3000} ω_{3250} ω_{3500} ω_{3750}

ω_{4000} ω_{4250} ω_{4500} ω_{4750}

Figure 5.28: A visual representation of the explanation along a Markov chain ($n_{mc} = 5000$), where dots denote person detections equally spaced between 0700H and 1700H. Drops (red dots) and picks (blue dots) are linked by a straight line to form drop-pick events.

sequence	$\mid\mathcal{F}\mid$
1	784
2	171
3	1492
4	2381
5	1303
6	1484
7	7963

Table 5.7: The number of partial explanations in the integer program for each of the 7 sequences.

plete results for RJMCMC search under different starting points, simulated annealing and online performance for the 5^{th} sequence. The complete results are shown in Appendix E.1 for the remaining sequences. The coming subsections explain several aspects regarding the different ways to run Markov chains.

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	4937.10	4941.01	4.06
	×			5,000	4943.71	4939.37	1.96
×			×	5,000	4943.71	4943.71	3.59
	×		×	5,000	4943.71	4939.33	1.87
×		×		1000/au	4927.60	4963.7	22.45
	×	×		1000/au	4956.55	4968.5	5.16
×		×	×	1000/au	4924.08	4945.8	12.60
	×	×	×	1000/au	4929.63	4956.3	16.17

Table 5.8: MAP results using different variations of the RJMCMC search for the 5^{th} sequence. Results for the other sequences are shown in Appendix E.1.

5.6.1.1 RJMCMC proposal distribution choices

To assess the effect of proposal distribution choices on the convergence of the Markov chain, this section presents results using different choices of the proposal distributions. In RJMCMC, first the move-type is to be selected j_m, then a move from the within-move proposal distribution g_m is chosen. These choices can be made uniformly at random (u.a.r.) or weighted. The choice of the move type is weighed by the estimated count of possible moves of that type, while the choice of individual moves is weighted by the designed measurements (Equations 5.7- 5.14). Figure 5.29 shows an example of convergence for both RJMCMC and RJMCMC-SA under various choices of the proposal distribution. Three choices are presented, the first choice is when both the move type and the individual move are chosen 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.

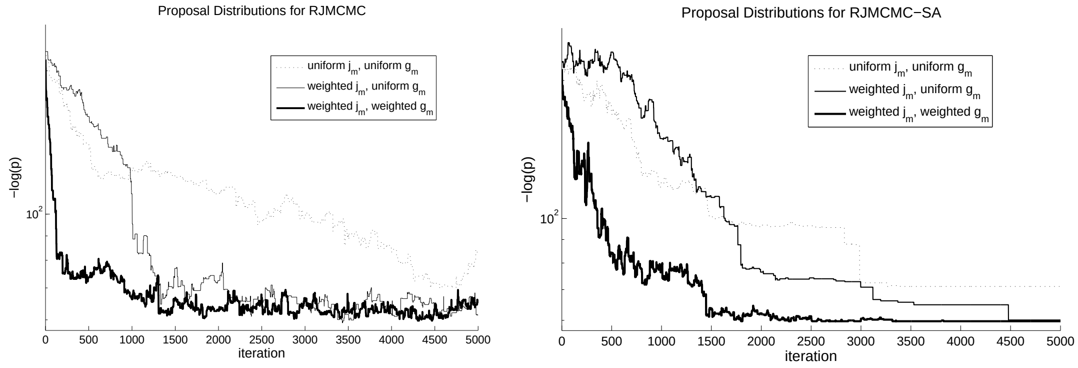


Figure 5.29: Two figures presenting convergence under various u.a.r and weighted proposal distribution $Q(\omega'|\omega)$ choices using RJMCMC (left) and RJMCMC-SA (right) for the 4th sequence.

Table 5.9 compares the results across the seven sequences for uniform and weighted g_m choices. 100 chains are run with a weighted j_m and uniform g_m proposal distributions, and another 100 chains were run with the weighted j_m and g_m distributions. The results show lower $-\log(p)$ for all the datasets when weighted g_m distributions are used. When testing for significance using the two-sample t-test, 6 out of 7 sequences (except the 2nd sequence) proved the difference is statistically significant [6].

	Uniform g_m		**Weighted** g_m	
	μ	σ	μ	σ
1	59.51	1.28	58.57	0.60
2	4.78	0.72	4.74	0.58
3	451.27	10.03	437.19	5.37
4	6165.87	17.21	6130.34	19.47
5	4986.91	10.47	4950.89	6.27
6	862.30	6.09	819.20	3.73
7	486.95	9.41	469.90	12.12

Table 5.9: $-\log(p)$ compared for all the sequences, with 100 chains with a uniform g_m, and 100 chains with weighted g_m.

5.6.1.2 Running multiple Markov chains

Being a Monte Carlo process, which is inherently random, it is believed that running multiple chains can result in a better chance of finding the global maximum [142]. These chains can run in parallel and are independent of each other. Figure 5.30 shows the posterior and acceptance rate for three RJMCMC chains tested on the 5th sequence. Similarly, Figure 5.31 compares three RJMCMC-SA chains.

[6] at 5% significance level

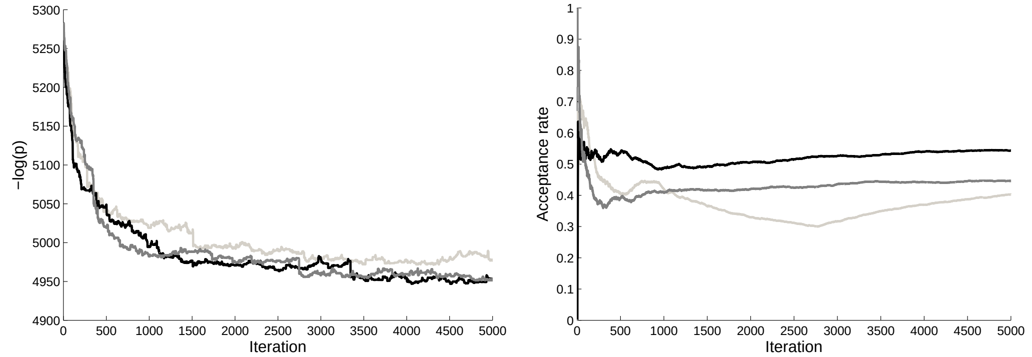


Figure 5.30: Three RJMCMC runs for the 5^{th} sequence. In this figure, the chain plotted in black finds the highest posterior. The figure to the right shows the acceptance rate ρ_{accept} for the three chains in corresponding line styles.

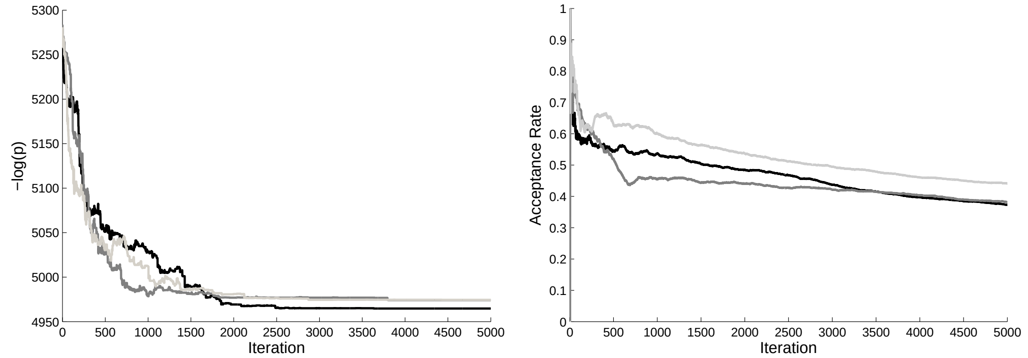


Figure 5.31: Three RJMCMC-SA runs for the 5^{th} sequence. Though all SA runs converge, they tend to converge to different peaks (local maxima) of the distribution. The chain plotted in black finds a higher posterior. ρ_{accept} results are shown to the right.

Table 5.10 compares a single long chain with multiple shorter chains containing the same total number of sample elements. The experiments are run 10 times to estimate the mean and the standard deviation [7]. The table shows a higher posterior mean for the single chain in three sequences, in comparison with a higher posterior mean for the multiple chains in two other sequences. The performance is thus comparable for multiple short chains and a single long chain. As the multiple chains are shorter and were run in parallel, the potential time it takes is significantly reduced [8]. Multiple chains were used in the results presented in Table 5.8. For all experiments shown next, n_{mc} was set to 5000 for offline search and to 1000 for each activity unit during online search.

[7] Statistical significance cannot be concluded from this the small sample size

[8] A 64-dual processor parallel cluster was used. This service was provided by the White Rose Grid.

	single chain (n_{mc}=50,000)		**multiple chains (10 $\times n_{mc}$=5000)**	
	μ	σ	μ	σ
3	424.99	5.44	428.68	4.12
4	6077.32	3.98	6080.57	3.88
5	4947.30	5.56	4941.11	3.65
6	814.77	1.72	813.83	2.24
7	429.39	15.03	433.01	8.43

Table 5.10: $-\log(p)$ compared for 10 runs of a single long chain versus 10 runs of multiple parallel shorter chains.

5.6.1.3 Adding simulated annealing

When using simulated annealing, a choice of the temperature range (initial and final temperatures) and a cooling schedule is needed. The choices were $T_0 = 4.00$ and $T_{n_{mc}} = 0.01$. Linear cooling was found suitable for the training sequence. In the remaining results, RJMCMC-SA uses linear cooling (Equation 4.7) when updating the temperature.

To test whether adding simulated annealing is a statistically significant improvement, Table 5.11 shows the results of a two-sample t-test. The test assumes the two samples are generated from Gaussian distributions. For each case, 400 independent chains were run from a local maximum [9] for each of RJMCMC and RJMCMC-SA. Linear cooling schedule was used for SA. To test that each sample is generated from a Gaussian distribution, the chi-square goodness-of-fit (χ^2gof) is tested for each sample. The χ^2gof test checks whether the sample is a random sample from a normal distribution with a mean and standard deviation estimated from the sample. Then, the tailed Welch t-test at α=0.05 is used, as it does not assume the variances of the two samples are equal. The test returns 1 if RJMCMC-SA generates statistically significant higher MAP than RJMCMC at 5% significance level. The table demonstrates the statistical significance for the third, fourth, sixth and seventh sequences. The χ^2gof test failed in the remaining three cases. Though RJMCMC was used in previous work to find the MAP solution, the experimental results here support the theoretical concept that adding SA can better search the distribution for the MAP solution.

5.6.1.4 Initialising the Markov chain

Two methods were used to initialise the Markov chain. The first method is to start from scratch with all people considered passing-by and all bicycle-clusters labeled as noise. Another way to initialise the Markov chain is to start from the explanation found by the greedy search. Choosing the second initialisation is expected to speed convergence. Nevertheless, the theory of MCMC proves its immunity to initial states. MCMC's con-

[9] These are the same as the independent chains used in Table 5.8

sequence	RJMCMC			RJMCMC-SA			Welch t-test
	μ	σ	χ^2 **gof**	μ	σ	χ^2 **gof**	
1	58.71	0.84	1	59.29	2.14	1	0
2	5.19	1.30	1	5.53	1.56	0	0
3	438.66	6.04	0	432.34	5.88	0	1
4	6092.93	9.93	0	6090.40	9.11	0	1
5	4956.36	9.17	1	4947.11	6.08	1	1
6	819.61	3.20	0	817.62	4.12	0	1
7	472.51	13.16	0	457.74	16.67	0	1

Table 5.11: Welch t-test to compare 400 runs of RJMCMC and RJMCMC-SA. In the last column, 1 indicates the right-tailed null hypothesis was rejected at 5% significance level. This means the $-\log(p)$ was higher for RJMCMC when compared with RJMCMC-SA (recall that this means lower MAP). For the χ^2gof columns, 0 indicates the sample is drawn from a normal distribution when tested with χ^2 goodness-of-fit at 5% signifcance level.

vergence to the target distribution is independent of the initial state. Figure 5.32 shows two different initialisations of the Markov chain. Initialising the chain with the solution in which all people are passing-by and all bicycle-clusters are noise takes longer to achieve convergence.

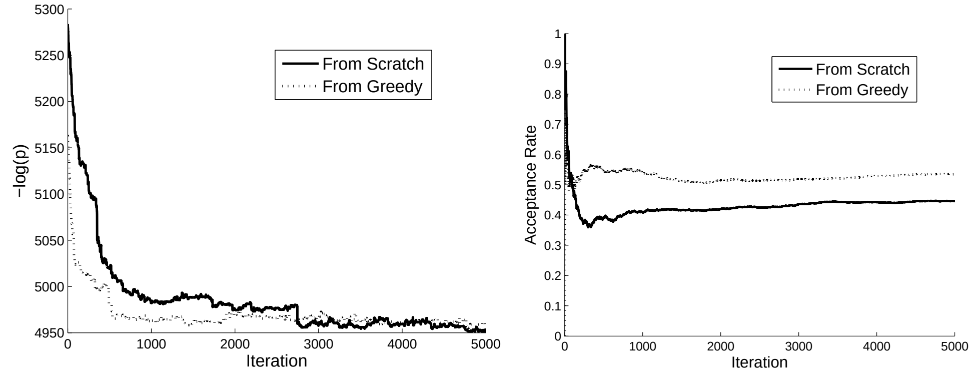


Figure 5.32: Two runs of RJMCMC from different initial explanations applied to the 5^{th} sequence.

For the complete results of the dataset, 400 chains are run from each of the two initial explanations. Table 5.12 compares the results. In 6 out of 7 sequences, the means are within 1 standard deviation (1σ) of each other. Also, in four out of the seven sequences, the difference in means of the two samples is not considered statistically significant using the two-sample t-test, i.e. they originate from the same proposal distribution.

5.6.1.5 Online optimisation

Figure 5.33 shows online RJMCMC-SA, run in two phases at the end of each activity unit, as explained in Section 4.4.6. For each chain, the best performance initialises the Markov chain for the next activity unit. Some activity units have higher ambiguity in the

	From scratch		From greedy	
	μ	σ	μ	σ
1	58.61	0.63	58.71	0.84
2	4.73	0.58	5.19	1.30
3	438.45	6.26	438.66	6.04
4	6127.57	21.70	6092.93	9.92
5	4951.30	6.19	4956.36	9.17
6	819.30	3.33	819.61	3.20
7	469.18	30.15	472.51	13.16

Table 5.12: $-\log(p)$ compared for all the sequences, with 400 chains started from scratch, and 400 chains started from the solution found by greedy search.

detections. The plot in Figure 5.33 is though misleading as the normalising factor in the posterior changes when new detections are added. Accordingly, the y-axis data cannot be compared across activity units. Complete results for online optimisation can be found in Appendix E.1.

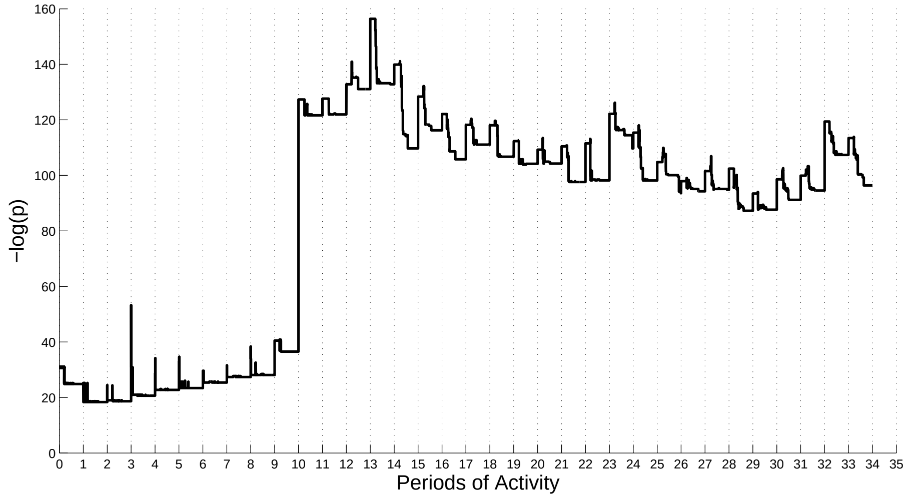


Figure 5.33: Online RJMCMC-SA for the 1^{st} sequence. Vertical dotted lines separate the optimisation at each activity unit. The Markov chain length is 1000 steps for each activity unit.

5.6.2 Accuracy results

The ground truth was manually obtained for each sequence, labelling each person with the event accomplished (dropping, picking or passing-by), then connecting any pick to its earlier drop. Figure 5.34 shows an example of the recorded ground truth. Notice that this ground truth is partial, as it does not connect people to bicycle-clusters. This was avoided due to the complexity of manually deciding on those connections. Recall that a drop event cannot be connected to its pick event unless the bicycle-clusters are correctly connected. Thus, comparing an explanation to this partial ground truth is sufficient to assess the accuracy of the global explanation.

Upon retrieving the MAP global explanation, it is compared to the ground truth to calculate the accuracy. Figure 5.35 presents a partial explanation from the 4^{th} sequence that corresponds to the ground truth in Figure 5.34. When compared to the ground truth, the accuracy equals the ratio of the correctly explained records to the total number of records in the sequence. A record is explained correctly if it matches the ground truth, or is redundant to a correctly-explained record. The latter case explains broken tracks. For example, the last two records in Figure 5.34 are 2 tracklets of the same track. Explaining any of them correctly, while explaining the other as an unconnected pick (as in Figure 5.35) is considered a correct explanation for both records. When this is compared to the ground truth, 9 out of 10 records in Figure 5.35 are correctly explained resulting in 90% accuracy.

track ID	**track Name**	**event (d/p/s)**	**previous drop**	**diff Person?**	**diff Clothing?**
19234	N	s			
21586	O	d			
26353	Q	d			
28402	R	d			
29978	Z	d			
30310	OP	p	O	0	1
30355	OP	p	O	0	1
31027	Y	s			
31445	QP	p	Q	0	0
31623	QP	p	Q	0	0

Figure 5.34: A sample ground truth from the 4^{th} sequence. Each track is assigned an ID by the tracker. A unique name is given to each person to show broken trajectories, along with the event performed. For pick events, the track name of the previous drop is recorded. Simulated thefts and people with different clothing are recorded as Boolean variables.

track ID	**bicycle-cluster No**	**event (d/p/s)**	**previous drop**
19234	0	s	
21586	124	d	
26353	126	d	
28402	127	d	
29978	128	d	
30310	130	p	21586
30355	0	s	
31027	129	d	
31445	131	p	-
31623	131	p	26353

Figure 5.35: A sample partial explanation from the 4^{th} sequence.

The MAP results from the previous section are compared to the ground truth. Several explanations evaluate to the same accuracy rate if they contain the same number of correctly explained records. Figure 5.36 plots the posterior probability along with the accuracy results for one RJMCMC run from the 5^{th} sequence. The figure demonstrates

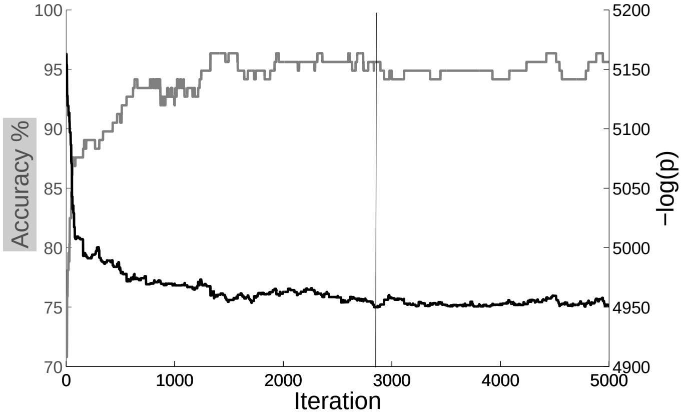


Figure 5.36: The posterior results and the corresponding accuracy rates for one RJMCMC run (5^{th} sequence). The vertical line shows the MAP solution, and its corresponding accuracy. Higher accuracy rates are present, yet those have lower posteriors.

a general trend of increase in accuracy with the increase in posterior. Yet, the MAP explanation does not correspond to the highest retrieved accuracy. This could have resulted from two different causes:

1. Incorrect modeling of the priors and conditional probabilities.
2. Insufficient information in the features selected.

The accuracies for the MAP explanations from table 5.6 are shown in Table 5.13. In the table, five out of the seven sequences have the highest accuracy corresponding to the MAP. In all sequences, a higher posterior corresponds at times to a lower accuracy. The complete tables of accuracies are shown in Appendix E.2.

	Local	**Global**									
		Greedy	**MHT**			**RJMCMC**		**RJMCMC-SA**		**IP**	
			k=50	**k=100**	**k=500**	μ	σ	μ	σ	**MATLAB**	**XPRESS-MP**
1	74.13	72.41	91.38	91.38	91.38	88.36	1.09	87.46	1.79	91.38	91.38
2	85.19	85.19	100.00	100.00	100.00	100.00	0.00	100.00	0.00	100.00	100.00
3	64.06	58.59	84.38	84.38	84.38	87.68	0.89	83.36	1.65	88.28⋆	87.5⋆
4	74.60	73.81	74.60	75.40	75.40	83.93	1.09	83.15	1.31	81.75⋆	83.33⋆
5	86.13	89.05	82.48	84.67	88.32	91.90	0.79	92.65⋆	0.90	94.16	94.16
6	65.18	66.07	60.71	60.71	60.71	68.53	1.68	70.98	1.04	73.21	73.21
7	46.18	45.69	44.67	45.69	-	47.28	1.18	47.61	0.88	-	46.70

Table 5.13: 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.

The table also compares local and global analysis. A local solution is finding the best event for each person independently then linking drops and picks allowing the same drop

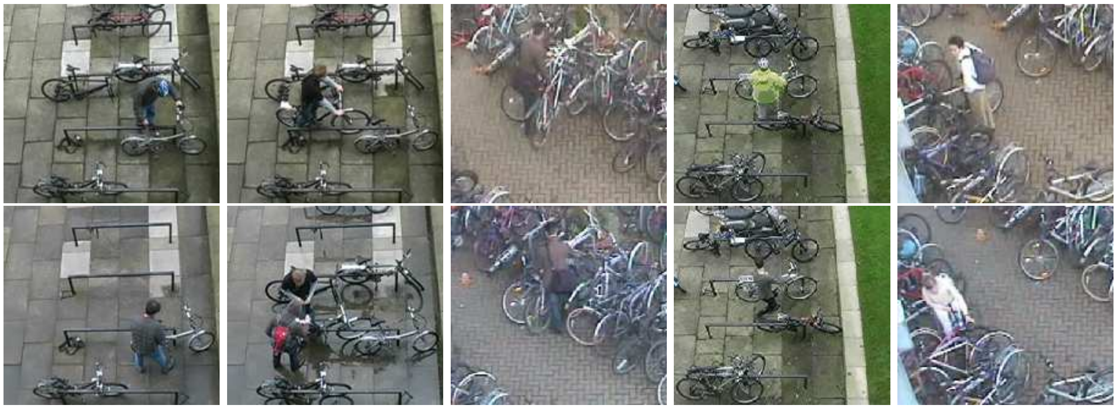


Figure 5.37: Five examples of connected events. The first four are correctly connected. The fourth column represents a simulated theft. The fifth example shows an incorrect connection. Recall that no clothing color comparison is performed. Individuals are connected by linking the person to a cluster and correctly linking dropped to picked bicycle-clusters.

to link to several pick events and vice versa. The local solution is thus a complete but possibly inconsistent set of events, as the inter-activity constraints are not maintained. The results show higher accuracy for global explanations. This indicates that global explanations can resolve ambiguities that cannot be resolved by local analysis.

It is expected that the accuracies for sequences (6-7) are lower due to the increase in clutter. The 7^{th} sequence suffers from frequent abrupt lighting changes that result in bicycle clusters being poorly detected. Figure 5.37 gives some examples of recognized and linked drop and pick events across the dataset.

5.6.3 Theft detection

The explanations for the *Bicycles* problem can be used to detect thefts. A *theft* is defined as a drop-pick compound event where the person who dropped the bicycle does not ‘look-like’ the person picking the bicycle. Soft-biometrics can be used to compare

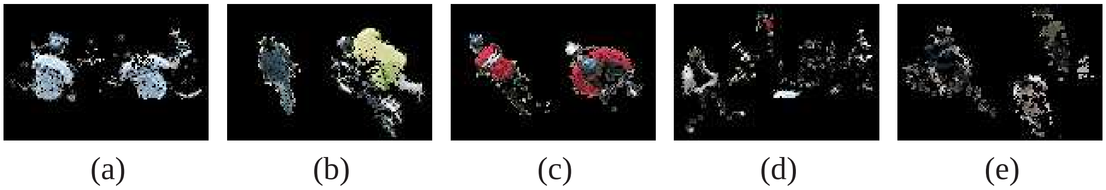


Figure 5.38: Examples of the best-matched pair of frames when comparing two people. This technique finds the best match for the same person (a), and a poor match for different individuals (b). Nevertheless, it tries to find as high match as possible across different people (c) and fails in cases of poor segmentation (d) and occlusion (e).

the appearances from the CCTV camera. Clothing colour, height or body build can be compared. In this section, only clothing colour is used for matching. This is based on the assumption that people do not change their clothing between the two events. This assumption is of course not always valid. Moreover, colour matching is affected by lighting changes in outdoor scenes. This section first explains how the colour information can be retrieved, summarised and compared. Then, the results are presented.

The tracker [100] provides colour information for foreground pixels per frame throughout the trajectory's duration. For efficiency, the colour information is summarised across the whole trajectory. First, data is summarised per frame. An $8 \times 8 \times 8$ scale-normalised equal-bin-size RGB colour histogram was generated from the foreground pixels at each frame. 'Scale-by-max' per channel is used as a simple colour constancy algorithm [11]. One way to measure the similarity between two people x_1 and x_2 is to compare the histograms for all pairs of frames. Assume $H(x_1, j)$ represents the colour histogram for person x_1 at frame j, and that H_i is a certain bin in the histogram of size m. The similarity between histograms is produced using the symmetric version of the *histogram intersection* introduced in [136].

$$\chi(H(x_1,l),H(x_2,k)) = \sum_{i=1}^{m} min\Big(\frac{H_i(x_1,l)}{\sum_j H_j(x_1,l)}, \frac{H_i(x_2,k)}{\sum_j H_j(x_2,k)}\Big) \tag{5.16}$$

Since $x_i.n$ measures the number of frames for each person, assume $l = 1 \ldots x_1.n$ and $k = 1 \ldots x_2.n$, then the overall similarity $\delta_c(x_1,x_2)$ between two people is defined in Equation 5.17

$$\delta_c(x_1,x_2) = \max_{l,k}(\chi(H(x_1,l),H(x_2,k))) \tag{5.17}$$

This computation is expensive as it requires maintaining a histogram for each frame, and calculating the intersection of all pairs. Moreover, it is error-prone to foreground segmentation problems. Figure 5.38 contains a collection of examples showing the best matched pair of frames.

Alternatively, all histograms for a person can be combined together. A per-bin median histogram $\tilde{H}$ was calculated across all frames as explained in [19].

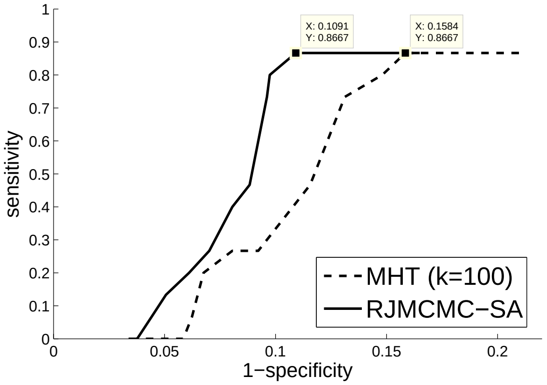
$$\tilde{H}_i(x_1) = \text{median}_{j=1\ldots x_1.n} H_i(x_1, j) \tag{5.18}$$

The similarity between two people is then calculated as the intersection between the per-bin median histograms.

$$\delta_c(x_1,x_2) = \chi(\tilde{H}(x_1),\tilde{H}(x_2)) \tag{5.19}$$

After the global explanation is found, the drop-pick events are studied further, and the clothing colour is compared for the dropping and picking people. If the clothing match δ_c is below a certain threshold, a theft warning is raised. Figure 5.39 presents the ROC curve for different thresholds summed over all the sequences. Recall that clothing colour was not used in the activity recognition.

The ground truth contains information about whether the picking individual is a different person. When compared to ground truth, a True Positive (TP) is a labeled theft case when different people in the ground truth are recorded. A True Negative (TN) indicates the same individual, and the explanation not raising a theft warnings. A False Positive (FP) is recorded when a warning is raised while the ground truth records the events are performed by the same person. A FP can be generated from an incorrect explanation, different clothing, or poor colour matching. Finally, a False Negative (FN) is caused by a theft case that is not detected.



MHT	**Predicted**	
Actual	Thief	Non-Thief
Thief	13	2
Non-Thief	122	648

RJMCMC-SA	**Predicted**	
Actual	Thief	Non-Thief
Thief	13	2
Non-Thief	84	686

Figure 5.39: ROC curve (left) representing theft detection results. 0.7 was selected as the threshold to calculate the confusion matrix (right).

At a threshold of 0.7, 87% (13 out of 15) of the theft cases were caught for a 10.9% (RJMCMC-SA) false-positive rate. This section shows how global explanations can be used for theft detection. The results are a promising start, but further soft biometrics (see Section 2.3.3) and colour constancy analysis are required before the application can be tested. Moreover, a theft warning should marginalise over possible explanations rather than conclude from the MAP. Using the application in practice requires a wider analysis of the risk and the reward in raising theft warnings (see Section 8.3).

5.7 Conclusion

This chapter shows how the *Bicycles* problem can be solved using the framework presented in Chapters 3 and 4. This includes the formulation of an AMG and the estimation of the likelihood functions for different attribute values. The method is evaluated on 67 hours of video from two bicycle racks.

Searching for the best explanation is performed using the techniques from Chapter 4. Tested on 7 sequences of varying length and complexity, RJMCMC-SA achieved the best heuristic search results. The results section presents an extensive analysis for finding the MAP solution. This solution is then compared to the manually obtained ground truth to calculate accuracy rates.

Results presented demonstrate the ability of the framework to find the best global explanation. It makes the case for global explanations over local analysis of events, by proving global analysis achieves better accuracy results. The next chapter applies the framework to a related but different problem.

Chapter 6

Case II: The *Enter-Exit* Problem

This chapter presents a different challenging problem that requires tracking people, and any objects they might be carrying, as they enter and exit a building. The number of interleaved events is substantial, and the combinatorics of the problem can easily prove intractable. A global explanation links the event of a 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 him/her returning later. The linking depends on comparing the person and the baggage biometrics between both appearances. Matching the objects these people are carrying could assist in highlighting potential threats from a security perspective - for example discovering any baggage abandoned within the building.

In achieving this task, the carried object detector, to be presented in Chapter 7, is used. Section 6.2 presents a complete attribute multiset grammar for the task. The grammar parses all detections into a global explanation. The Bayesian network can then be searched for the MAP solution. Section 6.3 reviews how the values of the synthetic attributes are assigned, and assesses the ability of each feature to recognise the occurring event. Next, the reversible moves used by the RJMCMC search are reviewed (Section 6.4). The prototype was evaluated on 12 hours of recorded video. Results are analysed in Section 6.5.

6.1 The *Enter-Exit* problem

The *Enter-Exit* problem discussed in this chapter is the task of recognising people as they enter and exit a building, using one camera mounted next to a building entrance. Natural constraints govern the possible sequence of detections, e.g. a person entering the building can be observed departing only once, and at a later point in time. The explanation is a consistent set of links between people entering and exiting the building together with information on any objects they might be carrying.

This problem is similar to the task of tracking people across a blind area, or between non-overlapping cameras. Two essential differences should though be highlighted. The first is that blind area tracking usually relies on temporally linking the detections. A person is expected to emerge again within a certain amount of time. This limits the number of interleaved events making the number of possible explanations tractable in most cases. The second difference is that blind area tracking classifies trajectories in advance into those entering and exiting the blind area. This classification cannot be amended.

The *Enter-Exit* problem resembles the *Bicycles* problem presented in Chapter 5 in that two types of detections are available; people and bicycles in the first case, and people and bags in the second. It also has two types of events to be linked; drops and picks versus enters and exits. It differs though in that each event can relate to both an earlier and a subsequent event of the opposite type. This enables recognising sequences of events: enter-exit-enter-exit-enter, while the *Bicycles* problem only recognises a single drop-pick instance of the bicycle. This adds extra complexity to the domain and the search space. The next section presents the complete attribute multiset grammar for the *Enter-Exit* problem that tracks people and carried objects around one building entrance.

The person detections were retrieved using the same off-the-shelf tracker [100]. As before, a separate person detection is derived from each trajectory. The identity of the person cannot be maintained by the tracker after departing the field of view. Detecting bags is based on a novel detector presented in the next chapter. For each trajectory, protrusions representing candidate carried objects are retrieved. The location and colour of the pixels representing the candidate bag are recorded for each frame along the trajectory.

6.2 An AMG for the *Enter-Exit* problem

For this new problem, the activity is defined using the following AMG.

Terminals (T):	t	person detection
	b	baggage detection

	u	unobserved enter or exit events
Nonterimanls (N):	S	Start symbol
	X	Exit-Enter link - linking an exit event to a subsequent enter event
	E	Enter-Exit link - linking an enter event to a subsequent exit event
	C	Linking the person to a collection of carried objects
	B	Collection of carried objects

Attributes (A):

symbol	**att. name**	**type**	**domain**	**description**	**pdf**
t	id	A_0	$\mathbb{Z}$	a unique id differentiating people detections	
	time	A_0	$\mathbb{Z}^2$	duration in which the person is tracked	
	n	A_0	$\mathbb{Z}$	number of frames with the person visible	
	medianColour	A_0	$\mathbb{R}^{512}$	per-bin median histogram of pixel colours	
	projectedHeights	A_0	$\mathbb{R}^n$	list of projected heights across frames	
	angle	A_0	$\mathbb{R}$	mean walking direction	p(t.angle|t)
	count	A_1	$\{0,1\}$	number of enter or exit events for the detection	
	action	A_1	$\{$enter, exit, pass-by$\}$		
b	id	A_0	$\mathbb{Z}$	id of the trajectory	
	frequency	A_0	$\mathbb{R}$	ratio of frames in which the protrusion is detected	p(b.frequency|b)
	colourSimilarity	A_0	$\mathbb{R}$	colour similarity with neighbouring clothing	p(b.colourSim|b)
	relativeHeight	A_0	$\mathbb{R}^2$	vertical extent of the carried object relative to the individual	
	medianColour	A_0	$\mathbb{R}^{512}$	per-bin median histogram of pixel colours	
	count	A_1	$\{0,1\}$	number of enter or exit events for the bag	
	action	A_1	$\{$carried, other$\}$		
X	bagDiff	A_0	$\mathbb{Z}$	number of bags that do not match	
	match	A_0	$\mathbb{R}$	likelihood of matching an exit to a later enter	p(X.match|X)
	action	A_1	$\{$exit-enter, exit-u, u-enter, f$\}$		
E	bagDiff	A_0	$\mathbb{Z}$	number of bags that do not match	
	match	A_0	$\mathbb{R}$	likelihood of matching an enter to a later exit	p(E.match|E)
	action	A_1	$\{$enter-exit, enter-u, u-exit, f$\}$		
C	trajID	A_0	$\mathbb{Z}$	= t.trajID	
	nb	A_0	$\mathbb{Z}$	= B.nb	
	time	A_0	$\mathbb{Z}^2$	= t.time	
	relativeHeights	A_0	$\mathbb{R}^{2\times nb}$	= B.relativeHeights	
	medianColours	A_0	$\mathbb{R}^{512\times nb}$	= B.medianColours	
	medianColour	A_0	$\mathbb{R}^{512}$	= t.medianColour	
	projectedHeights	A_0	$\mathbb{R}^n$	= t.projectedHeights	
	angle	A_0	$\mathbb{R}$	= t.angle	
	eCount	A_1	$\{0,1\}$	the number of enter-exit events	
	xCount	A_1	$\{0,1\}$	the number of exit-enter events	
	action	A_0	$\{$enter, exit, f$\}$		
B	trajID	A_0	$\mathbb{Z}$	= b.trajID	
	nb	A_0	$\mathbb{Z}$	number of carried bags	
	relativeHeights	A_0	$\mathbb{R}^{2\times nb}$	list of b.relativeHeight	
	medianColours	A_0	$\mathbb{R}^{512\times nb}$	list of b.medianColour	
	action	A_1	$\{$enter, exit, f$\}$		

Attribute Functions

$\text{time}_1 < \text{time}_2$	this operator ensures the $time_1$ ends before $time_2$ starts.
$\psi_M(C_1, C_2)$	measures the likelihood of matching two people

Production Rules (P)

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

Figure 6.1 presents the attribute dependency graph for the AMG. In accordance with the framework presented in Chapter 3, given a set of detections, a parse tree of this grammar represents a global explanation. The Bayesian network, modelling the probability

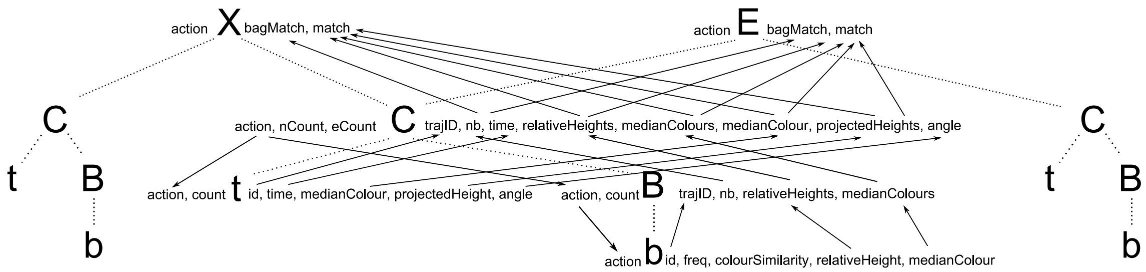

action X bagMatch, match
action E bagMatch, match
C
action, nCount, eCount C trajID, nb, time, relativeHeights, medianColours, medianColour, projectedHeights, angle
C
t B
action, count t id, time, medianColour, projectedHeight, angle
action, count B trajID, nb, relativeHeights, medianColours
t B
b
action b id, freq, colourSimilarity, relativeHeight, medianColour
b

Figure 6.1: The attribute dependency graph for the Enter-Exit problem AMG.

distribution over all possible parse trees, can be built and searched in a similar way to that presented in the previous case study.

Given a multiset of detections with some attribute values D = $\{t_1$ (trajID=1, time=1), t_2 (trajID=2, time=2), b_1 (trajID=1), b_2 (trajID=1), b_3 (trajID=1), b_4 (trajID=2)$\}$, Figure 6.2 shows a parse tree and the corresponding Bayesian network.

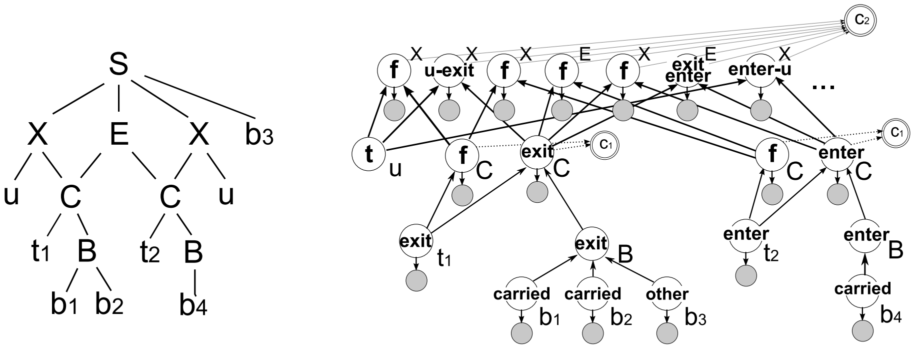
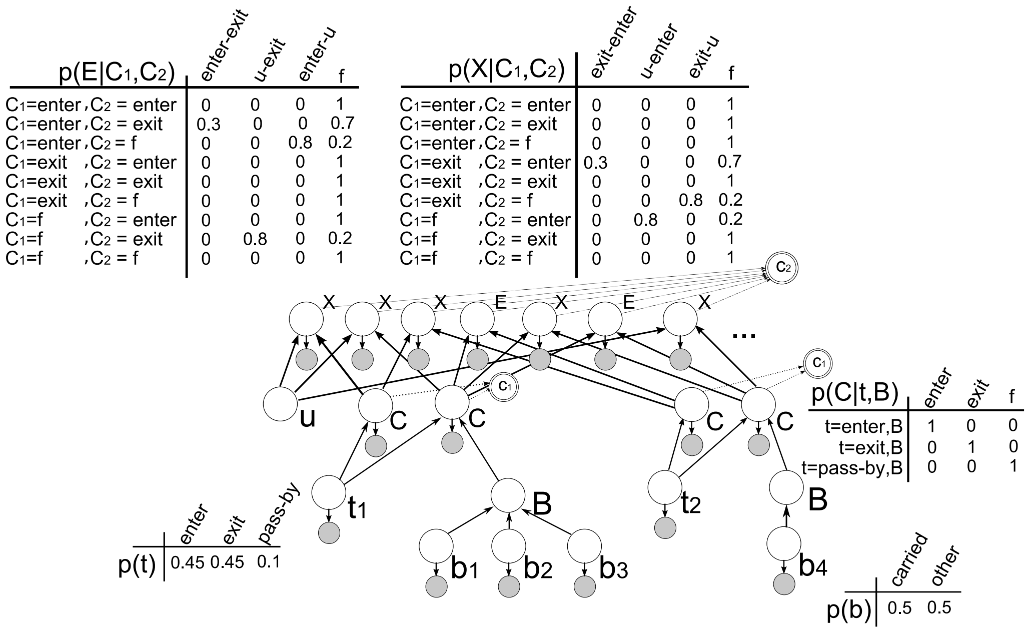


Figure 6.2: A sample parse tree and labelled BN for the Enter-Exit problem.

After building the structure of the Bayesian network for a set of detections based on the AMG, the Bayesian network's parameters (i.e. priors and conditional probabilities) can be defined. Figure 6.3 shows the set of priors and conditional probabilities for the problem based on expertise knowledge. The next section describes how the observed RV likelihoods were trained.

6.3 Features selection and supervised training

The detectors are expected to retrieve a multiset of terminals along with specified values for the synthetic attributes of each terminal. These synthetic attributes are described in the AMG above. This section describes how these features are retrieved from the video, and trained for the possible labels. The median colour feature is identical to that used for the *Bicycles* problem (Section 5.6.3). This section describes how the remaining features



p(E|C1,C2)	enter-exit	u-exit	enter-u	f
C1=enter ,C2 = enter	0	0	0	1
C1=enter ,C2 = exit	0.3	0	0	0.7
C1=enter ,C2 = f	0	0	0.8	0.2
C1=exit ,C2 = enter	0	0	0	1
C1=exit ,C2 = exit	0	0	0	1
C1=exit ,C2 = f	0	0	0	1
C1=f ,C2 = enter	0	0	0	1
C1=f ,C2 = exit	0	0.8	0	0.2
C1=f ,C2 = f	0	0	0	1

p(X|C1,C2)	exit-enter	u-enter	exit-u	f
C1=enter ,C2 = enter	0	0	0	1
C1=enter ,C2 = exit	0	0	0	1
C1=enter ,C2 = f	0	0	0	1
C1=exit ,C2 = enter	0.3	0	0	0.7
C1=exit ,C2 = exit	0	0	0	1
C1=exit ,C2 = f	0	0	0.8	0.2
C1=f ,C2 = enter	0	0.8	0	0.2
C1=f ,C2 = exit	0	0	0	1
C1=f ,C2 = f	0	0	0	1

p(C|t,B)	enter	exit	f
t=enter,B	1	0	0
t=exit,B	0	1	0
t=pass-by,B	0	0	1

	enter	exit	pass-by
p(t)	0.45	0.45	0.1

	carried	other
p(b)	0.5	0.5

Figure 6.3: Priors and conditional probabilities estimated for the Enter-Exit BN.

are obtained and trained. The training sequence was two hours recorded from the same viewpoint on a separate day.

6.3.1 $t.projectedHeights$

Previous work tried to estimate the actual height of the individual to be used in matching people between non-overlapping cameras [99]. Estimating the actual height requires a full camera calibration. This section presents a way to compare the height distributions for two person detections viewed within the same camera using the ground-plane homography.

Given the vanishing point and the horizon's vanishing line, the height of a vertically-standing object can be computed, up to a constant factor at each frame. As the person is not standing upright during walking, only the elevation of the top of the head from the ground plane can be estimated. The top of the head is approximated to be the highest point of the foreground segmentation blob. The elevation of this point above the ground is referred to as the projected height. The projected height is expected to vary with the phase of the gait. The distribution of projected heights can be estimated from all frames of the person's trajectory. Two such distributions representing the projected heights of two people can then be compared as will be shown in Section 6.3.6.

The projected height of the individual, up to a constant factor, can be calculated from the cross-ratio illustrated in Figure 6.4. x is the position of the top of the head, x' is the vertical projection of that point on the ground plane, v is the vanishing point representing

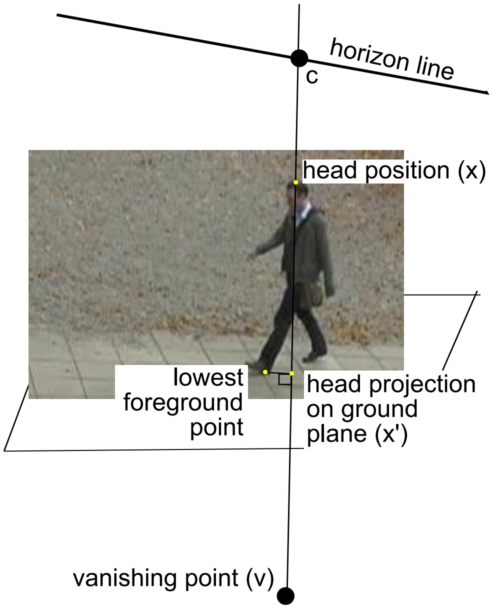


Figure 6.4: The cross-ratio is calculated using the points x, x', v, c.

the vertical projection of the camera's position onto the ground plane, and c is the intersection of the line connecting the head x to the vanishing point v with the horizon line. The horizon line can be calculated for a given scene, using sets of parallel lines on the ground plane. The vanishing point is calculated from the intersection of parallel lines that are perpendicular to the ground plane.

The vertical projection of the head on the ground plane x' was estimated by projecting the lowest vertical point in the foreground segmentation onto the line $\overline{vx}$. The Euclidean distance d between two points can then be used to find the cross-ratio in the image plane. Given the camera's height above the ground plane Z_c, the height of the individual Z is given by [30]:

$$Z = Z_c(1 - \frac{d(x,v)d(x',c)}{d(x,c)d(x',v)}) \tag{6.1}$$

Though Z_c is unknown, the cross-ratio $r = \frac{d(x,v)d(x',c)}{d(x,c)d(x',v)}$ can be calculated for each frame. Figure 6.5 shows an example of the variation in cross-ratio r for a single person over several frames. Assuming a Gaussian distribution, the mean μ and the standard deviation σ are calculated for the complete trajectory.

6.3.2 *b.relativeHeight*

The relative height of each carried object is the vertical extent of the baggage's bounding box relative to that of the temporal template (Section 7.2). Assume (h_1, h_2) define the

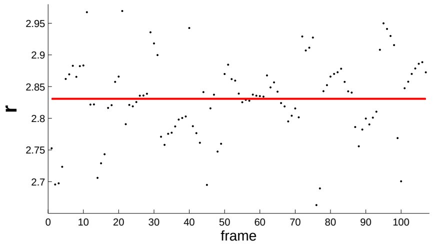


Figure 6.5: An example of the cross-ratio r along the trajectory of a walking person. The horizontal line marks the mean of the Gaussian distribution estimated from this data.

top and bottom vertical positions of the temporal template, and (b_1, b_2) are the top and bottom vertical positions of the carried object as in Figure 6.6, then the relativeHeight is the tuple:

$$b.\text{relativeHeight} = (\frac{b_1 - h_1}{h_2 - h_1}, \frac{b_2 - h_1}{h_2 - h_1}) \tag{6.2}$$

This attribute is used to match carried objects between people as will be seen in Section 6.3.6.

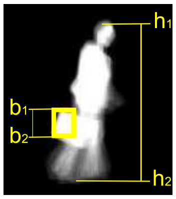


Figure 6.6: The relative height of the baggage from the temporal template. Temporal templates will be explained in Section 7.2.

6.3.3 $p(t.angle|t.action)$

The angle of the walking direction is calculated from the means of the foreground blobs in the image plane. Considering the positions of the person, a best fitting vector is found by linear regression, and the angle of that vector is used to classify people entering, exiting or passing by. The conditional probability of an angle given one of these classes is estimated from labelled examples using a wrapped Gaussian. A wrapped Gaussian (also referred to as the von Mises distribution) is suitable for representing directional statistics as it is wrapped around the circumference of a unit circle [46]. The wrapped Gaussian probability density function p_w is defined in terms of the Gaussian function p_g in Equation 6.3.

$$p_w(\theta) = \sum_{k=-\infty}^{\infty} p_g(\theta + 2\pi k) \tag{6.3}$$

The mean of the wrapped Gaussian is defined in terms of the mean of the sine and cosine values of the angles in the sample. If $\overline{S}$ is the mean of the sines and $\overline{C}$ is the mean of the cosines, then the mean angle $\overline{\theta}$ is as defined in Equation 6.4 [46].

$$\overline{\theta} = arctan(\overline{S}/\overline{C}) \tag{6.4}$$

Figure 6.7 illustrates the trained wrapped Gaussians for the three possible event types.

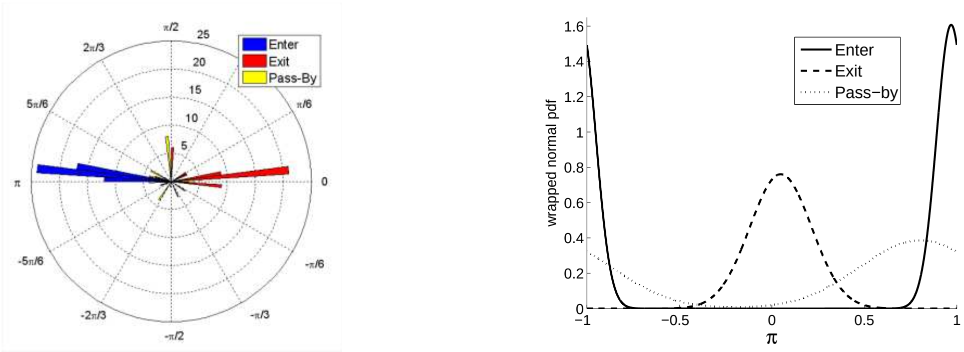


Figure 6.7: Angular histogram of walking directions used in training (left) and wrapped Gaussians estimated from the angular histograms (right).

6.3.4 $p(b.frequency|b.action)$

The frequency of a protrusion is the ratio of the number of frames during which the protrusion was detected to the total number of frames. This is one way to classify protrusions into carried objects and other protrusions, but proved to be only weakly discriminative. Figure 6.8 shows the trained Gaussians. It is still included in the posterior calculations.

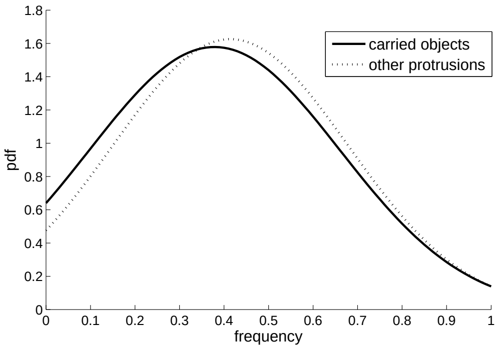


Figure 6.8: Trained Gaussians for the frequency of carried objects and other protrusions.

6.3.5 $p(b.colourSimilarity|b.action)$

Another feature to classify protrusions into carried objects and other protrusions is to compare the colour of the foreground segment representing the protrusion to that of the neighbouring foreground region. The set of pixels representing the neighbouring foreground region is defined as $\{p : \exists q \in bagPixels : (d(p,q) < \varepsilon)\}$. The horizontal distance is used $d(p,q) = |p.x - q.x|$, and ε is set to one sixth of the person's height. Figure 6.9 shows an example of this region.

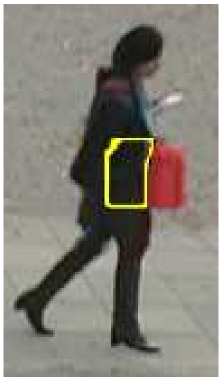


Figure 6.9: The red-coloured region signifies the pixels added to the carried object's colour histogram, while the yellow-bounded region signifies the neighbouring region's pixels.

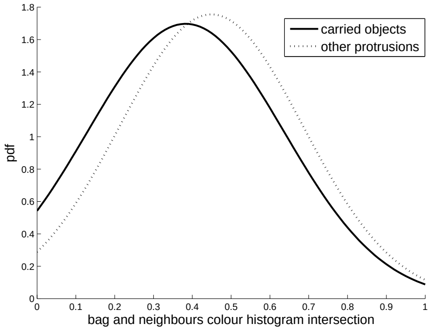


Figure 6.10: Trained Gaussians for the baggage colour similarity.

The per-bin median histogram of all frames is then accumulated for the bag pixels H_b and the neighbouring pixels H_n. The histogram intersection [136] $\chi(H_b, H_n)$ is used to measure the colour similarity between the bag and the neighbouring foreground pixels. The cpdf for these similarity values given carried objects and other protrusions is a Gaussian with mean and standard deviation estimated from examples. Figure 6.10 shows the trained Gaussians. The results demonstrate that this feature is not very discriminative either. This is because colour is illumination variant, and many people carry bags of matching colours to their clothes. A person wearing a black suit and carrying a black suitcase is a common detection within the recorded data.

6.3.6 $p(X.match|X.action)$ and $p(E.match|E.action)$

The function $\psi_M(C_1, C_2)$ matches the median colour histograms and projected height distributions of two person detections. It also considers matching any carried objects these people are carrying; the colour and relative height of the bags are compared. This section describes how the matching is performed.

Section 6.3.1 described how the distribution of projected heights is calculated for a single person. When matching two people, the Welch t-test compares the two samples of projected heights and generates a matching score. For two distributions $\mathcal{N}_1(\mu_1, \sigma_1, N_1)$, $\mathcal{N}_2(\mu_2, \sigma_2, N_2)$, t is evaluated using Equation 6.5

$$t = \left| \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2}}} \right| \tag{6.5}$$

Figure 6.11 shows Gaussians estimated from the t values for examples of projected height matches for the same person and different people. Correct matches were based on ground truth pairs within the training data. All the remaining possible pairings, within the training data, are used to train for incorrect projected height matches.

Next, clothing colour matching is achieved by histogram intersection of per-bin median histograms (Equation 4.7). By training for the same and different people, Figure 6.12 shows the probability density functions for clothing colour matches.

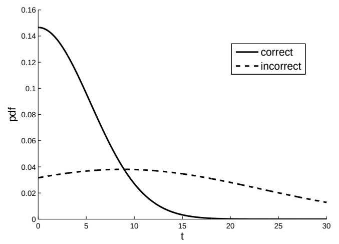


Figure 6.11: Gaussian density functions for height match scores given the same person and different people.

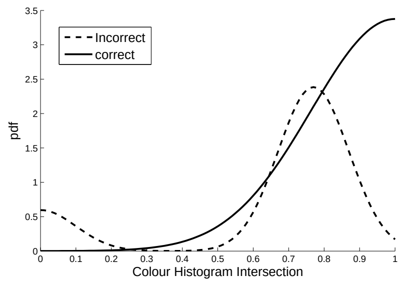


Figure 6.12: Gaussian density functions for colour histogram intersection given the same and different people.

When matching carried objects, $\psi_{heightOverlap}$ is the likelihood of matching two bags based on their relative heights. The relative height of each bag is calculated as explained in Section 6.3.2, and is matched using the interval overlap in Equation 6.6 where relative height tuples are treated as closed intervals. For two bags b_1 and b_2,

$$\psi_{heightOverlap} = \frac{|b_1.relativeHeight \cap b_2.relativeHeight|}{|b_1.relativeHeight \cup b_2.relativeHeight|} \tag{6.6}$$

$\psi_{heightOverlap}$ is trained for correct and incorrect bag matches, as shown in Figure 6.13. Also the colour of the bags are compared. The pdfs of the median colour histogram intersection ($\psi_{bagColour}$), modelled as Gaussians, are shown in Figure 6.14

Given two events C_1 and C_2, where C_i.nb signifies the number of objects carried by

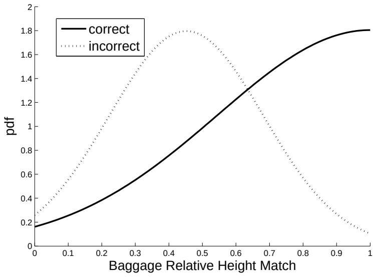


Figure 6.13: Correct and incorrect carried baggage relative height matchings trained into Gaussians.

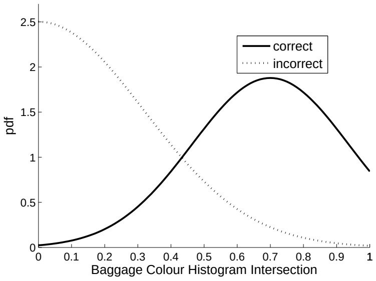


Figure 6.14: Correct and incorrect baggage colour matchings trained into Gaussians.

each person, assume $h = \psi_{heightOverlap}(i,j)$ and $c = \psi_{bagColour}(i,j)$, then the baggage match

$$\psi_{carried}(C_1,C_2) = \begin{cases} C_1.nb = 0\&C_2.nb = 0 & \kappa \\ C_1.nb = 0|C_1.nb = 0 & 1-\kappa \\ otherwise & \max\{\kappa \times \max\limits_{i=1..C_1.nb, j=1..C_2.nb}(pdf_{heightOverlap}(h|correct)pdf_{colour}(c|correct)), \\ & (1-\kappa) \times \max\limits_{i=1..C_1.nb, j=1..C_2.nb}(pdf_{heightOverlap}(h|incorrect)pdf_{colour}(c|incorrect))\} \end{cases} \tag{6.7}$$

In Equation 6.7 κ is the expected prior of baggage matches, and was set to 0.7 in all experiments.

Thus, $p(X.match|X.action)$ and $p(E.match|E.action)$ match the person's projected height and clothing colour along with matching any carried objects using $\psi_{carried}$.

6.4 Reversible moves for the *Enter-Exit* problem

The same general set of reversible moves from Section 4.4.2 is used to traverse the space of explanations. Figure 6.15 shows a three-step Markov chain similar to Section 5.4 for the *Bicycles* problem. Yet, within-move proposal distributions are application-dependent. This section presents the proposal distribution within each move type.

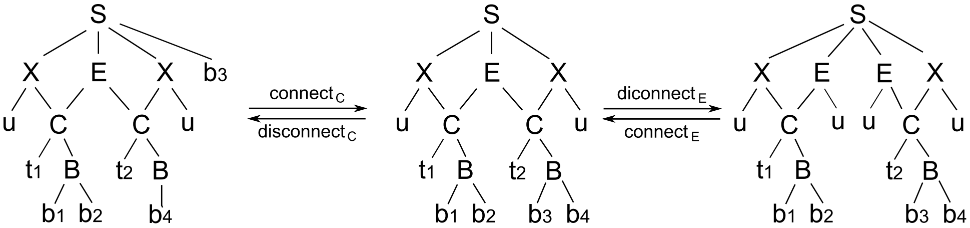


Figure 6.15: A three-step reversible Markov chain for the Enter-Exit problem.

Move type (A) connect$_c$/disconnect$_c$

Connecting a person to a bag is achieved by changing the bag from an ‘other’ protrusion to a ‘carried’ protrusion. For a protrusion b_i, the measurement for weighting moves of type *connect$_c$* is defined in Equation 6.8.

$$\delta_{connect_c}(b_i) = \frac{p(b_i.colourSimilarity|b_i.action = carried)}{p(b_i.colourSimilarity|b_i.action = noise)} \tag{6.8}$$

The measurement for the $disconnect_c$ move is the inverse of that for the $connect_c$ move.

$$\delta_{disconnect_c}(b_1) = \frac{p(b_i.colourSimilarity|b_i.action = noise)}{p(b_i.colourSimilarity|b_i.action = carried)} \tag{6.9}$$

Notice that the moves $change_c$ and $switch_c$ are not defined as the protrusion can only be related to one trajectory ($t.trajID = b.trajID$ in p_8).

Move type (B) connect$_e$/disconnect$_e$ and connect$_x$/disconnect$_x$

To connect an enter to a subsequent exit, or an exit to a subsequent enter, each possible move is weighted by:

$$\delta_{connect_e}(C_i,C_j) = \delta_{connect_x}(C_i,C_j) = \psi_M(C_i,C_j) \tag{6.10}$$

The disconnect move is weighted by the inverse.

$$\delta_{disconnect_e}(C_i,C_j) = \delta_{disconnect_x}(C_i,C_j) = \frac{1}{\psi_M(C_i,C_j)} \tag{6.11}$$

Move type (C) change$_e$ and change$_x$

Equation 6.12 shows the weight of changing an enter-exit event. The approach tries to find whether better alternatives are provided. Similar to the weighted measures in Chapter 5, $\delta_{change_E} > 1$ when better alternatives are available, and < 1 when the current connection has the highest likelihood.

$$\delta_{change_e}(C_i,C_j) = \frac{\max\left(\max_{\{k:C_k=exit\}-\{j\}} \psi_M(C_i,C_k), \max_{\{k:C_k=enter\}-\{i\}} \psi_M(C_k,C_j)\right)}{\psi_M(C_i,C_j)} \tag{6.12}$$

δ_{change_x} is calculated in the same way.

Move type (D) switch$_e$ and switch$_x$

$$\delta_{switch_e}((C_i,C_j),(C_k,C_l)) = \frac{\psi_M(C_i,C_l)\,\psi_M(C_k,C_j)}{\psi_M(C_i,C_j)\,\psi_M(C_k,C_l)} \tag{6.13}$$

δ_{switch_e} favours better connections, and similarly for δ_{switch_x}.

6.5 Experiments and results

While the *Bicycles* problem was applied to an extended dataset across two sites, this section only presents results for the *Enter-Exit* problem applied to one day of people entering and departing a building. It demonstrates though the power of the framework, and how it can be applied to analyse a different activity. A full day (12 hours) was recorded outside a building entrance. Figure 6.16 shows the viewpoint. The vanishing line was estimated from the image using the paving slabs on the ground. People standing upright were used in approximating the vanishing point, as the scene is clear of static vertically standing objects.

326 instances of someone passing through the entrance area were detected after manually rejecting groups of people walking together. The baggage detector from Chapter 7 resulted in 429 candidate bags. Section 7.3.2 will present a set of results for applying the baggage detector to this video sequence. It should be mentioned that previous research had investigated automatically counting the number of peoples in a group of walking pedestrians [39, 66]. Automatic detection of groups could thus be performed based on such research.

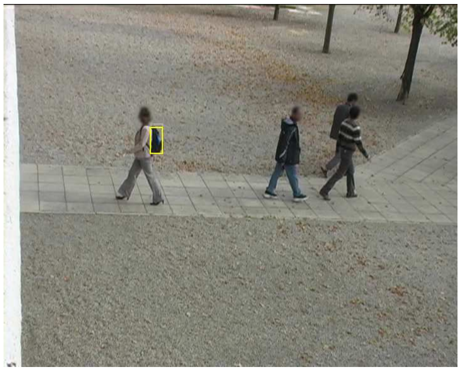


Figure 6.16: The camera viewpoint.

For 326 person detections and 429 candidate bags, a BN is constructed for the AMG presented in Section 6.2. The number of hidden RVs in the generated BN is 190849 ($|I(B)| = 116, |I(C)| = 435, |I(X)| = |I(E)| = 95149$). The MAP solution is obtained using greedy search, MHT and RJMCMC. Table 6.5 compares the MAP (represented as $-\log(p)$) obtained using the heuristic search techniques for the 12 hours video sequence.

	Greedy	**MHT**		**RJMCMC**		**RJMCMC-SA**	
		k=1	**k=20**	μ	σ	μ	σ
MAP	1143.47	1146.58	1137.70	1143.09	0.40	1123.02	1.12

Table 6.5: $-\log(p)$ compared across greedy, MHT (k = 1, 20), 40 runs of 10 parallel chains (n_{mc} = 5000) of RJMCMC and RJMCMC-SA. The result was not available for the MHT search with larger k due to the implementation running out of memory.

The IP solvers could not exhaustively search the space of explanations in reasonable time [1] as the constraints in this problem are far more complex than those in the *Bicycles* problem. Recall that the drop or the pick event in the *Bicycles* problem can participate in only one higher level event. Conflict constraints (refer to Section 4.5) are not required in that case. In the *Enter-Exit* problem, on the other hand, the enter event can be linked to an earlier exit as well as a later one. Conflict checking is thus required, which increases the number of constraints to be satisfied by the solver considerably. Both linear solvers (MATLAB and XPRESS-MP) were not able to reach a solution for the complete problem.

For a smaller-scale problem, Table 6.6 shows the MAP solution for the first 25 people (out of 326 in the dataset) and their corresponding candidate bags. The table shows that RJMCMC-SA is once again the best heuristic search technique. It's the only technique that was able to find the exact MAP (at some chains).

	Greedy	**MHT**			**RJMCMC**		**RJMCMC-SA**		**IP**
		k=1	**k=50**	**k=500**	μ	σ	μ	σ	**XPRESS-MP**
$-\log(p)$	85.61	85.49	84.47	84.47	85.55	0.13	84.29	0.03	84.27

Table 6.6: For a smaller-scale version, the results are compared for the first 25 people detections and the corresponding candidate bags. $-\log(p)$ compared across greedy, MHT (k = 1, 50, 500), 40 runs of parallel chains RJMCMC and 40 runs of parallel chains RJMCMC-SA and Integer Programming.

The ground truth was manually obtained, in which 62 pairings are found, with each pair connecting a person entering the building to him/her leaving later, or a person leaving the building and subsequently returning to it. Performing a manual ground truth proved to be a tedious task. For each observed person, the observer has to go through the rest of the recorded video and check whether the person has been seen again. It was noticed that one cannot keep in the memory more than a few people (2-3) at a time to correctly perform

[1] even using 20GB of memory for about 10 hours

the matches. After several rounds, 62 pairs were found, and set as the ground truth for the video.

Figure 6.17 presents a precision-recall curve that compares the three search techniques when varying the priors for connecting enter and exit events. Table 6.7 shows the number of correctly paired activities using the priors in Figure 6.3. Notice that the best search technique only found 19 of the 62 ground truth pairs. This is because the selected features (height and clothing colour) are only weak cues, as they vary with segmentation errors and illumination changes. Moreover, a high number of false positive connections indicates that while the priors are favouring connections more than they should, the weak cues make it hard to distinguish correct from incorrect connections.

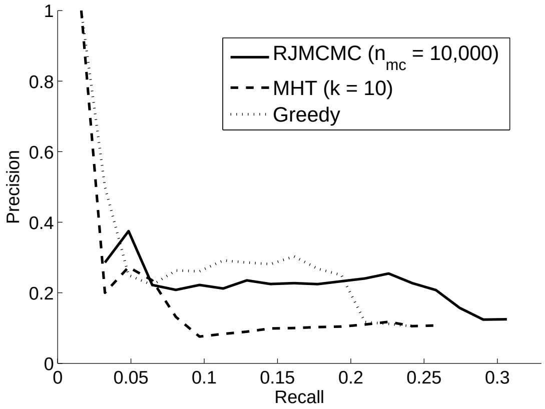


Figure 6.17: Precision-Recall curve for the three heuristic search techniques.

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

Table 6.7: The number of correctly paired activities, given expertise knowledge priors, comparing the unconstrained local explanation with global explanations found using heuristic search techniques.

Figure 6.18 shows three sequences that were correctly retrieved only when the global explanation is found using RJMCMC-SA. The intermediate example failed to be correctly paired originally because the object carried as the person returns to the building was not recognised as a carried protrusion. As the search progressed, a higher posterior was found by labelling the protrusion as a carried object and linking the 'exit' to the subsequent 'enter'. The figure also shows the framework's ability to correctly discover an 'exit-enter-exit-enter' sequence.

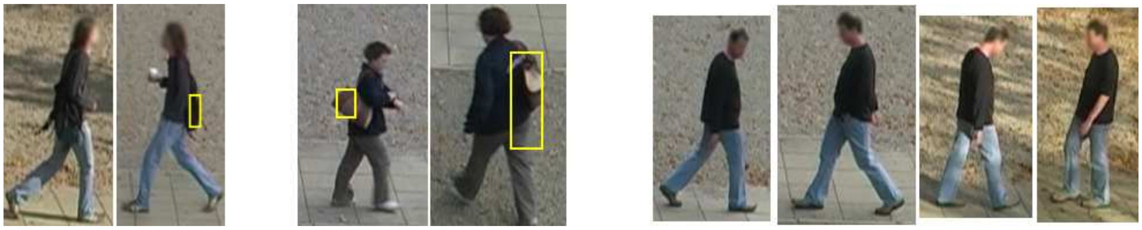


Figure 6.18: Correctly paired sequences when global explanations are considered.

6.6 Conclusion

This chapter has presented a second case study using the framework presented in Chapters 3 and 4. The *Enter-Exit* problem, introduced in this chapter, is formally defined using an AMG. All attributes, rules and constraints enable parsing the detections of people and carried objects into global explanations. The detectors retrieve a multiset of terminals along with values assigned to the synthetic attributes defined in the grammar. The framework introduced in Chapter 3 is used to transform the AMG, for a multiset of detections, into a Bayesian network.

Tested on 12 hours of recorded video, the framework enables finding global explanations that relate people tracked around a building entrance. The global explanation tracks people, along with any objects they might be carrying, in and out of the building. This problem demonstrates the generality of the framework and further supports the case for searching the solution space using RJMCMC-SA. Results indicate MHT does not scale well and IP linear solvers could not cope with the increase in the number of constraints.

When compared to ground-truth data, the *Enter-Exit* global explanation achieves a recall of around 30%, yet a precision of only 12%. This is because the features used to link events are weakly discriminative. People tracked in and out of the building cannot be linked by matching their projected height and clothing colour alone. A high number of false links originate from people of similar clothing colour and height. Decreasing the priors would increase the precision yet decrease the recall. Other features like gait [63, 110], spatial histograms [151], build and skin tone [141] or clothing description [26] can be incorporated. When people are not expected to leave their carried objects behind, carried objects can assist the matching of individuals as demonstrated in this chapter's results.

A different variation of the *Enter-Exit* problem is to distinguish groups of people walking together using a global explanation. An AMG would then parse both individual and group trajectories as detections. The explanation would try to distinguish group trajectories and link them to subsequent appearances of the group, or separate appearances of its individuals. A global explanation can be used to disambiguate the uncertainty in the number of people within each tracked blob. If two people were observed entering a building together and yet each of them left alone, the global explanation can provide a more reliable estimate of the number of people in the enter event, and probably segment the blobs into separate individuals. This application is an interesting one that could benefit from pursuing global explanations, and is mentioned here to motivate other scholars.

Chapter 7

Detecting Carried Objects in Short Video Sequences

The detection of carried objects is a potentially important objective for many security applications of computer vision. However, the task is inherently difficult due to the wide range of objects that can be carried by a person, and the different ways in which they can be carried. This makes it hard to build a detector for carried objects based on their appearance in isolation or jointly with the carrying individual. An alternative approach is to look for irregularities in the silhouette of a person, suggesting they could be carrying something. This is the approach that the method presented in this chapter adopts, and whilst there are other factors that may give rise to irregularities, such as clothing and build, experiments on a standard dataset are promising.

The detector assumes a static background and starts by averaging aligned foreground regions of a walking pedestrian to produce a representation of motion and shape (known as a *temporal template*) that has some immunity to noise in foreground segmentations and phase of the walking cycle. This representation, introduced in [34], was used in [64, 66] for the same application. The temporal template is then matched against a pre-compiled exemplar temporal template of an unencumbered pedestrian viewed from the same direction. Protrusions from the exemplar are detected as candidate pixels for carried objects. Finally, prior information about the expected locations of carried objects is incorporated together with a spatial continuity assumption in order to improve the segmentation of pixels representing the carried objects. Figure 7.1 summarises, along with an example, the

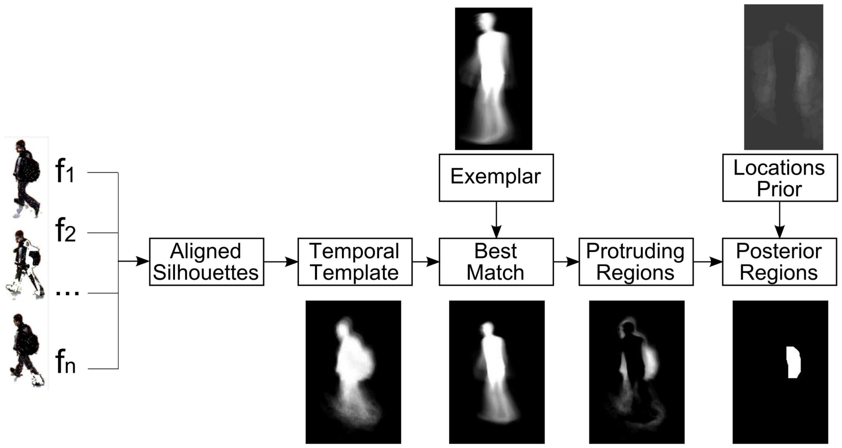


Figure 7.1: All the frames across the sequence are first aligned. The temporal template represents the frequency of each aligned pixel being part of the foreground. The exemplar temporal template from a similar viewing angle is transformed (translation, scaling and rotation) to best match the generated temporal template. By comparing the temporal template to the best match, protruding regions are identified. MRF with a trained map of prior locations is used to decide on the exact pixels representing carried objects.

process of detecting carried objects.

Section 7.1 reviews previous work on the detection of carried objects. Section 7.2 presents the new method, based on matching temporal templates. It studies the pros and cons of using periodicity analysis to classify protrusions, and then incorporates locational priors and a spatial continuity assumption for segmenting carried objects. Experiments comparing the performance with the earlier work from Haritaoglu *et al.* on the PETS2006 dataset [44] are presented in Section 7.3, along with a qualitative discussion on applying the results to the *Enter-Exit* problem dataset. The chapter concludes with an overall discussion.

7.1 Previous work

Several previous methods have been proposed for detecting whether an individual is carrying an object. The *Backpack* [64, 66] system detects the presence of carried objects from short video sequences of pedestrians (typically lasting a few seconds) by assuming the pedestrian's silhouette is symmetric, and that people exhibit periodic motion while moving unencumbered. Foreground segmentations are aligned using edge correlation. The aligned foreground masks are combined into the temporal template that records the proportion of frames in the video sequence in which each pixel was segmented within the foreground. Next, symmetry analysis is performed. The principal axis is computed using principal component analysis of two-dimensional locations, and is constrained to pass through the median coordinate in the vertical and horizontal directions. For each location *x*, relative to the median of the blob, asymmetry is detected by reflecting the point in the principal axis (Figure 7.2). The proportion of frames in which each location was classified as asymmetric is calculated. Consistent asymmetric locations are grouped into connected components representing candidate blobs.

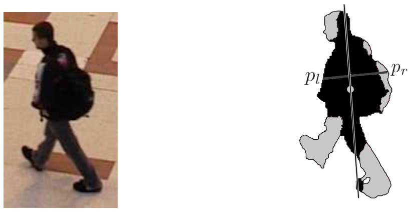


Figure 7.2: For each foreground segmentation, the principal axis is found and is constrained to pass through the median coordinate of the foreground segmentation. Light grey represents the asymmetric regions.

Backpack then distinguishes between blobs representing carried objects and those being parts of limbs by analysing the periodicity of the horizontal projection histograms. The periodicity analysis calculates the periodic frequency of the full body, and that of each asymmetric region. *Backpack* assumes the frequency of an asymmetric blob that represents a limb is numerically comparable to that of the full body. Otherwise, it is believed to be a carried object. Figure 7.3 reviews the process using an re-implementation of *B*ackpack based on their published work.

From the re-implementation, errors in the *Backpack* method arise from four sources. Firstly, the asymmetric assumption is frequently violated. Secondly, the position of the principal axis is often displaced by the presence of the carried object. It should be men-

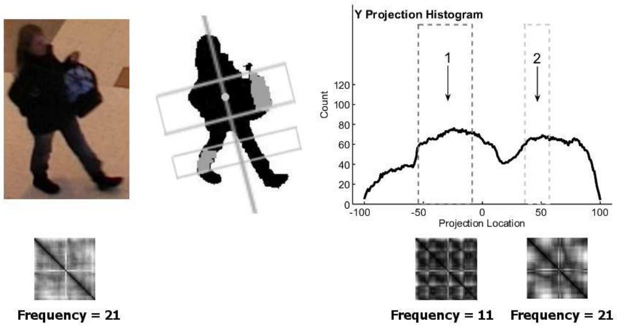


Figure 7.3: Light grey represents the two detected asymmetric regions. Asymmetric regions are projected onto the horizontal projection histogram. Periodicity analysis is performed for the full histogram [Freq = 21] and for regions 1 [Freq = 11] and 2 [Freq = 21]. As region 2 has the same frequency as the full body, it is not considered a carried object.

tioned that there are other methods to position the major axis, like forcing it to pass through the centroid of the head [65] or the ground point of the person walking [70]. Thirdly, accurate periodicity analysis requires a sufficient number of walking cycles to successfully retrieve the frequency of the gait. Fourthly, the periodicity of the horizontal projection histogram does not necessarily reflect the gait's periodicity.

Later work by Benabdelkader and Davis [12] expanded the work of Haritaoglu *et al.* by dividing the person's body horizontally into three slices. The periodicity and amplitude of the time series along each slice are studied to detect deviations from the 'natural' walking person and locate the vertical position of the carried object. They verified that the main limitation in Haritaoglu *et al.*'s method is the sensitivity of the axis of symmetry to noise, as well as to the location and size of the carried object(s).

The work of Lee and Elgammal also uses silhouettes for predicting the locations of carried object and segmenting them on per-frame basis [93]. The training process finds a low-dimensional representation of the kinematics manifold given the joint angles in three dimensional space. For each silhouette, an iterative process finds the best match of the pose, the viewpoint and the shape. The iterative process fills the holes in the foreground segmentation to find better matches, as matching relies on aligning the centres of gravity of the shape and the foreground region. Carried objects are then defined as the unmatched pixels in the foreground region. The approach does not assume spatially continuous carried object pixels. Similar to the method presented in this chapter, this work only detects protruding carried objects, yet is sensitive to foreground segmentation errors as it does not use temporal templates [12, 66]. The approach was only qualitatively analysed.

Branca *et al.* [22] try to identify intruders in archaeological sites. Intruders are defined as those carrying objects such as a probe or a tin. It assumes a person is detected and segmented. Their approach thus tries to detect such carried objects within the segmented foreground region. Detection is based on wavelet decomposition, and the classification uses a supervised three layer neural network, trained on examples of probes and tins in foreground segmentations.

Differentiating people carrying objects, without locating the carried object, has also been studied. One example is the work by Nanda *et al.* [107]. Supervised learning was accomplished based on examples of unencumbered pedestrians and outliers. Outliers are "unusual-looking pedestrians... caused by wearing a hat or carrying an object". A three-layer neural network was used for classification. This work's performance depends on the presence of a similar object within the same viewpoint in the training data.

Alternatively, the work of Tao *et al.* [137] tries to detect pedestrians carrying heavy objects by performing gait analysis. The task was performed using general tensor discriminant analysis, and was tested on the USF HumanID gait analysis dataset.

Recent work by Ghanem and Davis [51] detects abandoned baggage by comparing the person before approaching a region of interest and after leaving it. Carried objects are detected by comparing the temporal templates (the term ‘occupancy map’ is used in their work to reference the same concept) and colour histograms of the ‘before’ and ‘after’ sequences. The approach assumes the person is detected twice, and that the trajectory of the person before approaching the region of interest and after departing are always correctly connected. It also assumes all observed individuals follow the same path, and thus uses two static cameras to record similar viewpoints.

Similarly, Chuang *et al.*’s recent work assumes the person is seen with and without the bag [27]. The ratio of the colour histograms between consecutive frames is used to detect the change in colour components and thus the presence or removal of an object. By observing people coming in close proximities, the work aims to detect the exchange of carried baggage, which signifies suspicious events like thefts. The assumption of observing the person before and after the change in carrying status is application-specific and cannot be used as a general carried object detector.

The novel method, described in Section 7.2, also uses the temporal template but differs from earlier work [51, 64] in matching the generated temporal template against an exemplar temporal template generated offline from a 3D model of a walking person. Several exemplars, corresponding to different views of a walking person, are generated from reusable silhouettes. The temporal templates provide better immunity to noise in foreground segmentations, and enable matching each sequence only once to the exemplar. The new approach does not require the pedestrian to be detected with and without the carried object, and can handle different viewpoints. It detects any type of carried object (not merely backpacks), and can be considered a general approach to detecting protrusions from other deformable tracked objects.

7.2 Description of the method

The method starts by creating the temporal template from a sequence of tracked pedestrians as proposed by Haritaoglu *et al.* [66]. The foreground segmentations at each frame are often noisy due to shadows and camouflage. The temporal template is created by aligning and then averaging the foreground segmentations. Figure 7.4 shows a set of foreground segmentations and their corresponding temporal template. To align the segmentations, Haritaoglu *et al.* suggested an edge correlation with a 5×3 search window. To avoid a predefined displacement window, Iterative Closest Point (ICP) is applied, instead of

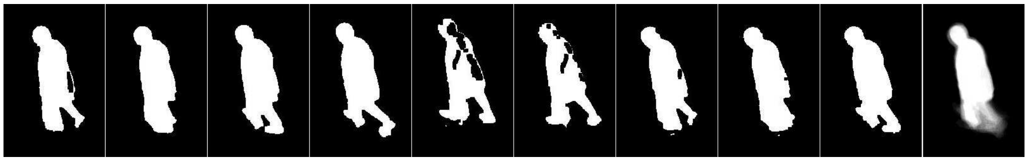


Figure 7.4: Foreground segmentations along with the created temporal template.

edge correlation, to align successive boundaries. The ICP algorithm aligns two clouds of points. It finds the closest match for each point and estimates the least square error transformation. The calculated transformation (translation, rotation and scaling) is applied, and the procedure is iterated until the error falls below a threshold or the maximum number of iterations is reached [15]. ICP is performed on the edge points of the traced boundary around the foreground segmentation. Experimentally, it gives a more accurate alignment in the presence of shape variations between consecutive frames (Figure 7.5). While the original method averages all aligned silhouettes [66], an additional step is introduced to further decrease the noise in the temporal templates. L_1 ranks the frames by their similarity to the generated temporal template. The highest ranked p% of the frames are used to re-calculate a more stable template. p is set to 80 in the results shown below. The more expensive Least Median of Squares (LMedS) estimator [122] gave similar results.

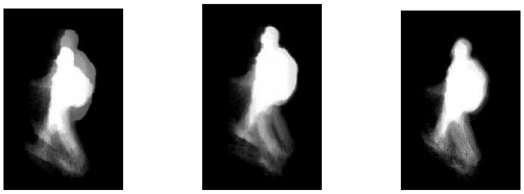


Figure 7.5: Edge correlation temporal template within 15×15 *(left) and* 30×30 *(middle) displacement windows. ICP model (right) does not require any parameters.*

Having derived a temporal template from a tracked pedestrian, one of eight exemplars are used to identify protrusions by matching. These exemplar temporal templates represent a walking unencumbered pedestrian viewed from different directions. A set of exemplars for eight viewing directions was created using the dataset of silhouettes gathered at the Swiss Federal Institute of Technology (EPFL) [37]. The dataset is collected from 8 people (5 men and 3 women) walking at different speeds on a treadmill. Their motion was captured using eight cameras and mapped onto a 3D Maya model (Figure 7.6). The dataset is comprised of all the silhouettes of the mapped Maya model, and has previously been used for pose detection, 3D reconstruction and gait recognition [37, 47]. The temporal templates of different individuals in this dataset are averaged to create the exemplar for each camera view. The eight exemplars (Figure 7.7) are used for detecting the areas representing the pedestrian. The unmatched regions are expected to correspond to carried object(s).

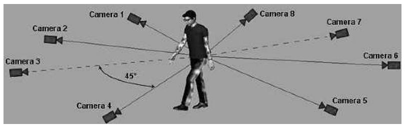


Figure 7.6: Eight cameras for capturing the Silhouettes at EPFL. Diagram from [37]

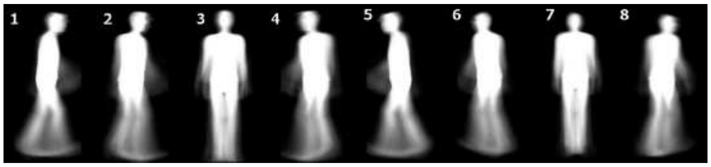


Figure 7.7: Eight exemplar temporal templates, created to represent eight viewpoints.

To decide on which exemplar to use, a homography is estimated from the image plane to a coordinate frame on the ground-plane. This allows estimation of the position and direction of motion of each pedestrian on the ground. The point on the ground-plane directly below the camera is estimated from the vertical vanishing point. The angle between the line connecting this point to the pedestrian and the direction of the pedestrian's motion gives the viewing direction, assuming the pedestrian is facing their direction of motion. This ignores the elevation of the camera above the ground to avoid generating new exemplars for different elevations, although this approximation may be unnecessary since generating the prototypes is fast and need only be done once. The mean of the computed viewing directions over the short video sequence is used to select the corresponding exemplar. Diagonal views (2,4,6,8) are used to match a wider range of angles (60°) in comparison to frontal views. This is because the silhouettes change more drastically near frontal views.

The chosen exemplar is first scaled so that its height is the same as that of the generated temporal template. The median coordinate of the temporal template is aligned with that of the corresponding exemplar. An exhaustive search is then performed for the best match over a range of transformations. In the results, the chosen ranges for scales, rotations and translations are [0.75:0.05:1.25], [-15:5:15] and [-30:3:30] respectively. The cost of matching two templates is an L_1 measure, linearly weighted by the y coordinate of each pixel (plus a constant offset), giving higher weight to the head and shoulders region. Equation 7.1 represents the cost of matching a transformed model (M_T) to the person's temporal template (P), where h represents the height of the matched matrices.

$$d(M_T,P) = \sum_{x,y} |M_T(x,y) - P(x,y)|(2h-y) \tag{7.1}$$

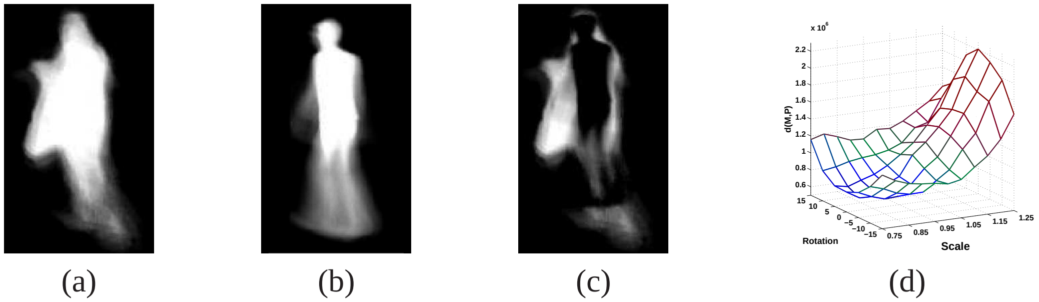
The best match $\widehat{M_T}$ is the one that minimises the matching cost

$$\widehat{M_T} = \operatorname*{argmin}_{T} d(M_T, P) \tag{7.2}$$

Figure 7.8 shows an example of such a match and the located global minimum. The best match $\widehat{M_T}$ is then used to identify areas protruding from the temporal template:

$$\text{protruding}(x,y) = \max(0, P(x,y) - \widehat{M_T}(x,y)) \tag{7.3}$$

Pixels where $P(x,y) < \widehat{M_T}(x,y)$ are assumed to have been caused by noise, or poor foreground segmentation. For the initial results in Section 7.3, the protruding values are thresholded and grouped into connected components representing candidate segmentations of carried objects. Another threshold limits the minimum area of accepted connected components to remove very small blobs. An enhanced approach, not constrained by selecting thresholds, is presented in Section 7.2.2 where segmentation is achieved using binary-labeled MRF formulation, combining prior information and spatial continuity.



(a) (b) (c) (d)

Figure 7.8: The temporal template of the person (a) is matched to the corresponding exemplar (b), the global minimum (d) results in a map of protruding pixels (c). In (d), the best translation for each scale and rotation is only shown.

7.2.1 Periodicity analysis

Periodicity analysis was proposed by Haritaoglo *et al.* to distinguish carried objects from other asymmetric regions. This section is devoted to explaining periodicity analysis, as results demonstrate improved performance when periodicity analysis is used classify protrusions. The algorithm for periodicity analysis described here is based on the original work by Cutler and Davis [31,32]. This is because the method presented in [64] to find the periodicity from horizontal projection histograms lacks mathematical justification when compared to the work of Cutler and Davis.

After aligning foreground segmentations using ICP, L_1 is used to compare two foreground segmentations. Figure 7.9 (a) shows the similarity matrix (S) where darker cells indicate higher similarity. The contrast in the similarity image is sometimes not so clear. Thus an adaptive histogram equalisation is used to enhance the contrast within the image. This contrast-enhancement technique is added to the original Cutler and Davis technique as it improves performance for noisy foreground segmentations.

Next, the similarity matrix (S) was converted to an autocorrelation matrix (A) using Equation 7.4 from [32]. The size of the autocorrelation matrix depends on the autocorrelation region R around each point in the similarity matrix.

$$A(d_x, d_y) = \frac{\sum\limits_{(x,y)\in R} (V(x,y) - V(x+dx, y+dy))}{\sqrt{\sum\limits_{(x,y)\in R} V(x,y)^2 \sum\limits_{(x,y)\in R} V(x+dx, y+dy)^2}} \tag{7.4}$$

In Equation 7.4, $V(x,y) = S(x,y) - \overline{S_R(x,y)}$ where S_R is the region of size R centred around (x,y). The function V subtracts the mean of the values in region R centred at (x,y) from the similarity value $S(x,y)$.

After obtaining the autocorrelation image, 45° square lattices are used to find the dominant frequency. For a range of possible frequencies $d \in [minFreq, maxFreq]$, square lattices are compared to the autocorrelation matrix to find the autocorrelation matrix's frequency. The L_1 measure between the autocorrelation image and a square lattice of frequency d is normalised (i.e. divided by the number of points in the lattice). The lattice with the minimum normalised L_1 measure is selected as the dominant frequency. If multiple minima are found, the smallest frequency is considered as the image's frequency. Figure 7.9 presents an example of how the dominant frequency is found.

In addition to the periodicity analysis performed for the full body, a similar analysis is performed for each protruding region. The foreground images are masked by the detected protrusion region, and the masked foreground images are re-analysed for periodicity. The periodicity analysis though requires a sufficient number of cycles to produce accurate autocorrelation images. The baggage detector presented in this chapter relies on short video sequences, as the person is not expected to change the walking direction within the sequence. Short sequences often fail to show any detectable periodicity. By implementing the periodicity analysis, only 35% of the retrieved protrusions showed any detectable periodic motion.

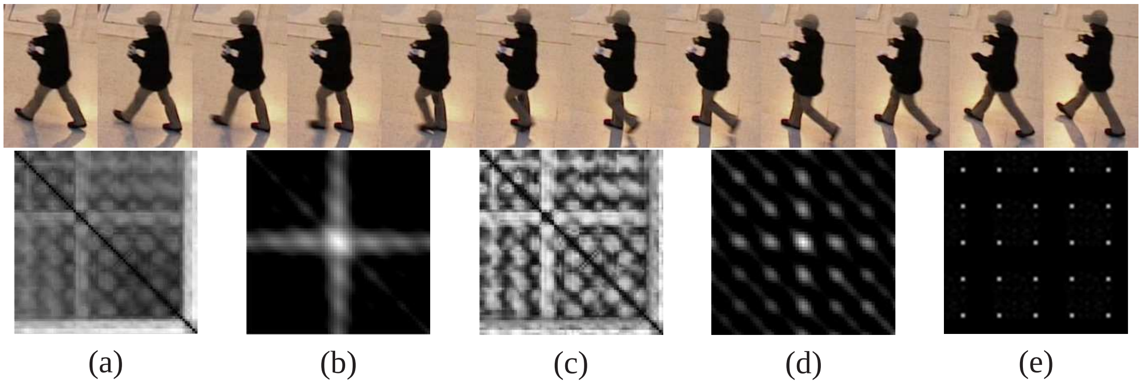


Figure 7.9: The sequence on top shows 12 frames of a sequence representing half a walking cycle. The frequency (f=12) is found using periodicity analysis. First, the similarity matrix (s) is calculated (a). When (a) is directly converted to an autocorrelation image (b), the periodicity is not obvious. Adaptive histogram equalisation is applied to (a) to generate a contrast enhanced image (c). The resultant autocorrelation image (d) would then show clear periodicity, and the chosen square lattice (e) represents the correct frequency (f=12).

7.2.2 Using prior information and assuming spatial continuity

The protruding connected components can be at locations where carried objects are not expected like hats on top of heads. Training for carried object locations relative to the person's silhouette can better differentiate carried objects from other protrusions. This could also be considered a labelling problem that benefits from assuming spatial continuity amongst neighboring locations.

Training is used to generate a map of prior locations Θ_d for each viewpoint d. Prior information for each location is calculated by the frequency of its occurrence within a correctly-detected carried object across the training set. Training values are also used to estimate the distribution of protrusion values conditioned on their labelling. Finally, this information is combined into a Markov Random Field (MRF), determining an energy function which is minimised.

Training for carried object locations is accomplished by mapping the temporal template, using the inverse of the best transformation, to align to its corresponding exemplar. Each location x within the person's temporal template has to be labeled as belonging to a carried object ($m_x = 1$) or not ($m_x = 0$). Using the raw protrusion values $v = \text{protruding}(x)$ calculated in Equation 7.3, the class-conditional densities $p(v|m_x = 1)$ and $p(v|m_x = 0)$ are modeled based on training data. The energy function to be minimised $\mathrm{E}(m)$ over Image I is given by Equation 7.5.

$$\mathrm{E}(m) = \sum_{x \in I} \Big(\phi(v|m_x) + \omega(m_x|\Theta) \Big) + \sum_{(x,y) \in \mathscr{C}} \psi(m_x, m_y) \tag{7.5}$$

$\phi(v|m_x)$ represents the cost of assigning a label to the location x based on its protrusion value v in the image:

$$\phi(v|m_x) = \begin{cases} -\log(p(v|m_x = 1)) & \text{if } m_x = 1 \\ -\log(p(v|m_x = 0)) & \text{if } m_x = 0 \end{cases} \tag{7.6}$$

$\omega(m_x|\Theta)$ is based on the map of prior probabilities Θ given a specified walking direction:

$$\omega(m_x|\Theta) = \begin{cases} -\log(p(x|\Theta)) & \text{if } m_x = 1 \\ -\log(1 - p(x|\Theta)) & \text{if } m_x = 0 \end{cases} \tag{7.7}$$

The interaction potential ψ follows the Ising model over the cliques, where $\mathscr{C}$ represents all the pairs of neighboring locations in the image I:

$$\psi(m_x, m_y) = \begin{cases} \lambda & \text{if } m_x \neq m_y \\ 0 & \text{if } m_x = m_y \end{cases} \tag{7.8}$$

The interaction potential ψ is fixed regardless of the difference in protrusion values v at locations x and y. A data-dependent term was not chosen because the protrusion values represent the temporal continuity, and not the colour or texture information.

7.3 Experiments and results

This section presents results on two datasets. First a thorough evaluation on the publicly available PETS2006 dataset is presented. The ground truth for carried objects was manually obtained, thus a quantitative and qualitative analysis is provided for this dataset. Next, the trained priors from PETS2006 are used to detect carried objects in the video sequence used for the *Enter-Exit* problem. A qualitative discussion of the results is presented.

7.3.1 PETS2006

The third camera of the PETS2006 dataset is selected, as there is a greater number of people seen from the side. Side-views usually result in the carried objects protruding from the silhouette. The ground-plane homography was established using the ground truth measurements provided as part of the dataset. Moving objects were detected and tracked using the same tracker [100] to retrieve foreground segmentations. The tracker's shadow remover worked reasonably well on the dataset. Trajectories shorter than 10 frames in length were discarded. As this method cannot deal with groups of people tracked together, such trajectories were also manually removed. The carried objects in the dataset varied between boxes, hand bags, briefcases and suitcases. Unusual objects are also present like a guitar in one example. In some cases, people were carrying more than one object. The number of individually tracked people was 106. Ground truth for carried objects was obtained manually for all 106 individuals. 83 carried objects were tracked, and the bounding box of each was recorded for each frame (Figure 7.10). Bounding boxes were chosen instead of pixel masks for simplicity.

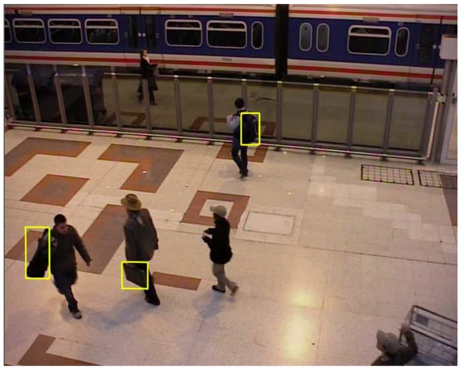


Figure 7.10: PETS2006 Third camera viewpoint showing ground truth bounding boxes representing carried objects.

The results compare the re-implementation of *Backpack* as specified in their papers [64, 66] with the proposed method (Section 7.2). To ensure fair comparison, the same temporal templates are used as the input for both methods. A detection is labeled as true if the overlap between the bounding box of the predicted carried object (b_p) and that of the ground truth (b_{gt}) exceeds 15% in more than 50% of the frames in the sequence. The measure of overlap criterion is defined by Equation 7.9 [41]:

$$\text{overlap}(b_p, b_{gt}) = \frac{\text{area}(b_p \cap b_{gt})}{\text{area}(b_p \cup b_{gt})} \tag{7.9}$$

A low overlap threshold is chosen because the ground truth bounding boxes enclose the whole carried object, while both methods only detect the parts of the object that do not overlap the body. Multiple detections of the same object are counted as false positives.

The results are first compared without periodicity analysis (Explained in Section 7.2.1). Each of the two algorithms has two parameters to tune, one for thresholding and one for the minimum size of the accepted connected component. Precision-Recall (PR) curves for the two methods are shown in Figure 7.11 (left). These were generated by linearly interpolating the points representing the maximum precision for each recall. They show

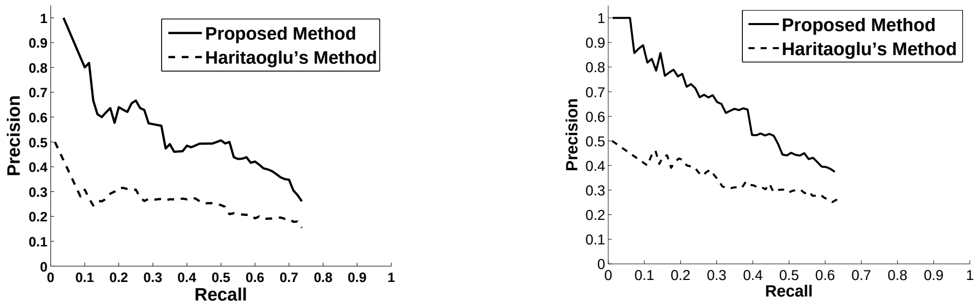


Figure 7.11: PR curves for the proposed method compared to Haritaoglu et al.'s method without (left) and with (right) periodicity analysis to classify the retrieved blobs.

a substantial improvement in performance for the proposed method. Maximum precision on a recall of 0.5, for example, was improved from 0.25 using asymmetry to 0.51 using matching. Maximum recall was 0.74 for both techniques, as noisy temporal templates and non-protruding carried objects affect both techniques. Figure 7.12 shows some examples comparing asymmetry analysis with matching temporal templates.

To further compare the methods, the results after performing periodicity analysis are compared. To achieve that, all optimal setting points along the curves in Figure 7.11 (left) are used, and the two thresholds for periodicity analysis are varied. These are for the minimum confidence for periodicity and the threshold for the difference in periodicity. Figure 7.11 (right) shows PR curves analogous to those in Figure 7.11 (left) but now including periodicity analysis, again taking the maximum precision for each recall. The improved performance of the matching method is still apparent. In addition, comparing the corresponding curves shows that periodicity analysis helps improving the performance for both methods.

Next, spatial continuity is assumed along with trained priors. Results are presented along with a discussion of the advantages of training for prior locations. The pedestrians in the dataset were divided into two sets, the first containing 56 pedestrians (Sets 1-4 in PETS2006) and the second containing 50 pedestrians (Sets 5-7). Two-fold cross validation was used to detect carried objects.

During training, connected components are obtained using a threshold of 0.5. Correct detections, by comparing to bounding boxes from the ground truth, are used to train for locations of carried objects separately for each directionally-specific exemplar. To make use of the small training set, maps of opposite exemplars are combined. For example, the first and the fifth exemplars are separated by 180°. Θ_1 and Θ_5 are thus combined by horizontally flipping one and calculating the weighted average $\Theta_{1,5}$ (by the number of blobs). The same applies for $\Theta_{2,6}$, $\Theta_{3,7}$ and $\Theta_{4,8}$. Figure 7.13 shows $\Theta_{2,6}$ using the two

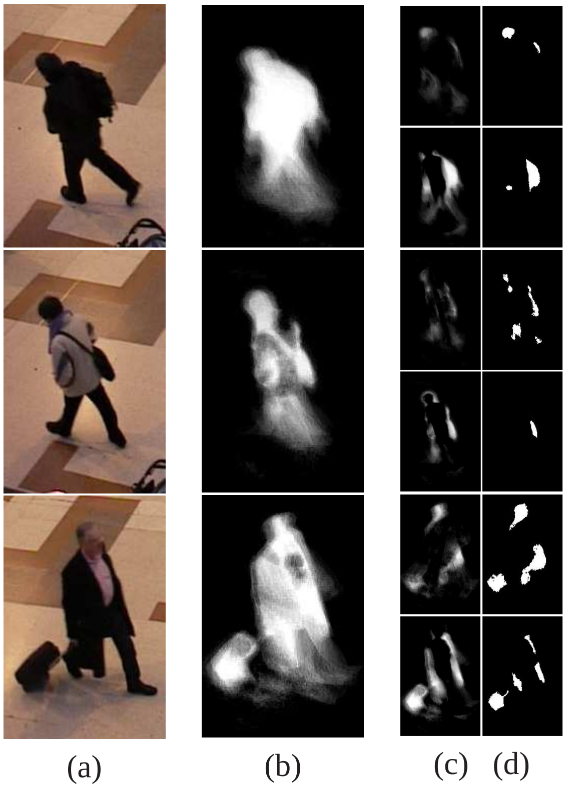


Figure 7.12: Three examples (a), along with their temporal templates (b) are assessed using both techniques. The asymmetric regions (c-top) thresholded (d-top) and the protruding regions (c-bottom) thresholded (d-bottom) show some examples of how template matching retrieves better estimate of the carried objects.

disjoint training sets.

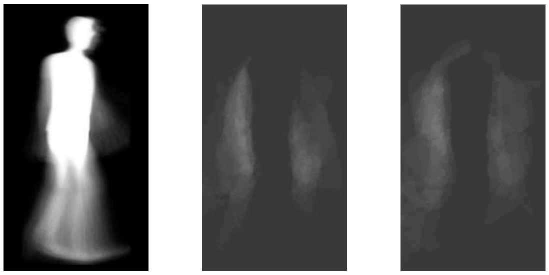


Figure 7.13: For the second exemplar (left), $\Theta_{2,6}$(middle) was generated using sets 1-4, and $\Theta_{2,6}$(right) was generated using sets 5-7. The location model Θ has high values where stronger evidence of carried objects had been seen in training. A prior of 0.2 was used when no bags were seen. By symmetry, Θ_6 is a horizontal flip.

Figure 7.14 presents the distribution of protrusion values for carried objects ($m_x = 1$) and other protrusions ($m_x = 0$). By studying these density distributions, $p(v|m_x = 1)$ was approximated by two Gaussian distributions, one for stable carried objects, and another for swinging objects. The parameters of the two Gaussians were manually chosen to

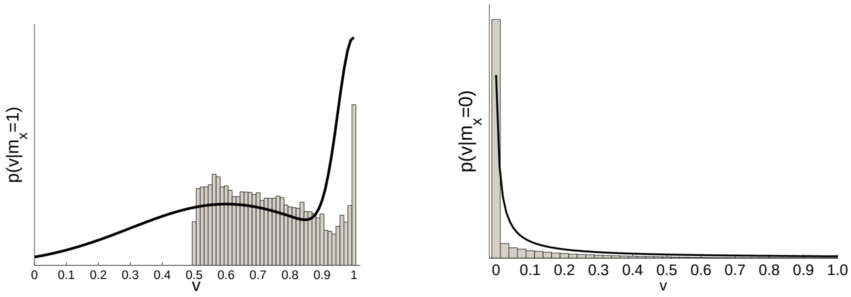


Figure 7.14: Pixel values distribution for objects (left) and non-objects (right) protruding pixels. Thresholded pixels (>0.5) that match true detections are compared to ground truth, then are used to train $p(v|m_x = 1)$. The rest are used to train $p(v|m_x = 0)$.

approximately fit the training density distributions.

$$p(v|m_x = 1) = \gamma\mathscr{N}(v; 0.6, 0.3) + (1-\gamma)\mathscr{N}(v; 1.0, 0.05) \tag{7.10}$$

γ is the relative weight of the first Gaussian in the training set. Its value resulted to be 0.64 for the first training set, and 0.66 for the second disjoint set. The density distribution $p(v|m_x = 0)$ resembles a reciprocal function. It was thus modeled as:

$$p(v|m_x = 0) = \frac{1/(v+\beta)}{\log(1+\beta) - \log(\beta)} \tag{7.11}$$

β was set to 0.01. The denominator represents the area under the curve for normalisation.

The max-flow algorithm, proposed in [21], and its publically available implementation, minimises the energy function (Equation 7.5) retrieving regions representing carried objects. The smoothness cost term λ was optimised based on the used training set. In order to compare the MRF formulation with simple thresholding, the parameters are optimised on each training dataset and tested on the other. For MRF, λ was optimised on the training datasets resulting in 2.2 and 2.5 respectively. Table 7.1 presents the precision and recall results along with the actual counts combined for the two test datasets, showing that MRF produces higher precision and recall results.

	Precision	Recall	TP	FP	FN
Thresholding	39.8%	49.4%	41	62	42
MRF - Prior	50.5%	55.4%	46	45	37

Table 7.1: Better performance was achieved by introducing the MRF representation.

To evaluate the effect of introducing location models, the term $\omega(m_x|\Theta)$ was removed from the energy function and the results were re-calculated. λ was varied between

[0.1:0.1:6] to produce the PR curves in Figure 7.15 that demonstrate the advantage of introducing location prior models. Examples in Figure 7.16 show how prior models affect estimating carried objects.

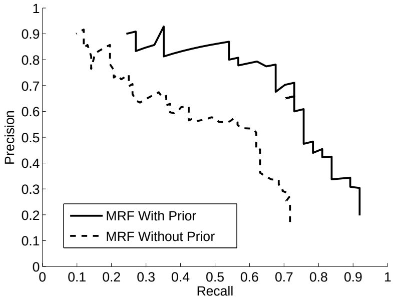


Figure 7.15: PR Curves for detecting carried objects using MRF. Introducing location maps to encode prior information about carried object locations produces better performance.

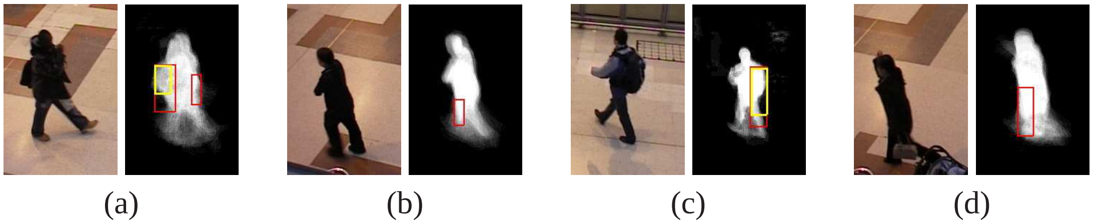


Figure 7.16: The yellow rectangles show the choice of carried objects using MRF with location models. Red rectangles refer to MRF without location models. Prior information drops candidate blobs at improbable locations (a,b), and better segments the object (a,c). It nevertheless decreases support for carried objects in unusual locations (d).

Quantitatively, for the 45 false positive, and 37 false negative cases, Figure 7.17 dissects these results according to the reason of their occurrence. Figure 7.18 presents a collection of results highlighting reasons of success and main sources of failure.

7.3.2 LEEDS 2009

This section details how the baggage detector was run on a different dataset, which has been used to test the global explanation for the *Enter-Exit* problem (Chapter 6). The dataset consists of a full working day (12 hours of recording). The tracker retrieved only the set of trajectories that passed through the interesting zone (marked with a grey rectangle in Figure 7.19, to track people around the building entrance. After manually removing groups of people walking together, 326 trajectories were considered for baggage detection.

The new dataset differs in that a person is tracked for a longer period, and people often change their walking direction. The depth of the viewpoint also introduces a change

Reasons behind FP detections	
Protruding parts of clothing	15
Protruding body parts	10
Extreme body proportions	6
Incorrect template matching	5
Noisy temporal template	5
Duplicate matches	4
Total	45

Reasons behind FN detections	
Bag with little or no protrusion	9
Dragged bag tracked separately by tracker	6
Carried object between legs	5
Carried object not segmented from background	4
Little evidence of prior location in training	3
Swinging small object	3
Noisy template	3
Incorrect template matching	2
Merging two protruding regions into one	2
Total	37

Figure 7.17: Reasons behind False Positive (FP) and False Negative (FN) detections.

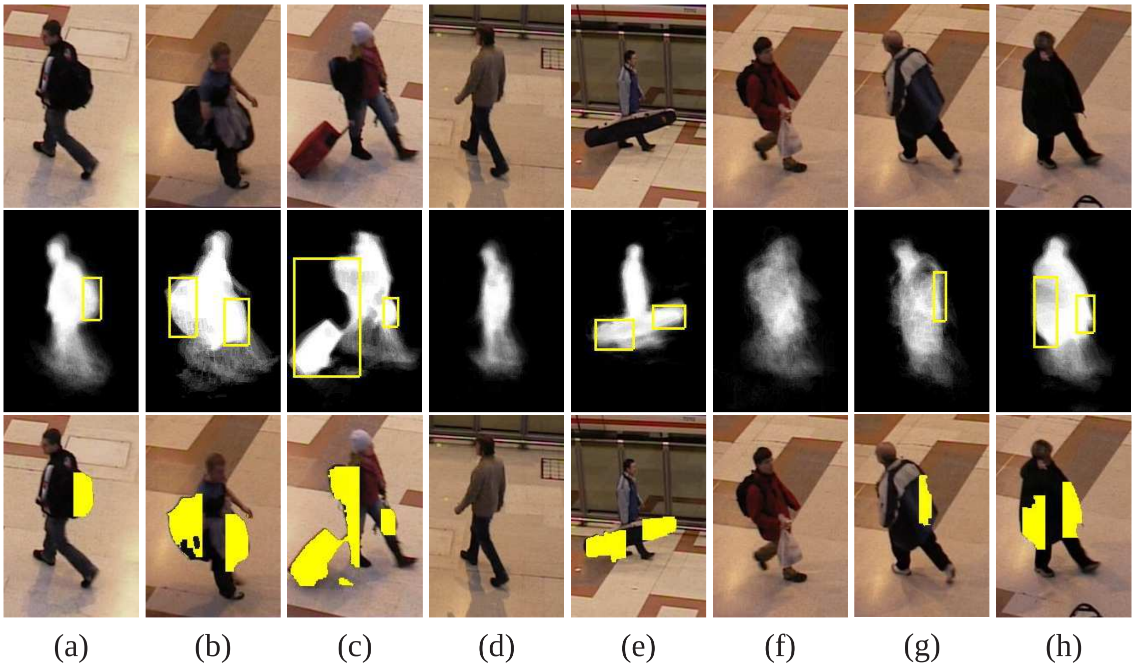


Figure 7.18: The proposed method can identify single (a) or multiple (b,c) carried objects. (d) shows its ability to classify true negative cases. Objects extending over the body are split into two (e). Failure cases may result from poor temporal templates due to poor foreground segmentation (f). The map of prior locations could favor some false positive objects (g). This method is not expected to cope with extreme body proportions (h). The second row shows the detections projected into the temporal templates, and the third row shows detections projected into a single frame of the sequence.

in scale along the trajectory for people walking toward or away from the camera. Thus, each trajectory is partitioned into sequences, each of 50 frames maximum. The temporal template is created separately for each partition. Figure 7.19 shows the dataset's viewpoint along with multiple foreground segmentations for the same trajectory. This trajectory was split into three parts, and Figure 7.20 shows the baggage detection results for one frame from each part. As the baggage detector assumes the bag is protruding from the normal silhouette, different viewpoints give rise to different detections. While the first viewpoint did not detect any protrusions, the second diagonal viewpoint enabled detecting the carried bag, while the third horizontal viewpoint showed both the carried bag and the held jacket as protrusions.

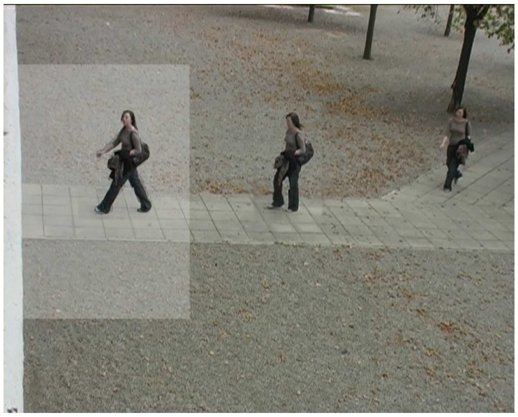


Figure 7.19: The viewpoint for the second baggage dataset showing the different viewpoints.

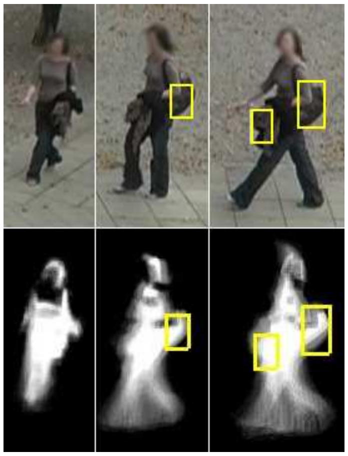


Figure 7.20: A trajectory was split into three sequences. In the first sequence, carried objects were not protruding. In the intermediate one, the carried bag was detected, while both the bag and the jacket were detected from the third viewpoint.

The trajectories are partitioned uniformly regardless of whether the viewpoint, the direction of motion or the scale have changed. Alternatively, a sliding window detector could be established instead of slicing the trajectory. The results presented here did not use a sliding window approach to speed detection. The ground-plane homography was manually obtained, along with finding the vanishing point. The baggage detections for the complete dataset were based on the same location priors trained using the PETS2006 dataset. The encouraging results prove the ability of location priors to be transformed between different camera viewpoints and elevations as they are mapped to the same 3D exemplars.

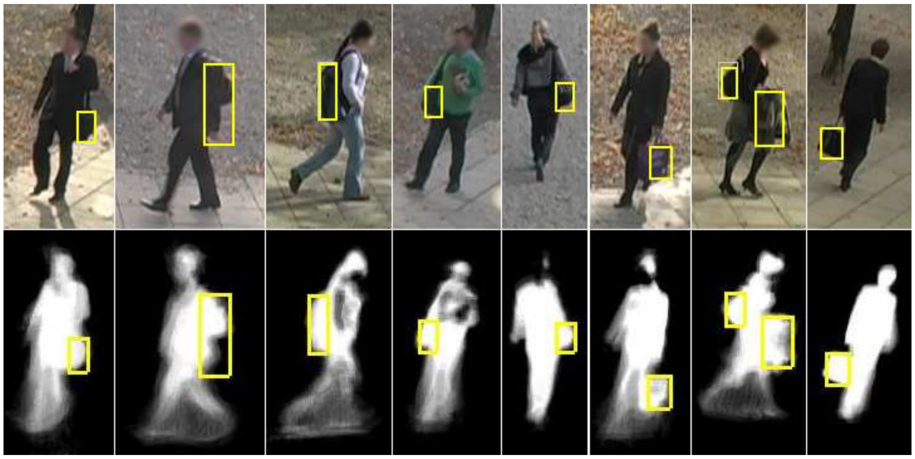


Figure 7.21: LEEDS2009 - a collection of correctly detected bags.

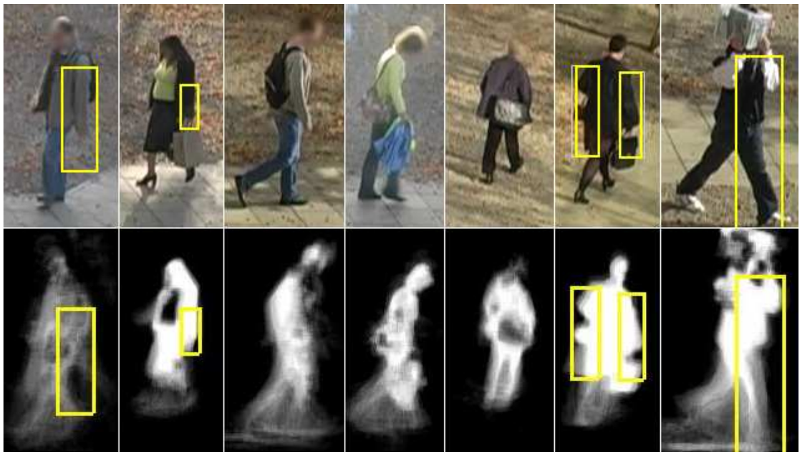


Figure 7.22: LEEDS2009 - a collection of incorrect baggage detections.

This section does not present any quantitative results, as a manual ground truth was not obtained. A selected collection of success and failure baggage detections are shown in Figures 7.21 and 7.22. Figure 7.21 shows 8 trajectories with successful detections. The detections are shown on the temporal template and projected on a single frame in each case. Figure 7.22 shows 7 incorrect detections. They cover a range of cases in which the detector fails. The first case results from poor foreground segmentation. The tracked individual is wearing a jacket which is very similar to the background's colour. Camouflaging results in a noisy temporal template and incorrect detections. The second failure case results from the baggage not being segmented as part of the foreground. The stationary arm holding the bag is detected as the carried object instead. The third, fourth and fifth examples are false negative cases where the carried object is not sufficiently protruding to be detected. Example six successfully detects two objects but the bounding box extends to include the stationary arms carrying the objects as well as a protruding coat. The last case fails in matching the temporal template to the unencumbered model. By reviewing Equation 7.2, the match gives higher weight to matching the head and the shoulders of the model. In this example, the head and the shoulders are occluded by the carried object, which resulted in an incorrect match. This collection of success and failure cases adds to the reader's understanding of the strengths and weaknesses of the baggage detector.

7.4 Conclusion

This chapter proposed a novel method to detect carried objects, aiming at higher robustness than noisy single frame segmentations. Carried objects are assumed to cause protruding regions from the normal silhouette. Like an earlier method, this method uses a temporal template but matches against exemplars rather than assuming that unencumbered pedestrians are symmetric. Evaluated on the PETS2006 dataset, the method achieves a substantial improvement in performance over the previously published method. Training for locations of carried objects and using an MRF to encode spatial constraints results in further improved performance.

The method depends on two assumptions, the first is that a temporal template can be constructed from foreground segmentations, and the second is that carried objects are protruding from the body's silhouettes. Temporal templates sometimes fail to produce adequate results due to poor foreground segmentation and unsegmented shadows. The baggage detector does not currently evaluate the quality of the calculated temporal template prior to matching the template to an unencumbered exemplar. A measure of the temporal template's quality is left for future work.

Due to its dependence on protrusion, the method cannot detect non-protruding carried objects. It may not be able to distinguish carried objects from protruding clothing or non-average build. Future improvements to this method might be achieved using texture templates to assist segmentation based on color information. In addition, the independence assumption in learning prior bag locations could be studied to utilise shapes of previously seen bags in producing better segmentations. When matured, this technique can be embedded into surveillance and security systems that aim at tracking carried objects or detecting abandoned objects in public places.

Chapter 8

Conclusion and Future Work

This thesis proposes a framework for explaining an activity given an input video sequence. The approach uses the natural constraints within the activity to find a consistent set of events that covers all detections. This complete and consistent set of events is 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). AMGs allow specifying attribute rules, as well as constraints that confine the grammar's parses to consistent ones. Each production rule in the grammar rewrites a nonterminal into an un-sequenced collection of simpler events (i.e. a multiset). The composition thus does not enforce any temporal relationships, and those are defined as constraints in the grammar only when needed. Using attribute rules, the features retrieved for each detection can be propagated up the parse tree to evaluate the interactions between objects representing compound events. Priors and conditional probabilities are assigned by expert knowledge. A labeled set of training sequences is used to learn the likelihoods for the selected features.

For each input video, detectors retrieve the set of detections, which represents terminal symbols along with the synthetic attribute values. An algorithm then builds a Bayesian Network (BN) to model the probability distribution over the set of global explanations for these detections. Each possible event, given the detections, is represented by a node in the BN. The set of possible labellings of the BN corresponds to the set of all global explanations. Heuristic search techniques are proposed to find the MAP, as combinatorial search becomes intractable when the complexity and duration of the activity increase.

The framework is tested for two problems, and experimental results are compared. This chapter discusses the ability to generalise the framework to other problems, along with its limitations. For a comprehensive conclusion, a few issues are incorporated into this chapter. Section 8.2 reviews alternative techniques for combining multiple features. Section 8.3 introduces risk management and utility theory, as the best global explanation need not be the MAP solution when used for a specific application. The chapter concludes with suggested future work for interested scholars.

8.1 Generalisation and limitations

Several aspects need to be emphasised to explain the generality as well as the limitations of the proposed framework. First of all, both case studies are defined as binary AMGs, where each nonterminal is rewritten as a multiset of two symbols. The method in Chapter 3 can build the BN structure for any production rule $X_0 \rightarrow X_1, \dots X_{n_p}$, and deals with direct recursion in the production rules. As to the search techniques, greedy, MHT and IP can deal with any Bayesian network, whether it is binary or not. RJMCMC and RJMCMC-SA on the other hand require more move types to deal with non-binary structures, because the proposed set of general moves suit binary event hierarchies. The moves can be extended, yet the same performance cannot be predicted. This is because an increase in the number of move types requires longer chains and more complex proposal distributions.

The generality of the framework can be tested by applying it to different activities. In addition to the two case studies, the thesis proposes other domains where the framework can be applied like car parks, and train platforms. These domains include multiple interleaved unordered events with natural constraints that define the consistent set of events. The domains are structured so the types of expected events are known in advance, and the gathered detections can be explained using the events in the activity's hierarchy. For example, sports games are structured activities that could be defined and recognised using this framework. Some scenarios in public surveillance are also structured like flowing traffic, metro stations and car parks. Similarly, the events one performs at the bank or the post office are also typically structured.

In scenes where the activity consists of a large independent set of possible events, the approach would obviously not show a significant improvement over local analysis. For example, consider the activity in the main hall of a train station. It is challenging to define in advance the possible events, and a person can perform any combination of events, like pausing to make a phone call, waiting, passing through, etc. There are no natural constraints in the relationship between those events which could assist recognition. Global explanations do not promise recognition improvement in this scenario. Moreover, unstructured or unpredictable activities, like chaotic scenarios or anomaly detection, are not suitable for our framework.

Another issue worth discussing is the choice of detectors for recognising the events. Detectors range from very general ambiguous ones to specific noisy detectors. For example, the bicycle-cluster detector used in Chapter 5 is a general detector of change and its detections are ambiguous as they include dropped and picked groups of bicycles. Alternatively, one can design a specialised detector for single dropped bicycles. Such a detector would be less ambiguous but subsequently more noisy. General ambiguous detectors increase the complexity of the global explanation, yet strengthen the power of constraints in disambiguating uncertain detections. Specific noisy detectors, on the other hand, result in simpler global structures. This trade-off is an interesting issue for future research. In this thesis, the detectors are general as the focus of the research is on testing the ability of global explanations to recognise events from ambiguous detections.

To apply the framework to a different activity, the AMG should be defined. Given a set of detections, the Bayesian network structure is built from the AMG. Then, priors and conditional probabilities need to be estimated. This is somehow different from the approach adopted in stochastic grammars. Figure 8.1 shows an example AMG and an equivalent stochastic grammar with prior probability associated with each rule. The posterior probability in the proposed framework is over all possible parse trees, compared to the stochastic grammar approach where the posterior probability depends only on the parsed rules for this parse tree. Both approaches are generative in that explanations can be sampled from the posterior probability distribution.

Using the proposed framework, the required Bayesian network models the probability distribution over all explanations, and is built bottom-up instead of top-down. The advantage of bottom-up is shown when events are shared. The parse of the AMG is not strictly a tree. Figure 8.2 shows an example where the event B is shared by two compound events A_1 and A_2. When this parse tree is evaluated, the probability of the event B should be included only once in the posterior. In the bottom-up BN, this is easily achieved as compound events are dependent on their constituent events. In top-down approaches, a list of already evaluated rules should be maintained by the parser to avoid duplication. It should though be clarified that a stochastic grammar and top-down approaches can be used instead. It is a mirrored version of the approach. Further research is needed to compare which probabilities are easier to define or learn.

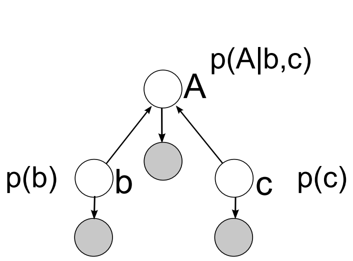
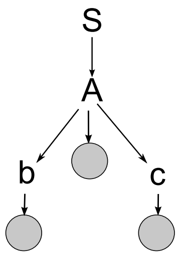
Proposed framework	**Stochastic grammar**	
S → A	S → A	[p_1]
S → b	S → b	[p_2]
S → c	S → c	[1-p_1-p_2]
A → c, b	A → c,b	[1.0]
		

Figure 8.1: Comparison between the proposed framework and stochastic grammar.

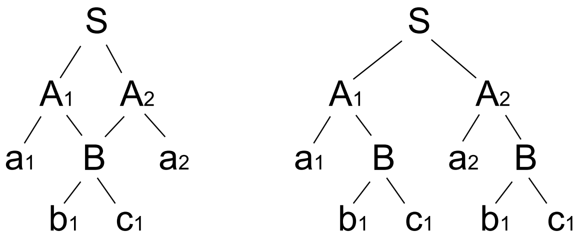


Figure 8.2: When an event is shared (B in this example), the tree is represented by a graph (left), or the sub-tree can be duplicated (right).

Learning the prior and conditional probabilities from training data would certainly facilitate applying the framework to solve other problems. One needs to be careful when learning the probabilities. While the probability associated with each rule in SCFG can be easily estimated from labelled training data, this is not the case with AMG. In SCFG, the weight of the rule $X \rightarrow Y$ is obtained from the ratio of times X is rewritten as Y to the total number of times X has been rewritten in the training data. This is referred to as the Empirical Relative Frequency (ERF) estimates. Abney shows that ERF estimates cannot be used to learn the probabilities for AMG from training data, as ERF estimates do not take into consideration the dependencies in applying the production rules [3]. ERF estimates do not converge to the correct distribution as the training set increases in size. Abney proposes sampling to learn the correct probabilities [3].

8.2 Likelihood of synthetic attributes

Chapter 3 assumed the synthetic attributes are independent and the likelihood is obtained from the product of cpdfs. Some synthetic attributes might be more discriminative than others, and attributes chosen by an expert might fail to produce significant differences due to noise in the measurements. For example, colour proved to be a very ambiguous cue when used in Chapter 6, though it was the obvious attribute to be chosen by the expert.

Instead of treating the attributes separately, boosting would be an efficient way to combine the different classifiers obtained from training the synthetic attribute values, and form a more powerful classifier [16]. Boosting has been successfully applied for combining features for classification [140]. A recent proposed approach is the HybridBoost approach for jointly ranking and classifying detections [95]. Ranking would favour the attribute values that correspond to correct events over the values of other events, while classifying distinguishes the values of correct events from those of incorrect events. The proposed HybridBoost combines Adaboost with RankBoost to learn the parameters for both ranking and classifying jointly.

Moreover, it is worth investigating whether dimensionality reduction techniques can compress the features and generate attribute values that better distinguish the occurrence of events. Though these attributes would not be conceptually meaningful, there is scope for unsupervised feature selection to combine features in a way that may better distinguish event types.

8.3 Decision theory and utility management

Throughout the previous chapters, the *best explanation* is thought to be the one that correctly recognises all the events. Given the uncertainty, searching for the MAP solution tries to decrease the missed or incorrectly recognised events. Often, when such a system is put to use, the objective is more complex than maximising the correctly recognised events [16]. This is well-explained in decision theory.

When used in surveillance, for example, recognising certain events would trigger actions. A reward for the recognition is measured by the client who would be using the system. A utility function u is a numerical measure of this reward. Thus, if one event s_1 is preferred over another s_2 by the client, then $u(s_1) > u(s_2)$ [130]. For example, the reward of catching a theft in the *Bicycles* problem is higher than detecting a bicycle was safely retrieved by its owner. The best explanation, in these terms, is one that maximises the utility of all recognised events. Some authors refer alternatively to a loss function $l(s)$, which represents the loss resulting in misclassifying the event s. The optimal explanation would then try to minimise the loss function [49].

When events are only probabilistically recognised due to uncertainty, decision analysis can be carried out in a Bayesian manner. The objective would then be to maximise the expected utility. The *maximum expected utility principle (MEU)* maximises the sum of the probability of each outcome times the utility of the outcome.

$$\max \sum_{i} p(s_i)u(s_i) \tag{8.1}$$

The recognition then extends beyond finding the MAP, to finding an optimal recognition strategy that maximises the expected utility. The utility is rarely a static function. Often the domain has a ‘finite horizon’, which means the client’s optimal explanation changes with time [49]. For example, the tolerance for abandoned baggage in surveillance changes according to the threat level at that time. Future work can study incorporating utility management in the proposed framework.

Moreover, sensitivity analysis is particularly important in decision making systems. *Sensitivity analysis* checks whether the decision taken is sensitive to small changes in the probabilities and utilities. In this case, the decision might not be safe to take, and the output should at least be labeled accordingly. This can be performed by systematically changing the probability values and evaluating the effect of the change on the decision taken.

8.4 Future directions

The ideas introduced in this thesis can be further expanded along different paths. First and most importantly, using the framework to recognise other activities is the best way to assess its generality or highlight any shortcomings. I intend to compile a toolbox that would enable researchers to define activities using AMG then recognise detections as seen in the given two case studies.

Second, learning the parameters of the BN from unlabeled data would facilitate the framework’s applicability. As previously mentioned, the constraints in the grammar make this learning difficult. This requires further research.

Third, learning the hierarchical structures themselves via mining spatio-temporal relationships is worth investigating. 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 [157], recent advancements in discovering activities using unsupervised learning are promising [61].

Fourth, researchers might wish to expand the carried object detector presented in Chapter 7. Although developed for a specific problem, the detector could be applied to the detection of irregularities in appearance for other categories of object that move in a periodic fashion.

On a wider scale, activity recognition would undoubtedly require less erroneous motion detectors (i.e. trackers) and better colour constancy algorithms. The recognition progress is hindered by these unsolved problems. Given the current ambiguities in the detections, a limit is present on how much can be achieved.

8.5 A final word...

This thesis proposes a method to recognise an activity, based on searching for a consistent set of events that best explains all the detections. It is used in scenarios where the number of possible events performed by each person is limited and can be defined. By satisfying natural constraints, global explanations can resolve local ambiguities and avoid inconsistencies. The thesis is thus a small step further to higher-level understanding of low-level visual detections. In perceiving the visual world, we undoubtedly use our understanding of possible outcomes to explain the detections.

In pursuing this research, I hoped to expand my understanding as well as highlight new ideas that can be investigated further to achieve reliable *computerised vision*, some time in the foreseeable future.

Appendix A

Markov Chain Monte Carlo (MCMC)

Monte Carlo simulation was first introduced by Stan Ulam (1946) as a way to compute the chances that a particular layout of cards would result in a successful solitaire game [8]. Ulam thought of randomly selecting layouts and calculating the chances from the random set. He proved that the chances calculated from a random set approximate the exact chances for 'large-enough' random sets. Monte Carlo simulation became an attractive way of approximating an intractable search space.

Assume π represents a probability distribution, $\pi : R^d \rightarrow R^+ \cup \{0\}$. Any distribution π can be approximated by a sample of size n where the distribution of the sample elements $\pi^\star$ satisfies Equation A.1.

$$\pi = \lim_{n \to \infty} \pi^\star \tag{A.1}$$

Monte Carlo simulation assumes independent and identically-distributed (i.i.d.) samples.

For some distributions, selecting an i.i.d. sample from the distribution is not an easy job to accomplish. When the distribution can be evaluated at any point up to a constant normalising factor, Monte Carlo processes can be substituted with Markov Chain Monte Carlo (MCMC) sampling where choosing a sample element depends on the choice of the previous element along the chain. The Markov chain is a sequence of variables $x_1, x_2, \ldots, x_n$ that represents a sample from the domain. The histogram of those sample elements approximates the proposal distribution for 'large-enough' examples. The probability for selecting the next variable along the chain x_{n+1} is solely based on the last variable added to the chain assuming a first-order Markovian property, $p(x_{n+1}|x_1, x_2, \ldots x_n) = p(x_{n+1}|x_n)$. Despite the dependency, MCMC converges to the invariant distribution that is independent of the starting point. For large n, the distribution of sample elements resembles that of the target distribution.

To define a Markov chain, the set of possible states R^d and the transition probabilities between these states should be specified. The transition probability is referred to as the **proposal distribution** $Q(y|x)$. By definition, the integral of the proposal distribution along the domain equals 1.

$$\int_{R^d} Q(y|x)dy = 1; \tag{A.2}$$

Designing a Markov Chain Monte Carlo sampler thus focuses on the choice of the proposal distribution Q. The next subsection explains how to choose a suitable Q that would converge to the required target distribution.

A.1 Markov chains for finite search space

If the search space is finite, then the proposal distribution Q can be represented by a matrix where the $(x,y)^{th}$ element is equal to $Q(y|x)$. Q is a right stochastic matrix[1] since the sum of elements along the row $\sum\limits_y Q(y|x)$ equals 1.

The Perron-Frobenius theorem states that for any square right stochastic matrix Q, there exists a stochastic vector V (associated with the eigen-value 1), where

$$\lim_{k\to\infty} Q^k = V_j \tag{A.3}$$

Given a Markov chain with a proposal distribution Q, the probability of selecting a state y after k steps given the current state is x equals $Q^k(y|x)$. Thus, according to the theorem, the Markov chain converges at the limit to a proposal distribution that is stationary and independent of the initial state. V defines the stationary distribution (also referred to as the invariant distribution) of the Markov Chain. If the Markov chain is irreducible and aperiodic, the stationary distribution is unique.

When using MCMC for sampling a probability distribution π, one needs to find a suitable transition matrix Q that converges to the required probability distribution $V_y = \pi(y)$. If the matrix satisfies the **detailed balance** condition stated in Equation A.4, then the invariant distribution is guaranteed to be unique and equals π.

$$Q(y|x)\pi(x) = Q(x|y)\pi(y) \tag{A.4}$$

The ‘detailed balance’ condition ensures the number of moves from x to y equals the

[1] A right stochastic matrix A is a matrix where $A(i,j) \geq 0$ and $\sum\limits_j A(i,j) = 1$

number of moves from y to x along the chain. The number of moves from x to y is the probability of being at x, $\pi(x)$, times the probability of proposing the next move to be y, $Q(y|x)$.

For continuous distributions, the Markov chain converges to the invariant distribution π if

$$\pi^{\star}(dy) = \int \pi(y)dy \tag{A.5}$$

The transition matrix Q is defined so that the $(i,j)^{th}$ element states the probability of moving from state i to state j. This is defined as $Q(j|i)$ in this appendix. At each step along the chain, the probability of picking a sample in an interval dy is defined by $\pi^{\star}(dy)$ as in Equation A.6. This is defined as the integral of the probability of being at any other point x along the domain R^d times the transition probability from that point x to the interval dy.

$$\pi^{\star}(dy) = \int_{R^d} Q(dy|x)\pi(x)dx \tag{A.6}$$

For a particular interval $A : dy$, assume the transition kernel $Q(dy|x)$ is expressed as:

$$Q(dy|x) = \int_A Q(y|x)dy + r(x)\delta_x(dy) \tag{A.7}$$

where $\delta_x(dy) = 1$ if $x \in dy$ and 0 otherwise, and $r(x) = 1 - \int_{R^d} Q(y|x)dy$ is the probability that the chain remains at x.

If function $Q(y|x)$ satisfies the “detailed balance” condition where

$$Q(y|x)\pi(x) = Q(x|y)\pi(y) \tag{A.8}$$

then $\pi(.)$ is the invariant stochastic vector of Q. The following derivation proves convergence of the target distribution when the detailed balance condition is satisfied.

From A.6,

$$\begin{aligned}
\pi^{\star}(dy) &= \int_{R^d} Q(dy|x)\pi(x)dx && \text{(A.9)}\\
&= \int_{R^d} \Big[\int_A Q(y|x)dy\Big]\pi(x)dx + \int_{R^d} r(x)\delta_x(A)\pi(x)dx && \text{(A.10)}\\
&= \int_A \Big[\int_{R^d} Q(y|x)\pi(x)dx\Big]dy + \int_A r(x)\pi(x)dx \ \{\delta_x\text{=1 for } x \in A\} && \text{(A.11)}\\
&= \int_A \Big[\int_{R^d} Q(x|y)\pi(y)dx\Big]dy + \int_A r(x)\pi(x)dx \ \{\text{detailed balance}\} && \text{(A.12)}\\
&= \int_A \Big[\int_{R^d} Q(x|y)dx\Big]\pi(y)dy + \int_A r(x)\pi(x)dx && \text{(A.13)}\\
&= \int_A (1 - r(y))\pi(y)dy + \int_A r(x)\pi(x)dx && \text{(A.14)}
\end{aligned}$$

$$= \int_A \pi(y)dy - \int_A r(y)\pi(y)dy + \int_A r(x)\pi(x)dx \tag{A.15}$$

$$= \int_A \pi(y)dy \tag{A.16}$$

The Markov chain that satisfies the ‘detailed balance’ condition is said to be ‘reversible’. To achieve the detailed balance, the simplest choice of a proposal distribution is one where $Q(y|x) = \pi(y)$. This implies the ability to sample directly from the target distribution. This is not helpful as MCMC was needed in the first place to approximate the sampling. An alternative solution is the Metropolis-Hastings algorithm described next.

A.2 Metropolis-Hastings algorithm for MCMC

In 1953, Metropolis et. al. placed the foundations of a general algorithm that guarantees convergence of the MCMC to the target proposal distribution π. This was later generalised by Hastings (1970) [67]. For a selected proposal distribution $Q(y|x)$, most likely Q will not satisfy the detailed balance for all (x, y) pairs. For some x and y choices,

$$Q(y|x)\pi(x) > Q(x|y)\pi(y) \tag{A.17}$$

The process would then move from x to y too often and from y to x too rarely. A convenient way to correct this is to reduce the number of moves from x to y by introducing an **acceptance probability** $\alpha(y|x) < 1$ that the move is made.

$$Q_{MH}(y|x) \equiv Q(y|x)\alpha(y|x), \qquad x \neq y \tag{A.18}$$

$\alpha(y|x)$ is to be determined. Notice that if $Q(y|x)\pi(x) > Q(x|y)\pi(y)$ then the move from y to x is not made enough times so $\alpha(x|y)$ should be made as large as possible. Being a probability, the largest is to set it to 1 ($\alpha(x|y) = 1$).

To satisfy the detailed balance

$$Q_{MH}(y|x)\pi(x) = Q_{MH}(x|y)\pi(y) \tag{A.19}$$

$$Q(y|x)\alpha(y|x)\pi(x) = Q(x|y)\alpha(x|y)\pi(y) \tag{A.20}$$

Since $\alpha(x|y) = 1$ then

$$\alpha(y|x) = \frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)} \tag{A.21}$$

To accommodate for both cases [67],

$$\alpha(y|x) = \min\{1, \frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)}\} \tag{A.22}$$

As mentioned earlier, MCMC can be used to sample from a distribution that can be evaluated at any point up to a constant normalising factor. As the acceptance probability α only contains the ratio $\frac{\pi(y)}{\pi(x)}$, the normalising factor cancels and is not required for the calculations.

The Metropolis-Hastings algorithm remains one of the most influential algorithms in modern science and engineering [8]. Many other common algorithms are special cases of the general Metropolis-Hastings algorithm, such as Gibbs sampling, hybrid MCMC and Monte Carlo Expectation-Maximisation [8]. Algorithm A.1 shows the general Metropolis-Hastings algorithm. The algorithm requires a choice of the sample size which represents the length of the Markov chain; n_{mc}, as well as an initial element x_0. Recall that the initial element does not affect the convergence of the algorithm. The distribution $\mathscr{U}[0,1]$ is a uniform distribution in the closed interval from 0 to 1.

```
initialise $x_0$
for $i$ = 1 to $n_{mc}$ do
    sample $x^\star$ from $Q(x^\star|x_{i-1})$
    calculate $\alpha(x^\star|x_{i-1}) = \min\{1, \frac{\pi(x^\star)Q(x_{i-1}|x^\star)}{\pi(x_{i-1})Q(x^\star|x_{i-1})}\}$
    sample $u$ from $\mathscr{U}[0,1]$
    if $u < \alpha(x^\star|x_{i-1})$ then
        $x_i = x^\star$
    else
        $x_i = x_{i-1}$
```

Algorithm A.1: The General Metropolis-Hastings Algorithm

Figure A.1 shows a distribution of sample elements generated using the Metropolis Hastings algorithm. For this example, $\pi(x) = 0.3\mathscr{N}(x, 0.3, 0.5) + 0.7\mathscr{N}(x, 0.7, 0.2)$, and $Q(y|x) = \mathscr{U}[-\delta, \delta](y-x)$, where $\mathscr{N}$ is the normal (i.e. Gaussian) distribution and $\mathscr{U}$ is a uniform distribution within a closed interval. The figure shows how the distribution converges as the sample size increases.

Accepting the moves with a probability guarantees convergence, yet the performance of the algorithm cannot be known in advance. It might take too long to converge depending on the choice of the transition matrix Q. A transition matrix where the majority of the moves are rejected converges slower. Thus the **acceptance rate** ρ_{accept} along the chain is often used to assess the performance, and thus the convergence. The acceptance rate

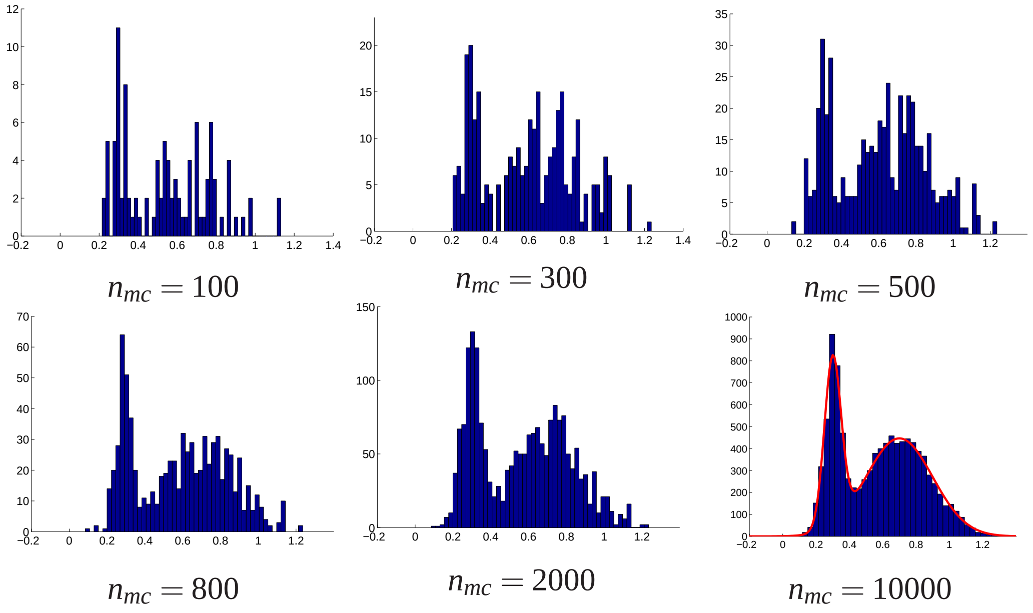


Figure A.1: Histogram of Markov chain sample elements for a given target distribution π and a closed interval uniform proposal distribution Q using Metropolis-Hastings Algorithm. The last plot superimposes the actual function π on the histogram.

ρ_{accept} is the ratio of the number of accepted moves to the length of the Markov chain. The acceptance rate should be around 0.5 for a random walk chain.

Another method to assess the convergence is to take one parameter, for example the mean of the sample, and run several independent Markov chains. The convergence is assessed by comparing the value of this parameter between chains. If $\pi(x) = \mathcal{N}(x, 0.7, 0.25)$ and $Q(y|x) = \mathcal{U}[-\delta, \delta](y-x)$ then Figure A.2 plots the mean of the retrieved sample using the Metropolis-Hastings algorithm for 3 different Markov chains. Convergence is believed to be reached for $n_{mc} > 500$.

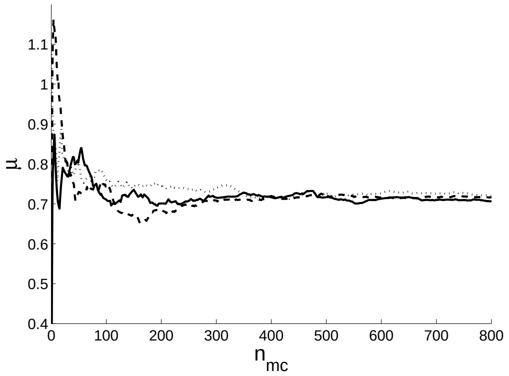


Figure A.2: Convergence of the sample mean under different runs of the Markov chain

Appendix B

Using MLE for Fitting a Gaussian to a Constrained Domain Training Data

When estimating the conditional probability density function $p(x|e)$ by a Gaussian, the area under the pdf equals 1 as the area under the Gaussian curve is one. If $p(x|e) : \mathbb{R} \rightarrow [0,1]$ is approximated with the normal $\mathcal{N}(\mu,\sigma)$ then

$$\int\limits_{R^d} p(x|e) = \int\limits_{\mathbb{R}} \mathcal{N}(\mu,\sigma) = 1 \tag{B.1}$$

Nevertheless, when the domain of the function x is to a closed interval $[a,b]$ or half-open interval $[a,\infty)$ or $(\infty,b]$, the area under the Gaussian would not be 1. For constrained domains, the conditional pdf needs to be normalized. If φ is the Gaussian function defined in Equation B.2,

$$\varphi_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{B.2}$$

then the conditional probability density function for a closed interval domain $[a,b]$ is defined to be,

$$p(x|e) = \frac{\varphi_{\mu,\sigma}(x)}{\int\limits_a^b \mathcal{N}(\mu,\sigma)} \tag{B.3}$$

To be able to calculate the conditional density function as in Equation B.3, one needs to evaluate the area under the Gaussian for a fixed interval. First, the z-score of each boundary limit is calculated to transform it into the standard distribution $Z = \mathscr{N}(0,1)$ Gaussian. Z-score for each value is calculated to be,

$$ZScore(x) = \frac{x-\mu}{\sigma} \tag{B.4}$$

Tables generated from numerically integrating the standard Gaussian distribution are available. The tables present the area above and below each point in the standard Gaussian distribution Z. Figure B.1 presents the standard Gaussian distribution Z and the area under the curve for the Z-score of 1 (representing 1 standard deviations from the mean). From the available calculators or tables, the area above the z-score of 1 equals 0.1586, while the area under the z-score is calculated from 1-0.1586 = 0.8414.

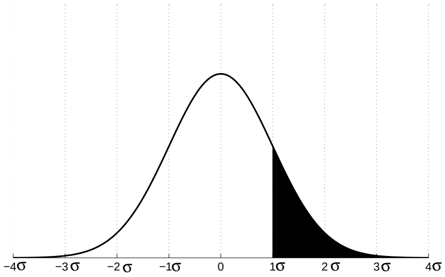


Figure B.1: Z-score transforms the Gaussian $\mathscr{N}(\mu,\sigma)$ into $\mathscr{N}(1,0)$

If the domain is constrained from both sides $R^d = [a,b]$, and $f(z)$ gives the area above a point in the distribution then the integral required for normalizing in Equation B.3 is calculated from:

$$\int_a^b \mathscr{N}(\mu,\sigma) = f(ZScore(a)) - f(ZScore(b)) \tag{B.5}$$

Appendix C

The Posterior Probability - a derivation

The posterior in Equation C.1 can be rearranged.

$$p(\omega|Y) = \tfrac{1}{\mathscr{G}} \prod_i p(o_{x_i}|x_i)p(x_i) \prod_j p(o_{y_j}|y_j)p(y_j) \prod_{ij} p(o_{z_{ij}}|z_{ij})p(z_{ij}|x_i,y_j)p(c|\{z_{ij}\}) \quad \text{(C.1)}$$

Using Bayes, the first product can be substituted

$$p(x_i|o_{x_i}) = \frac{p(o_{x_i}|x_i)p(x_i)}{p(o_{x_i})} \quad \text{(C.2)}$$

The denominator is a constant that can be part of the normalizing factor $\mathscr{G}$. Similarly $p(y_i|o_{y_i})$ can be rewritten. Accordingly Equation C.1 because:

$$p(\omega|Y) = \tfrac{1}{\mathscr{Z}} \prod_i p(x_i|o_{x_i}) \prod_j p(y_j|o_{y_j}) \prod_{ij} p(o_{z_{ij}}|z_{ij})p(z_{ij}|x_i,y_j)p(c|\{z_{ij}\}) \quad \text{(C.3)}$$

For the third product $\prod_{ij} p(o_{z_{ij}}|z_{ij})p(z_{ij}|x_i,y_j)$, then

$$\begin{aligned}
p(o_{z_{ij}}|z_{ij})p(z_{ij}|x_i,y_j) &= \frac{p(z_{ij}|o_{z_{ij}})p(o_{z_{ij}})}{p(z_{ij})}p(z_{ij}|x_i,y_j) && \text{(C.4)}\\
&\propto \frac{p(z_{ij}|o_{z_{ij}})}{p(z_{ij})}p(z_{ij}|x_i,y_j) && \text{(C.5)}\\
&= \frac{p(z_{ij}|o_{z_{ij}})p(z_{ij}|x_i,y_j)}{\sum_{x_i,y_j} p(z_{ij}|x_i,y_i)} && \text{(C.6)}\\
&\propto p(z_{ij}|o_{z_{ij}})p(z_{ij}|x_i,y_j) && \text{(C.7)}\\
&= p(z_{ij}|o_{z_{ij}},x_i,y_j) && \text{(C.8)}
\end{aligned}$$

As $\sum_{x_i,y_j} p(z_{ij}|x_i,y_i)$ is constant

Appendix D

MOSEL program for formulating an Integer Program

```
model 'example'
uses 'mmetc','mmxprs';

declarations
terminals= 6
constraints= 3
nodesSize= 14
omegaSize= 17

THETAT: array(1..terminals,1..omegaSize) of integer
THETAC: array(1..constraints,1..omegaSize) of integer
THETAK: array(1..nodesSize,1..omegaSize) of integer
cost: array(1..omegaSize) of real
seed: array (1..omegaSize) of mpvar
DELTA: array (1..omegaSize, 1..omegaSize) of mpvar
end-declarations

! read data
diskdata(ETC_IN,'ch4_thetat.dat',THETAT)
diskdata(ETC_IN,'ch4_thetac.dat',THETAC)
diskdata(ETC_IN,'ch4_thetak.dat',THETAK)
diskdata(ETC_IN,'ch4_c.dat',cost)
!-----------------------------------------------------------------
! build ILP model
!-----------------------------------------------------------------
! objective is min cost*omega = co
f:= SUM(i in 1..omegaSize) cost(i) * seed (i)
```

```
! every terminal must be explained
forall(i in 1..terminals)
  PASSIGN(i):= SUM(k in 1..omegaSize) THETAT(i,k)*seed(k) >= 1

! every constraint must also be satisfied
forall (i in 1..constraints)
  BASSIGN(i):= SUM(k in 1..omegaSize) THETAC(i,k)*seed(k) <= 1

! check for conflict
forall (j in 1..omegaSize, k in (j+1)..omegaSize)
  CASSIGN(j,k) := DELTA(j,k) <= seed (j)
forall (j in 1..omegaSize, k in (j+1)..omegaSize)
  DASSIGN(j,k) := DELTA(j,k) <= seed (k)
forall (j in 1..omegaSize, k in (j+1)..omegaSize)
  EASSIGN(j,k) := DELTA(j,k) >= seed (j) + seed (k) - 1
forall (j in 1..omegaSize, k in (j+1)..omegaSize)
  FASSIGN (j,k) := SUM(i in 1..nodesSize) (THETAK(i,j) - THETAK (i,k))
                   * THETAK(i,j) * THETAK (i,k) * DELTA(j,k)  = 0

forall (i in 1..omegaSize) seed(i) is_binary

exportprob(EP_MIN,’ch4’,f)
exit(0)

end-model
```

Appendix E

Experimental Results for the *Bicycles* Problem

This appendix presents complete results for the seven sequences in the bicycles dataset from Chapter 5. For each record in the tables below, the minimum, mean and standard deviations are recorded from 40 runs. During each run, 10 parallel chains are run and the MAP is the maximum across the parallel chains. For each sequence, RJMCMC (two initial states) is compared to RJMCMC-SA (two initial states). Moreover, online performance is shown for the same settings. Some of the results printed here have been shown in various tables in Section 5.6.

E.1 MAP results

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	57.86	57.89	0.08
	×			5,000	57.86	57.86	0.00
×			×	5,000	57.86	57.90	0.11
	×		×	5,000	57.86	57.86	0.00
×		×		1000/au	57.86	59.60	1.13
	×	×		1000/au	57.86	60.80	1.80
×		×	×	1000/au	58.83	60.41	0.90
	×	×	×	1000/au	58.23	61.29	2.28

Table E.1: MAP results - 1st sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	4.63	4.63	0.00
	×			5,000	4.63	4.63	0.00
×			×	5,000	4.63	4.64	0.00
	×		×	5,000	4.63	4.64	0.00
×		×		1000/au	4.63	4.63	0.00
	×	×		1000/au	4.63	4.63	0.00
×		×	×	1000/au	4.63	6.97	4.17
	×	×	×	1000/au	4.63	15.32	6.49

Table E.2: MAP results - 2^{nd} sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	420.23	428.12	3.87
	×			5,000	420.20	424.31	2.19
×			×	5,000	421.00	429.30	3.23
	×		×	5,000	420.50	423.98	2.36
×		×		1000/au	426.64	434.42	4.24
	×	×		1000/au	435.90	442.53	3.71
×		×	×	1000/au	429.57	432.87	1.86
	×	×	×	1000/au	433.13	444.50	7.38

Table E.3: MAP results - 3^{rd} sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	6073.10	6086.67	15.69
	×			5,000	6071.30	6080.02	4.62
×			×	5,000	6073.60	6079.88	3.43
	×		×	5,000	6071.10	6078.40	2.36
×		×		1000/au	5895.99	5941.1	24.13
	×	×		1000/au	5950.38	5961.6	7.78
×		×	×	1000/au	5925.13	5949.1	16.45
	×	×	×	1000/au	5929.47	5943.7	10.96

Table E.4: MAP results - 4^{th} sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	4937.10	4941.01	4.06
	×			5,000	4943.71	4939.37	1.96
×			×	5,000	4943.71	4943.71	3.59
	×		×	5,000	4943.71	4939.33	1.87
×		×		1000/au	4927.60	4963.7	22.45
	×	×		1000/au	4956.55	4968.5	5.16
×		×	×	1000/au	4924.08	4945.8	12.60
	×	×	×	1000/au	4929.63	4956.3	16.17

Table E.5: MAP results - 5^{th} sequence

E.2 Accuracy Results

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	805.55	814.22	2.29
	×			5,000	806.05	811.62	2.02
×			×	5,000	811.70	814.71	1.69
	×		×	5,000	807.00	811.50	2.36
×		×		1000/au	800.35	804.00	2.62
	×	×		1000/au	787.62	797.96	4.54
×		×	×	1000/au	797.30	806.61	6.09
	×	×	×	1000/au	796.72	805.08	4.56

Table E.6: MAP results - 6th sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	418.14	437.32	8.51
	×			5,000	401.29	429.19	12.14
×			×	5,000	429.96	451.92	9.29
	×		×	5,000	411.58	433.50	7.76

Table E.7: MAP results - 7th sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	91.38	90.52	0.91
	×			5,000	91.38	90.69	1.45
×			×	5,000	91.38	88.36	1.09
	×		×	5,000	91.38	87.46	1.79
×		×		1000/au	91.38	90.34	2.18
	×	×		1000/au	91.38	91.20	2.98
×		×	×	1000/au	96.55	89.48	3.20
	×	×	×	1000/au	89.66	91.90	2.58

Table E.8: Accuracy results - 1st sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	100.00	100.00	0.00
	×			5,000	100.00	99.26	1.56
×			×	5,000	100.00	100.00	0.00
	×		×	5,000	100.00	100.00	0.00
×		×		1000/au	100.00	100.00	0.00
	×	×		1000/au	100.00	100.00	0.00
×		×	×	1000/au	96.30	96.30	0.00
	×	×	×	1000/au	96.30	96.30	0.00

Table E.9: Accuracy results - 2nd sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	82.81	82.87	1.27
	×			5,000	82.03	82.93	1.29
×			×	5,000	85.94	87.68	0.89
	×		×	5,000	82.03	83.36	1.65
×		×		1000/au	90.63	95.98	3.42
	×	×		1000/au	92.19	96.07	3.54
×		×	×	1000/au	91.41	96.30	2.99
	×	×	×	1000/au	93.75	97.02	2.23

Table E.10: Accuracy results - 3rd sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	82.54	82.96	1.21
	×			5,000	82.54	82.70	1.95
×			×	5,000	84.92	83.93	1.09
	×		×	5,000	82.54	83.15	1.31
×		×		1000/au	84.13	82.94	2.52
	×	×		1000/au	84.13	93.49	2.11
×		×	×	1000/au	84.82	84.68	3.18
	×	×	×	1000/au	88.89	86.75	1.63

Table E.11: Accuracy results - 4th sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	93.43	93.12	0.92
	×			5,000	91.24	92.65	0.87
×			×	5,000	93.43	91.90	0.79
	×		×	5,000	94.16	92.65	0.90
×		×		1000/au	94.89	90.66	2.92
	×	×		1000/au	91.97	88.10	2.67
×		×	×	1000/au	93.43	89.05	3.10
	×	×	×	1000/au	92.70	88.25	2.19

Table E.12: Accuracy results - 5th sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	69.64	68.97	1.10
	×			5,000	70.53	69.62	1.02
×			×	5,000	68.75	68.53	1.68
	×		×	5,000	71.43	70.98	1.04
×		×		1000/au	68.75	64.38	3.02
	×	×		1000/au	70.54	63.39	2.82
×		×	×	1000/au	72.32	68.04	1.56
	×	×	×	1000/au	71.42	67.14	2.34

Table E.13: Accuracy results - 6th sequence

RJMCMC	**RJMCMC-SA**	**Online**	**From Local Max**	n_{mc}	min	μ	σ
×				5,000	45.18	45.23	1.30
	×			5,000	45.69	46.74	0.90
×			×	5,000	45.69	47.28	1.18
	×		×	5,000	47.21	47.61	0.88

Table E.14: Accuracy results - 7th sequence

Appendix F

Conference Posters

Bibliography

[1] *Merriam-Webster's biographical dictionary*. Springfield, Mass. : Merriam-Webster, 1995.

[2] V. Ablavsky, A. Thangali, and S. Sclaroff. Layered graphical models for tracking partially-occluded objects. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2008.

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

[4] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 798–805, 2006.

[5] A. Aho, R. Sethi, and J. Ulman. *Compilers: principles, techniques and tools*. Addison-Wesley, 1986.

[6] K. Alahari, P. Kohli, and P. Torr. Reduce, reuse and recycle: Efficiently solving multi-label MRFs. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[7] J. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.

[8] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[9] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Theory Algorithms and Software. John Wiley and Sons, 2001.

[10] R. Bardohl, G. Taentzer, M. Minas, and A. Schurr. Application of graph transformation to visual languages. In H. Ehrig, G. Engels, H. Kreowski, and G. Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformations*, volume 2, pages 105–180. World Scientific Publishing, Singapore, 1997.

[11] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms - part 1: Methodology and experiments with synthesized data. *IEEE Transactions on Image Processing*, 11(9):972–983, 2002.

[12] C. Benabdelkader and L. Davis. Detection of people carrying objects : a motion-based recognition approach. In *Proc. Int. Conf. on Automatic Face and Gesture Recognition (FGR)*, pages 378–384, 2002.

[13] B. Bennett, D. Magee, A. Cohn, and D. Hogg. Enhanced tracking and recognition of moving objects by reasoning about spatio-temporal continuity. *Image and Vision Computing*, 26:67–81, 2008.

[14] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 744–750, 2006.

[15] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992.

[16] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 1st edition, 2006.

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

[18] A. Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of Royal Society London*, 352:1257–1265, 1997.

[19] 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.

[20] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 26(9):1124–1137, 2004.

[21] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 23(11):1222–1239, 2001.

[22] A. Branca, M. Leo, G. Attolico, and A. Distante. Detection of objects carried by people. In *Proc. Int. Conf on Image Processing (ICIP)*, volume 3, pages 317–320, 2002.

[23] R. A. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17(1-3):285–348, 1981.

[24] F. Bunyak, I. Ersoy, and S. R. Subramanya. A multi-hypothesis approach for salient object tracking in visual surveillance. In *Proc. Int. Conf of Image Processing (ICIP)*, volume 2, pages 446–449, 2005.

[25] M. Chan, A. Hoogs, R. Bhotika, A. Perera, J. Schmiederer, and G. Doretto. Joint recognition of complex events and track matching. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1615–1622, 2006.

[26] H. Chen, Z.-J. Xu, Z.-Q. Liu, and S.-C. Zhu. Composite templates for cloth modeling and sketching. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[27] C. H. Chuang, J. W. Hsieh, L. W. Tsai, S. Y. Chen, and K. C. Fan. Carried object detection using ratio histogram and its application to suspicious event analysis. *IEEE Trans. on Circuits and Systems for Video Technology*, 19(6):911–916, 2009.

[28] I. J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[29] I. J. Cox and S. L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 18(2):138–150, 1996.

[30] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 1, pages 434–441, 1999.

[31] R. Cutler and L. Davis. View-based detection and analysis of periodic motion. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 495–500, 1998.

[32] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):781–796, 2000.

[33] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005.

[34] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proc. of Int. Conf. on Computer Vision (ICCV)*, 1997.

[35] L. Davis and T. Henderson. Hierarchical constraint processes for shape analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 3(3):265–277, 1981.

[36] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50(1):45–71, 2003.

[37] M. Dimitrijevic, V. Lepetit, and P. Fua. Human body pose detection using Bayesian spatio-temporal templates. *Computer Vision and Image Understanding*, 104(2):127–139, 2006.

[38] R. Donner, B. Micusik, G. Langs, and H. Bischof. Sparce MRF appearance models for fast anatomical structure localisation. In *Proc. British Machine Vision Conference (BMVC)*, volume 2, pages 1080–1089, 2007.

[39] A. Elgammal and L. Davis. Probabilistic framework for segmenting people under occlusion. In *Proc. of International Conference on Computer Vision (ICCV)*, 2001.

[40] 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.

[41] M. Everingham and J. Winn. The PASCAL visual object classes challenge (VOC2007) development kit. Technical report, 2007.

[42] 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.

[43] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2000.

[44] J. Ferryman, editor. *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*. IEEE, New York, 2006.

[45] D. O. FICO. XPRESS-MP solver - version 19.00.17, 2007.

[46] N. Fisher. *Statistical analysis of circular data*. Cambridge University Press, 1993.

[47] A. Fossati, M. Dimitrijevic, V. Lepetit, and P. Fua. Bridging the gap between detection and tracking for 3D monocular video-based motion capture. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[48] W. R. Franklin. PNPOLY - point inclusion in polygon test, 1994. http://www.ecse.rpi.edu/Homepages/wrf/Research.

[49] S. French and D. R. Insua. *Statistical Decision Theory*. Kindall's library of statistics. Arnold - Hodder Headline Group, London, 2000.

[50] 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.

[51] N. M. Ghanem and L. S. Davis. Human appearance change detection. In *Image Analysis and Processing (ICIAP)*, pages 536–541, 2007.

[52] N. Gheissari, T. B. Sebastian, and R. Hartley. Person reidentification using spatiotemporal appearance. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1528–1535, 2006.

[53] A. Gilbert and R. Bowden. Tracking objects across cameras by incrementally learning inter-camera colour calibration and patterns of activity. In *Proc. European Conference on Computer Vision (ECCV)*, volume 3952, pages 125–136, 2006.

[54] S. Godsill. On the relationship between Markov chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, 10(2):230–248, 2001.

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

[56] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *Proc. International Conference on Computer Vision (ICCV)*, 2003.

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

[58] 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.

[59] M. Hahnel, D. Klunder, and K. F. Kraiss. Color and texture features for person recognition. In *Proc. IEEE Int. Joint Conf. on Neural Networks*, volume 1, page 652, 2004.

[60] P. A. Hall. Equivalence between and/or graphs and context-free grammars. *Communications of the ACM*, 16(7):444–445, 1973.

[61] 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.

[62] 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.

[63] J. Han and B. Bhanu. Individual recognition using gait energy image. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 28(2):316–322, 2006.

[64] I. Haritaoglu, R. Cutler, D. Harwood, and L. S. Davis. Backpack: detection of people carrying objects using silhouettes. In *Proc. Int. Conf. on Computer Vision (ICCV)*, volume 1, pages 102–107, 1999.

[65] I. Haritaoglu, D. Harwood, and L. Davis. Hydra: Multiple people detection and tracking using silhouettes. In *Proc. IEEE Workshop on Visual Surveillance*, 1999.

[66] I. Haritaoglu, D. Harwood, and L. S. Davis. W^4: real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):809–830, 2000.

[67] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[68] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *Proc. European Conference on Computer Vision (ECCV)*, volume 5302, pages 30–43, 2008.

[69] 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.

[70] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 28(4):663–671, 2006.

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

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

[73] M. Isard and A. Blake. Condensationconditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[74] 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.

[75] Y. Ivanov, C. Stauffer, A. Bobick, and W. E. L. Grimson. Video surveillance of interactions. In *IEEE Workshop on Visual Surveillance (VS)*, pages 82–89, 1999.

[76] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 26–33, 2005.

[77] 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.

[78] 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.

[79] P. Jorge, J. Marques, and A. Abrantes. On-line tracking groups of pedestrians with Bayesian networks. In *Proc. Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2004.

[80] U. Kastens. Ordered attributed grammars. *Acta Informatica*, 13:229–256, 1980.

[81] R. Kaucic, A. G. Amitha Perera, G. Brooksby, K. J., and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 990–997, 2005.

[82] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2007.

[83] V. Kettnaker and R. Zabih. Bayesian multi-camera surveillance. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 259 – 265, 1999.

[84] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *Proc. Eurpoean Conference of Computer Vision (ECCV)*, volume 4, pages 279–290, 2004.

[85] Z. Khan, T. Balch, and F. Dellaert. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 28(12):1960–1972, 2006.

[86] 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.

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

[88] D. Knuth. The genesis of attribute grammars. In *Int. Conf. on Attribute Grammars and their Applications*, pages 1–12, 1990.

[89] P. Kohli, M. P. Kumar, and P. Torr. P3 and beyond: Solving energies with higher order cliques. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[90] V. Kolmogorov and R. Zabih. What energy functions can be minimized using graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 26(2):147–159, 2004.

[91] B. Korte and J. Vygen. *Combinatorial Optimization, Theory and Algorithms*. Algorithms and Combinatorics. Springer, 4th edition, 2008.

[92] I. Laptev and P. Perez. Retrieving actions in movies. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2007.

[93] C.-S. Lee and A. Elgammal. Carrying object detection using pose preserving dynamic shape models. In *Fourth Conference of Articulated Motion and Deformable Objects (AMDO)*, pages 315–325, 2006.

[94] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2007.

[95] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[96] 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.

[97] C. C. Loy, T. Xiang, and S. Gong. Multi-camera activity correlation analysis. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[98] F. Lv, X. Song, B. Wu, V. Singh, and R. Nevatia. Left luggage detection using Bayesian inference. In *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, pages 83–90, 2006.

[99] C. Madden and M. Piccardi. Height measurement as a session-based biometric for people matching across disjoint camera views. *Image and Vision Computing*, 1:282–286, 2005.

[100] D. Magee. Tracking multiple vehicles using foreground, background and motion models. In *Proc. Workshop on Statistical Methods in Video Processing*, pages 7–12, 2002.

[101] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 205–210, 2004.

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

[103] MATHWORKS. Optimization toolbox - MATLAB version r2008a, 2008.

[104] D. Moore and I. Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *National conference on Artificial intelligence (AAAI)*, pages 770 – 776, 2002.

[105] C. Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Trans. on Automated Control*, 22(3):302–312, 1977.

[106] K. G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16(3):682–687, 1968.

[107] H. Nanda, C. Benabdelkedar, and L. Davis. Modelling pedestrian shapes for outlier detection: a neural net based approach. In *Proc. Intelligent Vehicles Symposium*, pages 428–433, 2003.

[108] 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.

[109] 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.

[110] M. S. Nixon and J. N. Carter. Automatic recognition by gait. *Proceedings of the IEEE*, 94(11):2013–2024, 2006.

[111] 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.

[112] S. Oh, S. Russell, and S. Sastry. Markov chain Monte Carlo data association for multi-target tracking. *IEEE Trans. on Automatic Control*, 54(3):481–497, 2009.

[113] H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *Proc. Int. Joint Conferences on Aritificial Intelligence (IJCAI)*, pages 1160–1171, 1999.

[114] A. B. Poore. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Computational Optimization and Applications*, 3(1):27–57, 1994.

[115] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 29(1):65–81, 2007.

[116] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24(6):843–854, 1979.

[117] S. Richardson and P. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of Royal Statistical Society*, 59(5):731–792, 1997.

[118] N. Ripperda and C. Brenner. Reconstruction of facade structures using a formal grammar and RJMCMC. In *Annual Symposium of the German Association for Pattern Recognition (DAGM)*, pages 750–759, 2006.

[119] J. Roberts, editor. *The Oxford dictionary of the classical world*. Oxford University Press, Oxford, 2007.

[120] M. Rota and M. Thonnat. Video sequence interpretation for visual surveillance. In *IEEE Int. Workshop on Visual Surveillance (VS)*, Dublin, Ireland, 2000.

[121] C. Rother, V. Kolmogorov, and A. Blake. Grabcut -interactive foreground extraction using iterated graph cuts. In *ACM Transa. on Graphics (SIGGRAPH)*, 2004.

[122] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[123] S. Russell and P. Norvig. *Artificial intelligence : a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2nd edition, 2003.

[124] M. S. Ryoo and J. K. Aggarwal. Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[125] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, West Sussex, England, 1998.

[126] K. Shafique, L. Mun Wai, and N. Haering. A rank constrained continuous formulation of multi-frame multi-target tracking problem. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[127] 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.

[128] J. Siskind. Visual event classification via force dynamics. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 149–155, 2000.

[129] J. Sivic, C. L. Zitnick, and R. Szeliski. Finding people in repeated shots of the same scene. In *Proc. British Machine Vision Conference (BMVC)*, volume 3, pages 909–918, 2006.

[130] J. Q. Smith. *Decision Analysis: A Bayesian Approach*. Chapman and Hall Ltd., 1988.

[131] K. Smith. *Bayesian Methods for Visual Multi-object Tracking with Applications to Human Activity Recognition*. PhD thesis, Ecole Polytechnique Federale de Lausanne (EPFL), 2007.

[132] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing: Analysis and Machine Vision*. CL-Engineering, 3rd edition, 2007.

[133] C. Stauffer. Learning to track objects through unobserved regions. In *IEEE Workshop on Motion and Video Computing (WACV/MOTIONS)*, volume 2, pages 96–102, 2005.

[134] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 246–252, 1999.

[135] A. Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201, 1995.

[136] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[137] D. Tao, X. Li, S. J. Maybank, and W. Xindong. Human carrying status in visual surveillance. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[138] P. Torr. Tutorial: Markov random fields for vision and graphics - British and Machine Vision Conference, 2008.

[139] S. Tran and L. Davis. Event modeling and recognition using Markov logic networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2008.

[140] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518, 2001.

[141] Y. Wang, E. Chang, and K. Cheng. A video analysis framework for soft biometry security surveillance. In *Proc. ACM Int. workshop on Video surveillance and sensor networks*, Singapore, 2005. ACM Press.

[142] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer, 2nd edition, 2003.

[143] W. L. Winston. *Operations Research : Applications and Algorithms*. PWS-Kent Pub, Boston, 2nd edition, 1991.

[144] W. Woo and A. Ortega. Stereo image compression with disparity compensation using the mrf model. In *Proc. Visual Communications and Image Processing (VCIP)*, 1996.

[145] 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.

[146] G. Wu, A. Rahimi, E. Y. Chang, G. Kingshy, T. Tsai, J. Ankur, and Y. Wang. Identifying color in motion in video sensors. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 561–569, 2006.

[147] 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.

[148] 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.

[149] 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.

[150] W. Zajdel and B. J. A. Krose. A sequential Bayesian algorithm for surveillance with nonoverlapping cameras. *Int. Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 19(8):977–996, 2005.

[151] W. Zajdel, Z. Zivkovic, and B. J. A. Krose. Keeping track of humans: Have i seen this person before? In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2081–2086, 2005.

[152] L. Zhang, L. Yuan, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[153] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 21(8):690–706, 1999.

[154] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2004.

[155] L. ZhiHua and K. Komiya. Region-wide automatic visual search and pursuit surveillance system of vehicles and people using networked intelligent cameras. In *Proc. Int. Conf. on Signal Processing (ICSP)*, volume 2, pages 945–8, 2002.

[156] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and A. S. Fast human detection using a cascade of histograms of oriented gradients. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1491–1498, 2006.

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

[158] D. Zou, Q. Zhao, H. S. Wu, and Y. Q. Chen. Reconstructing 3D motion trajectories of particle swarms by global correspondence selection. In *Proc. Int. Conf. on Computer Vision (ICCV)*, 2009.